TDM V10.0 RELEASE NOTES

We are excited to announce the launch of TDM 10, a landmark release that advances how development and QA teams provision, execute, and govern test data at scale. These Release Notes describe the new features in TDM release V10.0 and list bugs that have been fixed since the V9.5.x release.

- Certification of this TDM release is based on:
  - Fabric 8.4.x
  - PostgreSQL 17

MAIN FEATURES AND IMPROVEMENTS

1. Simplified Task Execution

Self-service execution and enterprise-grade governance at scale — without the complexity

- TDM 10 introduces **Simplified Task Execution** — a powerful new operational model that separates task creation from task execution, enabling expert engineers to define governed templates while empowering developers, testers, and business users to execute them safely and independently within a **governed self-service hub**. The result: faster test cycles, improved operational consistency, and dramatically reduced dependency on centralized TDM teams.

1.1 What's New in TDM 10

Task Execution in TDM App

- Users now work within a **centralized self-service hub** that exposes only the parameters they are permitted to modify. Locked configurations remain protected by design — reducing execution errors, operational friction, and dependency on specialized teams.

Granular Creator Controls

- Task creators define exactly which attributes executors can override at runtime — source/target environments, entity selection, filters, data versions, variables, and more. Everything else is locked by design.

Domain-Centric Task Organization

- Tasks are organized into logical groups by domain, team, or use case. No more scrolling through flat lists. Collapsible navigation puts the right task in front of the right person, instantly.

Role-Based Permission Control

- Only authorized users and Fabric roles can create or execute specific tasks. Every team sees only what's relevant to them.

Unified Execution Dashboard

- A single operations center replaces scattered per-task views. Monitor live executions, review history, download reports, and rerun previous executions — all in one place.

API Enhancements

- TDM 10 extends the execution APIs to support the new task execution capabilities, including runtime parameter overrides, permission enforcement, and other execution enhancements.
- For migration information, see the **TDM Upgrade Procedure to v10.x** and **TDM_API_Migration_Guide_v9.5_to_v10**.

1.2 By the Numbers: TDM 9.x vs. TDM 10

Capability	TDM 9.x	TDM 10
Parameter overrides in UI	✗ API only	✓ Full self-service UI
Runtime parameter overrides (API)	✗ Limited runtime overrides	✓ Extended runtime override support
Executor permission control	✗ Not supported	✓ Role-based, per-task
Rerun previous executions	✗	✓
Unified execution dashboard	✗	✓
Task grouping & navigation	✗ Flat list	✓ Logical groups

1.3 Why It Matters

- As delivery organizations scale, test data operations often become a bottleneck, slowing releases and increasing reliance on specialized teams. TDM 10 addresses that challenge by enabling governed self-service execution across hundreds of users — without sacrificing control or compliance.

2. Task's Pre and Post Execution Enhancements

- **Pre & Post-Execution Processes for Table-Level Tasks** : Pre and post-execution processes are now supported for table-level tasks, extending the capability that was previously available for Business Entity-based tasks only. Task creators can define and configure pre/post execution flows for table-level tasks, and task runners can provide input parameter values at execution time when those parameters are marked as editable.
- Enabling a selection of all or some of pre or post-execution processes and adding the selected processes to the task instead of adding the processes one-by-one.

3. TDM App UI Modernization

- All TDM modules are now migrated from Angular to React

RESOLVED ISSUES

Task Window Fixes

- Fixed a bug in task cloning where the task would fail to save if the maximum number of entities were left empty in the subset (parameters) tab. The clone operation no longer requires this field to be pre-populated.
- Fixed an issue where task affinity and maximum worker settings reverted to previously configured values instead of being reset to the environment defaults when a task was updated.
- Fixed an issue where affinity and maximum number of workers values were not displayed when reopening a task. Although the API returned the overridden values correctly, the GUI displayed the environment defaults instead.
- Fixed the task window so that the affinity and maximum number of workers values are correctly reset to environment defaults when the source or target environment is changed in the task.
- Ticket #42046 – Fixed the task window to correctly validate the selection method before execution. The Random selection method is no longer selectable if the lu_param table or LU schema tables are empty for the source environment.
- Ticket #49803 – Fixed the creation of delete-only task: set the sync mode to the default value instead of populating with null value.

- Ticket #49725 – Fixed duplicate Data Center (DC) entries in task affinity settings for DCs with multiple Fabric nodes.
- Tickets #47367 & #44096 – Added the ability to customize the display order of business parameters in tasks. Parameters without a defined display order are displayed in alphabetical order.

Task Execution Fixes

- Ticket #50003 - Fixed an issue where task execution override parameters containing spaces, commas, or other special characters in TASK_GLOBALS were parsed incorrectly, causing task execution failures or incorrect parameter values.
- Fixed an issue where the execution status was stuck in an in-progress state when a task failed. The task_execution_summary table now correctly transitions to 'failed' status.
- Fixed an issue in the VersionForLoad API that caused it to not return data versions created on the current day.
- Ticket #45659 - Fixed a primary key conflict in the TDM LU that occurred when the same table name appeared in more than one schema and both tables were selected in the same task.
- Ticket #45195 - Improved performance when populating the TDM error and statistics tables in the task execution for high-volume workloads.
- Ticket #49042 - Added the LU name to the selected pre/post Execution flow, to allow using flows with the same name in different LUs.
- Ticket #49996 – Task Affinity: Fixed an issue where, in environments with multiple nodes assigned to the same affinity, the table cache was not loaded on all nodes. As a result, the TableLevelInterfaces metadata was unavailable on some nodes, causing table-level tasks to fail for certain tables.

Table-Level Task Fixes

- Ticket #49095 - Fixed inconsistent behavior when processing empty tables in in-place masking tasks. For Extract/Extract& load task of empty tables, the tables used to be marked as completed. For in-place masking - the tables used to be marked as failed. Now the tables are marked as completed for in-place masking tasks as well.

- Do not enable running a delete before load on processed tables when the source and target environments are identical. The delete is blocked to prevent the deletion of the table before its extraction.
- Ticket #49786 - Fixed an issue where table-level task execution failed while preparing task data, leaving the task stuck in 'Not Started' status.
- Ticket #49847 - Fixed the failure on counting a BigQuery table.
- Ticket #49844 – Fixed a table-level task failure when running a filter on a BigQuery table and processing a custom flow to extract the table's data.
- Ticket #49784 - Fixed the table-level task execution to correctly handle field names that are reserved words in SQLite (the table is stored as SQLite table in the Test Data Store).
- Ticket #49604 – Fixed an issue where opening a task could fail if an interface was disabled in the task's target environment. The API now retrieves the table configuration from the implementation (_dev) environment.
- Ticket #48409 - Fixed an issue when the table's subsets were not correctly reset after updating the extract & load task to an in-place masking task.
- Ticket #49969 – Fixed an issue where table aliases were unnecessarily included in source queries generated by the TDM code.
- Fixed the in-place masking task window: the In-Place Masking checkbox is now disabled with an explanatory message when the selected source environment is read-only.
- Fixed the in-place masking task window: populate the selected environment in the disabled target component.
- Ticket #48782 – Fixed the main batch process for table-level tasks to correctly apply the task's affinity settings.
- Ticket #48262 – Fixed handling dynamic schemas (defined as Globals) in the task execution to check if the target schema is a dynamic one regardless of whether the source schema is also defined as a dynamic schema.
- Ticket #49782 - Fixed an issue in the Task window for Entities & Referential Data tasks where tables with a dynamic schema (Global in RefList) were incorrectly displayed with a schema value of *null*.
- Tickets #49437 and #48636 - Fixed table-level tasks to correctly support table and field names containing special characters e.g. spaces, which previously caused failures in the task execution.

Rule-Based SDG Fixes

- Ticket #47954 - Fixed the count of generated entities in the load task for pre-generated entities. The count now reflects only successfully generated entities.
- Fixed rule-based generation flows to prevent double sequence replacement by catalog masking, which previously caused MDB export failures when sequences were added to table fields via the catalog.

TDMLUInit and Template Fixes

- Ticket #38579 - Fixed the TDMLUInitBasedOnFabric flow to correctly create the Target (TAR) LU table with a single population regardless of the populations for the related source LU table.
- Ticket #48800 - Fixed the generation of relationships between target LU tables. Only a single relationship is now created between each pair of target tables, even when multiple relationships exist between the corresponding source tables due to multiple population paths.
- Ticket #36505 - Fixed TDM load and delete flow templates to replace JDBC calls with the Fabric interfaceSchema class, improving compatibility and maintainability.
- Ticket #47650 - Fixed the indentation in the YAML files generated by TDM templates, which could cause invalid flows.
- Ticket #48910 - Fixed cloning behavior when using Catalog-based sequences: the sync of a cloned instance was incorrectly generating new sequences instead of reusing the existing ones.
- Fixed TDM flow templates to correctly handle field and table names containing special characters, preventing parsing failures.
- Ticket #49635 - Added the *tableLU.population.flow.yaml.template* to the TDM templates to enable a proper generation of the LU tables population flows if the TDM project did not install the TDM extension but imported the TDM library instead.

General and Miscellaneous Fixes

- Ticket #48909 - Fixed an issue when executing a task on multiple root LUs and the entity was already reserved. The task now fails the reserved entity for all root LUs.

- Ticket #48916 - Fixed vertical execution to support instance IDs containing special characters.
- Fixed an issue when removing a system from an environment automatically deleted related tasks without displaying a warning message. Tasks are no longer deleted when the environment can be edited in the task execution or when the removed system's LUs are not root LUs in the task's Business Entity.
- Ticket #48519 - Fixed DB schema parsing by the TDM templates.
- Ticket #48183 - Restored the TDM Global variables for *enable_masking* and *enable_sequences* flags. This fix enables a correct debug of the TDM processes in Fabric Studio (syncing an LU instance for example), since these flags are treated to be 'true' by Catalog Masking if they are not defined.
- Ticket #49076 - Fixed API descriptions and request payload examples for task-related APIs.
- Ticket #49042 – Fixed an issue where tasks could execute the wrong pre- or post-process when multiple LUs used the same process name. Pre- and post-processes are now mapped to their associated LU in the Business Entity window.
- Upgraded outdated third-party JAR files to address known security vulnerabilities.