



## TDM (TEST DATA MANAGEMENT) UPGRADE PROCEDURE TO V9.5.X

- This document describes the following:
  - How to upgrade TDM to the current version - 9.5.x
  - How to reimplement the modified product's features.

### Notes:

- This document does not cover the Fabric server topology changes, such as the addition of nodes, data centers, changes to replication factors or consistency levels.
- 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 9.5.X Installation — Prerequisites

- Upgrade to Fabric version 8.3.4 or higher.

### 2 Related Documents

- [FABRIC UPGRADE PROCEDURE TO V8.3](#)
  - Note that Step 1 of the Fabric Upgrade Procedure document is irrelevant for a TDM project as the latter does not contain the iidFinder process.
- For more information about the TDM V9.5.X installation, please read the TDM Installation article in the [TDM Configuration](#).

### 3 TDM Upgrade Steps

- Fabric Upgrade
- Upgrade TDM Code (TDM Library)
- TDM Project and DB Upgrade
- Catalog Setting — Rebuilding Artifacts
- Project Implementation Changes — Optional Manual Updates
- TDM Portal Setup — Current TDM Version is Older than 9.X
- Optional Features Setup



# TDM UPGRADE PROCEDURE

## 4 Fabric Upgrade

- Upgrade the Fabric version to 8.3.4 or higher.
- If your current Fabric version is older than 8.0, perform the following:
  - Before upgrading the project, edit the config.ini file and uncomment the PACKAGE\_NAMES\_CLASS\_LOADING\_FILTER parameter. Set it as empty to disable the isolation feature between Fabric's dependencies and the project's dependencies.
  - 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.](#)

## 5 Upgrade TDM Code (TDM Library)


- Development Environment — Web Studio
- Development Environment – Desktop (.NET) Studio
- Non-Development Environment

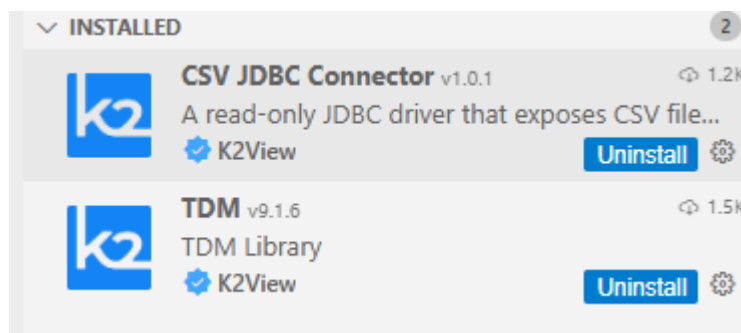
### 5.1 Development Environment — Web Studio

- TDM Extension Installation
- VSIX File Installation

#### 5.1.1 TDM Extension Installation

##### 5.1.1.1 The Source TDM Version is Installed via Extension

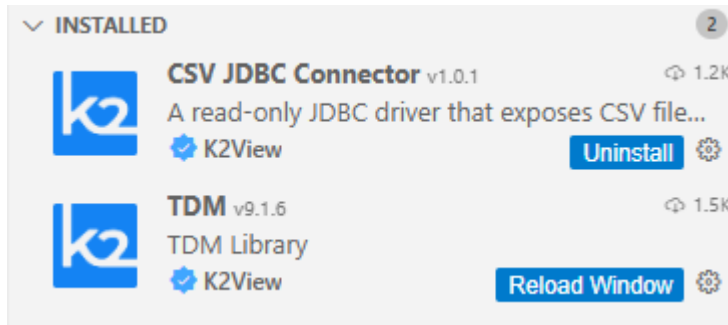
- Open the TDM project in Fabric Studio. The Fabric Studio will upgrade the code to the latest Fabric version.
- Click the Extensions icon (  ), select the TDM under the INSTALLED list and uninstall the old TDM version:




- Click the Reload Window button next to the uninstalled TDM version:



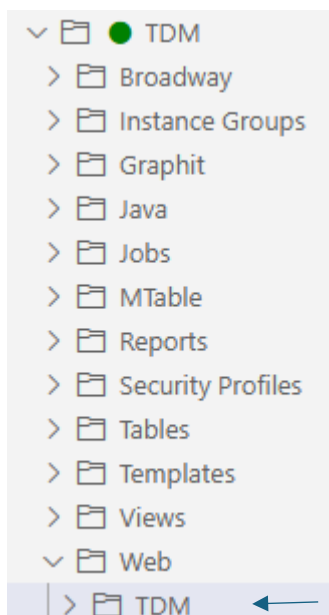
# TDM UPGRADE PROCEDURE




- Click the Extensions icon —  — and select TDM to install the TDM V9.5.x library. **Override the existing objects.**
- Redeploy TDM and the TDM\_TableLevel LUs.

## 5.1.1.2 The Source TDM Version is not Installed via Extension

- Open the TDM project in Fabric Studio. The Fabric Studio will upgrade the code to the latest Fabric version.
- Delete the TDM subfolder located under the TDM/Web folder in order to delete the current version of the TDM Portal from the TDM LU **before** clicking the Extensions icon:




- Note that you **must not delete the entire TDM LU** from the project if you wish to **keep the history of previous task executions in your development environment**. Deleting the TDM LU from the project would delete it from Fabric and remove the execution history stored in the local Fabric.
- Click the Extensions icon —  — and select TDM to install the TDM V9.5.x library. **Override the existing objects.**

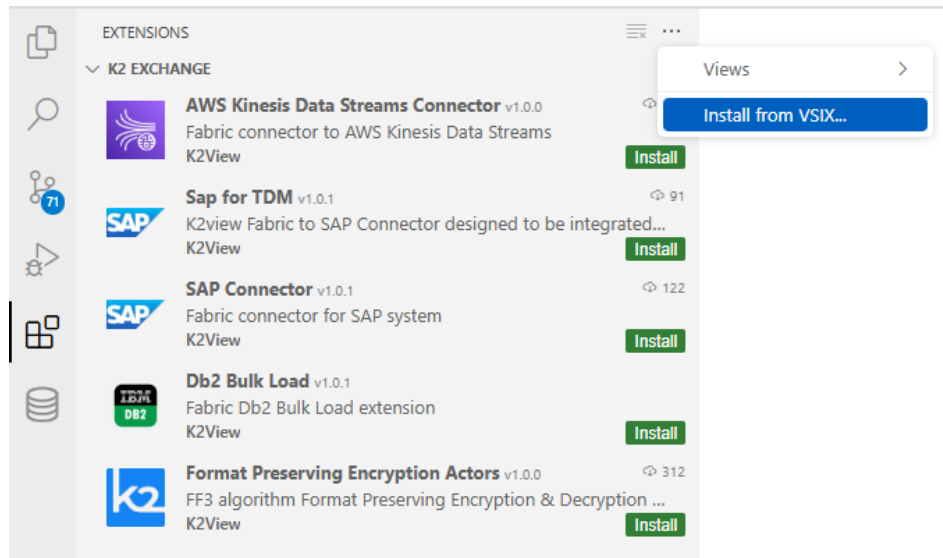


# TDM UPGRADE PROCEDURE

- Redeploy TDM and the TDM\_TableLevel LUs.

## 5.1.2 VSIX File Installation

- If access to the Extensions is not available, install the VSIX file. Download the VSIX file from the download page and upload it to the TDM project: Right-click the project-resources from the Project tree > Upload Files... > select the TDM VSIX file.
- Click the Extension icon —  — and then click the ... icon (top-right of the pane) and select the 'Install from VSIX...' option:



- A pop-up window opens: Select the uploaded TDM VSIX file and click the Install from VSIX button. Override the existing objects in your project.
- Redeploy TDM and the TDM\_TableLevel LUs.

## 5.2 Development Environment – Desktop (.NET) 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 in Fabric Studio. The Fabric Studio will upgrade the code to the latest Fabric version.



# TDM UPGRADE PROCEDURE

- If your current TDM version is older than 9.0, follow these steps:
  - Copy the following .jar files into the \K2View Fabric Studio\Projects\\lib folder:
    - json-20231013
    - handlebars-4.3.0
    - cron-utils-9.2.1
    - commons-lang3-3.11
  - Note that the commons-lang3-3.11.jar file is needed only for the upgrade flows. Remove it from the lib folder following completion of the TDM upgrade process as the TDM 9.X code no longer uses the StringUtils class.
- Manually delete the following:
  - TDM LU
  - TDM\_LIBRARY LU
  - If your current TDM version is 9.0 or older, delete the TDM\_Reference LU. Else, delete the TDM\_TableLevel LU.

## Step 3 – Import the TDM Library into the Project

- Import the TDM LUs export file into your project using the 'Import All' option for importing the following LUs:
  - TDM
  - TDM\_LIBRARY LU
  - TDM\_TableLevel LU
- Import the **Web Services** into your project using the Custom Import... option.
- Import the following **Shared Objects** into the Fabric project using the Custom Import... option:
  - Templates
  - Broadway
  - Java
- If your current TDM version is 9.0, use the Custom Import... option to import the TableLevelDefinitions.csv **MTable** into the project under the **References** LU.
- Note that the TableLevelInterfaces.csv will be updated by the TDM deploy flow.
- Optional — AI Interfaces:
  - If you would like to add the AI-based generation configuration to the TDM project, perform the following steps:
    - Import and edit the AI interfaces to the TDM project in the Studio.



# TDM UPGRADE PROCEDURE

- Add the AI environment to the Studio.
- Click [here](#) for more information about AI implementation.

## Step 4 – Additional Steps – Needed if the Current TDM Version is Older than 9.X

- A new Global has been added to TDM V8.1 for masking and sequence handling — 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 V9 sets the **POSTGRESQL\_ADMIN** as the **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 steps:
    - 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.
- Removing duplicate TDM objects from the project’s Shared Objects: The import process creates duplicate objects in the project since TDM V9 stores TDM objects in subfolders within the **Shared Objects’** Broadway and Java folders. The duplicated objects will be removed in the next step, i.e., running the UpgradeTDMProjectToTDM9 flow.

## 5.3 Non-Development Environment

- Prerequisite: The upgraded TDM project must be committed to GitHub.
- Pull the updated TDM project from GitHub

## 6 TDM Project and DB Upgrade

- General Prerequisites
- Development Environment
- Non-Development Environment

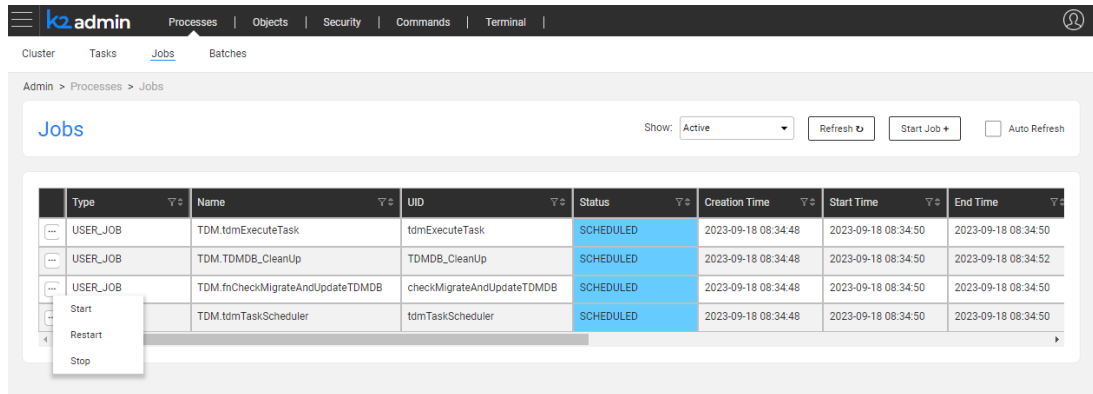
### 6.1 General Prerequisites

- If your current version is 9.3.1, add the TDMDB\_SCHEMA Global to the TDM Shared Global files.
- Verify that the TDMDB\_SCHEMA Global is correctly defined with the appropriate value for the TDM DB schema, and the TDM interface is properly set before the upgrade.
- Open and save the Environments file in Fabric Studio and redeploy it to Fabric.



# TDM UPGRADE PROCEDURE

- Stop all TDM jobs on Fabric, if any are running, to prevent the TDM DB tables from being locked by parallel execution of the upgrade flow and TDM jobs. Use the Web Admin to stop the TDM jobs:



- Back up the TDM DB before the upgrade.

## 6.2 Development Environment

- If you have the TDMOrchestrator flow under an LU, perform one of the following actions:
  - Delete the TDMOrchestrator from your LU and use the one in the Shared Object instead.
  - Update the TDMOrchestrator flow — use the DeleteFromTarget and LoadToTarget flows instead of the DeleteAllTables and LoadAllTables flows. This update is required since the upgrade flow renames the DeleteAllTables and LoadAllTables flows to DeleteFromTarget and LoadToTarget.
- **TDMCleanUp** MTable has been moved from the **TDM LU** to the **Reference LU**. A new field, **OVERRIDE\_CLEANUP\_RETENTION\_PERIOD**, has been added to this MTable. This field overrides the default cleanup retention period for specific tables.
  - By default, it is set to **6 months** for the **TASK\_EXECUTION\_SUMMARY** and **TASK\_REF\_EXE\_STATS** tables.
  - A longer retention period is defined for these tables because their data is required for **TDM Usage Reports**.
  - You can configure a different cleanup retention period for these TDM tables if needed.
- **TableLevelDefinitionsInterfacesTypesDefaults** has been added to the **TDM\_TableLevel LU**. This MTable defines the default handling of table partitions for **PostgreSQL** and **Oracle** databases.
  - If required, you can copy these records into the **TableLevelDefinitions** MTable.



# TDM UPGRADE PROCEDURE

## 6.2.1 Current TDM Version is Older than 9.X

- Deploy all LUs of the project to Fabric.
- Use the soft deploy option to deploy the TDM LU to Fabric without starting TDM jobs or executing the TDM deploy flow.
- 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** TDM DB table.
- Deploy (regular deployment) the TDM LU to Fabric.

## 6.2.2 TDM 9.X Upgrade

- The TDM DB upgrade is initiated by the TDM LU deployment. Deploy (regular deployment) the TDM LU to Fabric. The TDM deployment upgrades the TDM DB.

## 6.3 Non-Development Environment

- Prerequisite: The upgraded TDM project must be committed to GitHub.
- Pull the updated TDM project from GitHub and deploy the updated TDM LU. The TDM deployment upgrades the TDM DB.
- Redeploy the entire project, including the project's LUs and Web Services.

## 7 Catalog Setting — Rebuilding Artifacts

- Fabric V8.3 creates a separate catalog\_field\_info.csv file for each interface. Delete the old catalog\_field\_info.csv from the project and rebuild the artifacts in the Catalog for the required version in order to add the new Catalog files to the TDM project.

## 8 Project Implementation Changes — Optional Manual Updates

- Update the load flows — populate the 'table' input argument of the 'Load Stats To TDM Table' Actor. This fix is needed in order to display tables with zero records in the 'Statistics Report' tab of the TDM execution report (ticket #43620).
- Adding a description to the business parameters — a new field has been added to the LUParams and LuParamsMapping MTables — description. The description can be added and displayed on each business parameter in the Task window when selecting an entity subset based on business parameters.
- Adding Catalog-based masking to the LU population — if LU population flows do not include the CatalogMaskingMapper Actor, it should be added to the population flows to enable Catalog masking. Get the value from the SEQ\_CACHE\_INTERFACE Global and send it to the interface parameter of the CatalogMaskingMapper Actor.



## 8.1 Current TDM Version is Older than 9.X

### 8.1.1 Optional — 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). This step 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 V9.0 onwards, the user's Fabric role is concatenated to the user name in the Tasks TDM DB table. This is required for the purpose of identifying the task creator's Fabric role and deciding 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.

## 8.2 Current TDM Version is Older than 8.1

- Open the **TDMFilterOutTargetTables** Actor and add the Boolean column — **generator\_filterout** — if it is missing. Set it to **true** for all TDM product tables and for the **\_TAR** table.
- Open **CustomLogicFlows** Actor and add the Boolean column — **DIRECT\_FLOW** — if it is missing. Leave this field clear for the existing records.  
Click [here](#) for more information about Custom Logic implementation.

## 9 TDM Portal Setup — Current TDM Version is Older than 9.X

- TDM V9 does not support selecting the Delete option together with either the Replace IDs for the copied entities option 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.
- TDM V9 changes the way 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.



# TDM UPGRADE PROCEDURE

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

### 9.1.1 TDM Portal – Resaving Tasks with Parameters Selection Method

- TDM V8.1 changed the way 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.

### 9.1.2 Rerun an Extract Task to Repopulate the Tables of LU Parameters

- 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 would 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 following completion of the TDM upgrade process as a way to repopulate the missing <LU name>\_params records.

## 10 Optional Features Setup

- Supporting the Vertical Execution Mode
- Change Fabric Storage to a Type that does not Support TTL
- Change the Parameters mode to Parameters Coupling

### 10.1 Supporting the Vertical Execution Mode

#### 10.1.1 Upgrade the LUs to Support Vertical Execution

- Run the UpgradeFabricTDMRootPopulation flow to upgrade the population of the FABRIC\_TDM\_ROOT LU table in all LUs.
- Upgrading the population is required in order to support Vertical execution of the LUs.
- Redeploy all LUs.



# TDM UPGRADE PROCEDURE

## 10.1.2 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 Vertical and update it.

## 10.2 Change Fabric Storage to a Type that does not Support TTL

- TDM enables creating tasks with a retention period (TTL) on the task's entities as a way of saving these entities in Fabric only for a limited period. 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](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'.

## 10.3 Change the Parameters mode to Parameters Coupling

- From TDM V9.1 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 should be taken if you would like to set the parameter's mode to Parameters Coupling:

### 10.3.1 Set, Create and Alter Schema and Table Permissions for the TDM User

- The Parameters Coupling mode uses the MDB export Fabric command to export the parameter information 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.

### 10.3.2 Run the UpgradeToParamsCouplingMode Flow

- The `UpgradeToParamsCouplingMode` flow —



# TDM UPGRADE PROCEDURE

- 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, where applicable.
- Adds the TDM\_BE\_IIDS LU table to the LUs.
- Updates the FABRIC\_TDM\_ROOT table in the LU — adds a PK to this table in order to support the MDB export of the LU into the TDM DB.

## 10.3.3 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 PKs/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 as it supports 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 a field in an LU table. Unlike in the regular mode, in the Parameters Coupling mode you cannot define an SQL query to get a parameter from 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 creating FKs on the exported tables during their export 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.

## 10.3.4 Rerun the Extract Tasks

- Re-extract an entity subset for each Business Entity (BE) as a means to:
  - Create LU schemas in the TDM DB and export the entities' data into these schemas.
  - Repopulate the tdm\_params\_distinct\_values in the TDM DB table.