



TDM V9.5.0 RELEASE NOTES

These Release Notes describe the new features in TDM release V9.5.0 and list bugs that have been fixed since the V9.4.x release.

- Certification of this TDM release is based on:
 - Fabric 8.3.4
 - PostgreSQL 17

MAIN FEATURES AND IMPROVEMENTS

1. TDM Usage Reports

- New **TDM Usage Reports** are now available via the **TDM App** and the **Reports web application**, providing comprehensive visibility into how TDM is used across your organization.
- These powerful reports deliver actionable insights into:
 - Task creation and execution trends
 - User activity and adoption
 - Environment utilization
- In addition to the out-of-the-box reports, you can build **customized reports** tailored to your specific needs and access them through the Reports web application.
- Together, these capabilities enable organizations to **optimize resource usage, enhance operational efficiency, and maximize ROI** from their TDM investment, turning raw usage data into smarter decisions and measurable business value.

2. Affinity and Worker Configuration for Task Execution

- TDM now supports configuring **affinity** and/or **number of workers** at both the **TDM environment level** and the **task level**. These settings provide greater control over resource allocation and workload distribution.
- This feature applies to:
 - Entity-based tasks
 - **Table-level** tasks, with affinity applied as follows:
 - Extract-only processes – run on the source affinity
 - Extract & load processes – run on the target affinity
 - Delete processes – run on the target affinity (Ticket #46645)



3. Table-Level In-Place Masking

- Introduces **in-place masking** for table-level tasks. Unlike regular table-level tasks, which delete and reload the entire table or load data into a target table, in-place masking updates PII fields **directly within existing records**, without deleting or reloading the table data.
- This capability allows masking sensitive data while preserving existing records and non-PII content.

4. Table Concurrency Processing

- Significantly improved performance of table-level tasks by enabling parallel processing of table partitions, reducing execution time for large tables

5. Additional Table-level Improvements

Ticket #46758 – Retrieving the Tables List in the Task Window – Performance Improvement

- When creating table-level tasks, the tables list is now filtered to include only tables that belong to the selected interface and schema, improving responsiveness and usability.
- To search for tables across all interfaces and schemas, the legacy behavior can be preserved using the new configuration parameter, **TABLES_USE_SPLIT_API**, added to the **TDM_GENERAL_PARAMETER** table in the TDM database:
- By default, this parameter is set to **true** (new behavior).
- Set it to **false** to preserve the legacy behavior.

Task Window - Table Count Control

- TDM App users can now disable table row count calculation at the task level, improving task execution performance when row counts are not required.

Commit Size Configuration

- The commit size can now be defined at the **interface, schema, or table level**, overriding the default value defined by the **TDM_REF_UPD_SIZE** global parameter (1,000 records per commit).

Enhanced Error Handling for Task Execution

- Error handling has been enhanced to allow task execution to **ignore predefined database errors**—such as unique constraint violations—while continuing to process subsequent records without interruption. In addition, **successfully processed records**



TDM RELEASE NOTES

are committed to the target database even when some records fail, preventing a full task rollback due to partial errors.

- Important note: In databases that automatically abort a transaction when an error occurs (for example, PostgreSQL), further processing within the same transaction is not possible. As a result, the affected table partition is not loaded.

6. TDM Template Enhancements

Tickets #46534 and #47650 – Catalog-Based Target Field Resolution

- TDM templates now retrieve target fields directly from the catalog instead of the Fabric Logical Unit (LU). This change was introduced following issues identified in certain data sources (such as Aerospike) and applies to **all supported data sources**.
- This enhancement ensures correct handling of data sources with **case-sensitive field names** and improves overall consistency and reliability of template resolution.

Multiple Documents per Logical Unit

- TDM now supports multiple documents within a single Logical Unit, increasing flexibility for document-based data sources.

Ticket #46534 – Multi-Schema Support in a Single Run

- The TDMLUnitBasedOnFabric flow now supports adding TDM setup to an LU that spans multiple interfaces and/or schemas in a single execution.

Miscellaneous

- Task execution monitoring – Added an information icon with a link to the Batch Monitor window next to each processed table.

7. TDM App UI Modernization

- The following modules were migrated from Angular to React:
 - Data Centers
 - Permission Groups
 - Systems
 - Reserved Entities
 - Business Entities
 - Environments.

8. Connector Fixes

- Ticket #47555 – Fixed the **MongoDB**, **Couchbase** and **Cosmos** connectors to handle Documents without complex fields.



9. SDG Based on AI — Improvements

Special Fields and Key Column Generation

- Added full training and generation support for **Special Fields, Primary Keys (PKs), Foreign Keys (FKs), Unique Identifiers (UIs)**, including composite keys.
- These columns are now generated using dedicated ML models and algorithmic logic, preserving structural integrity even in complex schemas.

Enhanced 1:1 Logical Relation Handling

- Improved automatic extraction of 1:1 logical relation directly from source data.
- This enhancement eliminates the need for separate, manually defined mapping fields and simplifies overall configuration.

Removal of Manual Mapping Fields

- Manual 1:1 mapping parameters have been removed, as mappings are now derived automatically by the system.

Updated Execution Parameters

- New training and generation parameters have been introduced to enable fine-tuning of these processes:
 - Early stopping behavior for logical relations training
 - Range tolerance for special and key columns
 - Uniqueness handling during data generation

10. Miscellaneous

Task Update Behavior Enhancement

- **Tasks can now be updated without creating a new task version** (task_id). This reduces task duplication and enables a unified execution history across task updates. Legacy behavior can be preserved using the new configuration parameter: **CREATE_NEW_TASK_VERSION_ON_UPDATE**, added to the **TDM_GENERAL_PARAMETER** TDM DB table. By default, this parameter is set to **false**. Set it to **true** to create a new task version (task_id) on each update.

TDM Cleanup Enhancement

- Enables overriding the default cleanup period for selected TDM DB tables, allowing different cleanup periods per table. A longer cleanup period has been configured for **TASK_EXECUTION_SUMMARY** and **TASK_REF_EXE_STATS** to support TDM reporting.



TDM RELEASE NOTES

Environment and Task Variables Enhancements

- Tickets #45938 and #47609 – Global variables defined as **final** are now **excluded** from the list of available variables in both the **Environment Variables** and **Task Variables** screens.
- Ticket #46726 – The **final** definition has been **removed** from the following Global variables to **allow overriding their values** at the **Environment** and **Task** levels:
 - COMBO_MAX_COUNT
 - MASKING_CACHE_TTL
 - TDM_REF_UPD_SIZE
 - SEQ_CACHE_INTERFACE
- Task variables – Variable names are no longer truncated. The full name is now displayed when hovering over the ... icon, and copying the entire name is supported.

RESOLVED ISSUES

- Ticket #48010 & #48036– *Task window*: Fixed the data type definition for a numeric business parameter with a zero value (**0.0**), changing it from **Text** to **Numeric**.
- Ticket #47461 – Fixed an issue in the summary execution report for table-level tasks. Corrected the population logic of the task_ref_exe_stats table in the TDM LU.
- Tickets #47391 & #47921– Improved task execution performance. The population of the statistics table (task_exe_stats_detailed) has been optimized to check the entity status once per entity (for all related records) instead of validating it for each individual record.
- Ticket #47264 –
 - **Support for "Reserve Only" Permission Set**: Fixed handling of the reserve-only permission set on TDM environments.
 - **Entity List Reservation**: During execution, entities that are already reserved will now fail with an appropriate error message, instead of being silently filtered out.
- Ticket #47660 – Fixed the *SourceDBQuery* TDM Actor to skip validation of the LU table name when both the LU schema and LU table inputs are empty.
- Ticket #47051 – Fixed the TDM extension to support TDM templates in projects derived from a base TDM project (checked out from GitHub), without requiring installation of the TDM extension.
- Ticket #46473 – Fixed task execution to evaluate environment masking status at runtime rather than at task creation time, ensuring masking reflects the current environment sensitivity.



TDM RELEASE NOTES

- Ticket #47795 – Fixed an issue in table-level task execution where multiple tables with the same name, located in different interfaces and/or schemas, were not handled correctly. The task execution logic now uniquely identifies tables by their full context.
- Ticket #46881 – Added support for field names containing special characters in table-level tasks.
Ticket #46821 – Fixed an error in the task execution statistics flow for table names containing special characters.
Ticket #46796 – Fixed the StatsLoader actor to default to the table name from the input parameters when populating TDM statistics, instead of relying on the StatsReader output.
- Ticket #46702 – Fixed the TDM execution report for Load tasks. Previously showing only the LU list, it now also includes pre- and post-execution processes.
- Tickets #46696 & #46541 – Fixed the TDM setup flow (TDMLuInitBasedOnFabric) error occurring when a LU's table population included inputs missing from the source table columns.
- Ticket #46462 – Fixed test connection validations at task start to allow task execution to continue on unknown errors when the force parameter of the startTask API is set to true.
- Ticket #46149 – Fixed task creation to allow saving an Extract task after clearing the Target form.
- Ticket #47691 – Fixed the setting of the root entity ID (root_iid key) in Load-only tasks to ensure data consistency across LUs when masking PII fields.
- Fixed an error where tester users received an unauthorized message when opening table-level tasks, despite having permissions to create and run these tasks in the respective environments.
- Load Generated Entities Task – Fixed the list of available executions for load. The list now includes only previous generations and excludes prior reload executions.
- Task Execution Window – Fixed TDM App to allow opening the entity list for stopped executions.
- Task window - Fixed an issue where testers with permission were incorrectly shown a permission error when opening table-level tasks.
- Task Window – Fixed an issue with the calendar when modifying the reservation period for reserved entities.
- Task Window – Fixed calendar display for from/to date selection when loading pre-generated entities in Load tasks.