# TDM (TEST DATA MANAGEMENT) UPGRADE PROCEDURE TO V9.2

- This document describes the following:
  - How to upgrade TDM to the present version - V9.2.
  - 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 9.2 Installation - Prerequisites

- Upgrade Fabric to Fabric V8.1.5 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 the TDM V9.2 installation, please read the TDM Installation article in the TDM Configuration.

## 3 Fabric Upgrade

- Upgrade the Fabric version to 8.1.5.
- If your current Fabric version is older than Fabric 8.0 and is upgraded to Fabric 8.1.5, perform the following:
  - Before upgrading the project, edit the config.ini file and uncomment PACKAGE_NAMES_CLASS_LOADING_FILTER parameter. Set it to be empty in order to disable the Fabric and project dependencies isolation feature.
  - After upgrading the project, the PACKAGE_NAMES_CLASS_LOADING_FILTER parameter needs to be commented.
  - For more information, read FABRIC UPGRADE PROCEDURE TO V8.0.

# 4 TDM Upgrade from Version 9 or 9.1 to 9.2

## 4.1 Import the TDM 9.2 Library

### 4.1.1 Web Studio

Step 1 – Open the TDM Project in Fabric Studio

- Open the TDM project with Fabric Studio (8.1.5 and up). The Fabric Studio will upgrade the code to the latest Fabric version.
- Click the Extension icon -  - and select TDM to install the TDM 9.2 library. **Override the existing objects.**

### 4.1.2 .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 Studio

- Open the TDM project with Fabric Studio. The Fabric Studio will upgrade the code to the latest Fabric version
- Manually delete the following:
  - TDM LU
  - TDM_LIBRARY LU
  - If your current version is 9.0, delete the TDM_Reference LU

Step 3 – Import the TDM 9.2 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**
- If your current version is 9.0, custom import the **MTables** to the project under the **References** LU:
  - TableLevelDefinitions.csv
  - Note that the TableLevelInterfaces.csv will be updated by the TDM deploy flow.
- **Optional – AI Interfaces**:
  - If you wish to add the AI-based generation configuration to the TDM project, do the following:
    - Import and edit the AI interfaces to the TDM project in the Studio.
    - Add the AI environment to the Studio.
  - Click here for more information about the AI implementation.

## 4.2 Upgrade the TDM DB

- Deploy the updated **References** LU.
- Deploy the updated **TDM** LU. The TDM deploy flow runs the **TDMDBUpgradeScripts** flow in order to upgrade the TDM DB.
  - Note that the current version is taken from the TDM_GENERAL_PARAMETERS TDM DB table, and the target version is taken from the TARGET_TDM_VERSION Global.

## 4.3 Optional – Supporting a Vertical Execution Mode

### 4.3.1 Upgrade the LUs to Support a Vertical Execution

- Run the UpgradeFabricTDMRootPopulation flow to upgrade the FABRIC_TDM_ROOT LU table's population in all LUs.
- The upgrade population is required in order to support Vertical execution of the LUs.
- Redeploy all the LUs.

### 4.3.2 TDM Portal – Update the Execution Mode in the Business Entities

- Open the Business Entity window for the Business Entities that you need to set the execution mode to Vertical and update the execution mode.

# 5 TDM Upgrade from Older TDM Versions to 9.2

- If your current TDM version is older than TDM 9.0.x, take the following steps:

## 5.1 Optional – TDM Portal – Update the Load and Delete Tasks Before Running the TDM Upgrade Flow

- TDM 9 does not support selecting the Delete option together with either the Replace IDs for the copied entities or the Generate clones for an entity option.
- If you have tasks with Delete and Load task actions, whose Replace Sequence is selected or their selection method is Entity clone, open them in the TDM portal before running the upgrade flow, and update them as follows:
  - Delete + Load + Replace Sequence task — clear either the Delete or the Replace Sequence option.
  - Delete + Load + Entity clone — clear the Delete option.

## 5.2 Import the TDM 9.2 Library

### 5.2.1 Web Studio

Step 1 – Open the TDM Project in Fabric Studio

- Open the TDM project with Fabric Studio (8.1.x). The Fabric Studio will upgrade the code to the latest Fabric version.
- Manually delete 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.

### 5.2.2 .Net (desktop) Studio

Step 1 – Open the TDM Project in Fabric Studio

- Open the TDM project with Fabric Studio (8.1.x). The Fabric Studio will upgrade the code to the latest Fabric version
- 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 it must be removed from the lib folder after the TDM upgrade process has been completed, as the TDM 9 code no longer uses the StringUtils object.
- Manually delete the following:
  - TDM LU
  - TDM_LIBRARY LU
  - TDM_Reference LU
- If the legacy project is based on 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 the Fabric debug server.

Step 2 – Import the TDM 9.2 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** to the project under the **References** LU. If the current TDM version is 8.1 => remove the conflicting MTables from the import.
- **Optional – AI Interfaces**:
  - If you wish to add the AI-based generation configuration to the TDM project, do the following:
    - Import and edit the AI interfaces to the TDM project in the Studio.
    - Add the AI environment to the Studio.

- Note that the import process creates duplicated objects in the project due to TDM 9 locating the TDM objects in subfolders in the **Shared Objects**' 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's system DB, perform the following:
        - Open 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.

## 5.3 Optional Steps

### 5.3.1 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 no longer in use.

### 5.3.2 Add Catalog-based Masking to the LU Populations

- 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.

### 5.3.3 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). It should be implemented if you wish the TDM portal to enable executing tasks created by all users, including 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 to decide whether 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 **TDMDBUpgradeScripts** 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 **References** LU.

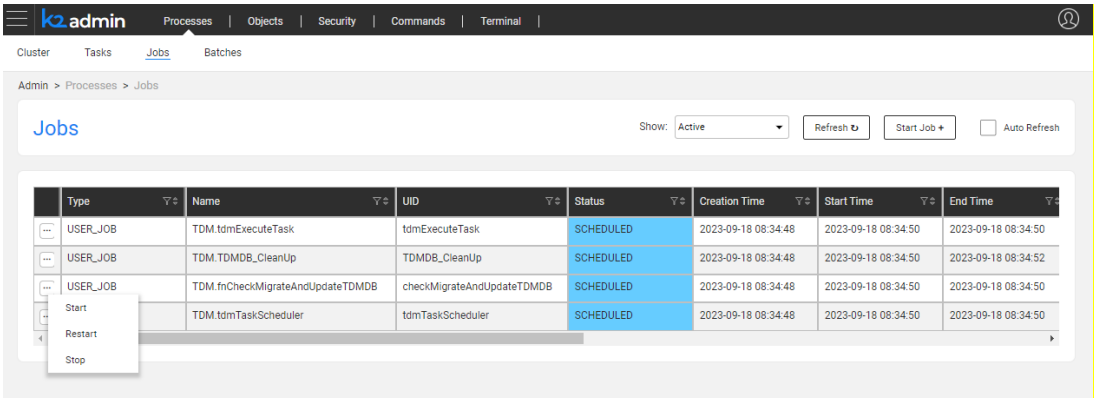
### 5.3.4 Increase the Maximum Number of Records per Table

The default limitation on number of processed records per table is 100K records. If your tables have a higher 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 greater than 100,000. For example:
  - `MAX_CONCRETE_ARRAY_SIZE=50000000`
- Restart Fabric.

## 5.4 Run the UpgradeTDMProjectToTDM9 Flow

- Take the following steps **before** running the upgrade flow:
  - Back up the TDM DB.
  - **Deploy all LUs, including the TDM LU**, to the Fabric server. It is recommended to use the soft deploy option in order to avoid starting the TDM jobs.
  - Stop all TDM jobs on Fabric if they are up and running in order to avoid locking the TDM DB tables by parallel executions of the upgrade flow 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 **current 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, perform the following to upgrade the TDM DB:

- Open the **UpgradeTDMProjectToTDM9** flow and disable the **TDMDBUpgradeScripts.**
- 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 the Fabric execution server.
- Open a Fabric console in the Fabric execution server and run the **TDMDBUpgradeScripts** flow using the following command:

  Broadway TDM.TDMDBUpgradeScripts;

  Note that the current version is taken from the TDM_GENERAL_PARAMETERS TDM DB table, and the target version is taken from the TARGET_TDM_VERSION Global.

- Run the following flows in the Studio to convert the translations to MTables and the old post process actor to the new one, if needed:
  - **convertLuTranslations**
  - **UpdatePostProcessList**

## 5.5 Validate the Upgraded TDM Objects After the Upgrade Flow Execution

- The following objects are now located in the Implementation/SharedObjects/Broadway/TDM/TDMImplementorActors directory. Open the following objects and verify that they have the same values as before the upgrade:
  - CustomLogicFlows.actor
  - TDMFilterOutTargetTables.actor
  - TDMSeqList.actor
  - TDMSeqSrc2TrgMapping.actor
  - TDMTargetTablesNames.actor
- Open the PostAndPreExecutionProcess object and verify that it is populated with the project's post-execution processes.
- TDM 9.0 locates the TDM library shared Globals under the TDM subfolder to separate them from the project's Globals. The upgrade flow moves the TDM Globals from the 'regular' shared Globals files into the new file (Implementation/SharedObjects/Java/src/com/k2view/cdbms/usercode/common/TDM/SharedGlobals.java). Open the TDM/SharedGlobals file and verify that the TDM Globals contain the required values.
- Open the following MTables and verify that they are populated properly:
  - ChildLink.csv
  - LuParams.csv

- MigrateList.csv
- MigrateListQueryFormats.csv
- RefList.csv

## 5.6 Upgrade Flow – Error Handing

- If the upgrade flow fails, re-run the flow, and re-verify the objects as specified in the previous step. Re-copy the backed-up objects in case the rerun of the flow empties these objects.

## 5.7 Optional – Supporting a Vertical Execution Mode

### 5.7.1 Upgrade the LUs to Support a Vertical Execution

- Run the UpgradeFabricTDMRootPopulation flow to upgrade the FABRIC_TDM_ROOT LU table's population in all LUs.
- The upgrade population is required for supporting a vertical execution of the LUs.
- Redeploy all the LUs.

### 5.7.2 TDM Portal – Update the Execution Mode in the Business Entities

- Open the Business Entity window for the Business Entities for which you need to set the execution mode to a vertical execution and update the execution mode.

## 5.8 Additional Steps – the Current TDM Version is Older than 8.1

### 5.8.1 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. It is therefore required to open and resave the TDM tasks with the Parameters selection method after upgrading TDM and before executing the TDM tasks.

### 5.8.2 Rerun an Extract Task to Repopulate the LUs Parameters' Tables

- The upgrade script **updates the <LU name>_params table and it is based on the task_execution_entities**. By default, the task_execution_entities table contains executions of only the last 7 days (=0.25 month). Consequently, **the <LU name>_params table will also contain the entities of only the last 7 days of 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 execution history longer than the last 7 days, rerun an Extract task on a large population, after the TDM upgrade process has been completed, in order to repopulate the missing <LU name>_params records.

### 5.8.3 Manual Updates

- Open the **TDMFilterOutTargetTables** Actor and add the Boolean column - **generator_filterout** - if it is missing. Set it to **true** for all the TDM product tables and the _TAR table.
- Open **CustomLogicFlows** Actor and add the Boolean column - **DIRECT_FLOW** - if it is missing. 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 **TDMDBUpgradeScripts** flow.

## 5.9 Deployment

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

# 6 Optional Settings

## 6.1 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), 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'.

## 6.2 Change the Parameters mode to Parameters Coupling

- From TDM 9.1 onwards, when the selection of an entity subset for a TDM task is based on business parameters, it can be based on the newly added mode - *Parameters Coupling*.
- Click here for more information about the Parameters Coupling mode.
- The following steps are needed if you wish to set the parameter's mode to parameters coupling:

### 6.2.1 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.

### 6.2.2 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.

### 6.2.3 Update the LU Implementation

- Verify that the linked fields are defined as either PKs or unique indexes in the parent LU table for supporting the MDB export of these tables. All parent LU table's PK/unique index fields

must be linked to the child LU table. This is required when creating the FK relation in the PG DB for the exported LU tables.

- 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 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 in the regular mode, in the Parameters Coupling mode you cannot define an SQL query to get a parameter in the LuParamsMapping Mtable.
- This new table with the calculated parameters should be added to the **TDMFilterOutTargetTables.actor** in a way that it would be excluded from creating the load, delete, and data generation flows for it.

  Verify that all the LU tables in the LuParamsMapping are linked to parent tables. This is required when 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 References LU.

### 6.2.4 Rerun the Extract Tasks

- Re-extract an entity subset for each Business Entity (BE) as a means 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.