

# Verify V1.1.0 Release Notes

These Release Notes describe the new features in Verify release V1.1.0 and list improvements introduced since the V1.0.x release.

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

## Main Features

### Environment

- Users can now select source and target databases using Fabric-defined environments, enabling environment-aware comparisons across deployments.
- The Create Task page now exposes four separate selectors: Source Environment, Target Environment, Source Interface, and Target Interface.
- Two comparison scenarios are now supported: comparing different interfaces within the same environment (`_dev` is pre-selected by default), or comparing the same interface deployed across two different environments (e.g., Prod vs. UAT).
- Source and Target environments now include a sensitivity indicator to specify whether PII data in that environment is unmasked. This allows Verify to apply the appropriate comparison logic based on whether the data has already been masked.
- The interfaces shown in each dropdown are scoped to the selected environment; only interfaces deployed in that environment are selectable.
- Task execution uses the environments defined in the task, not the active Fabric session environment.

### Add Tables to Task

- Users can now add new table pairs to a task at runtime from a JDBC source or the Catalog, without requiring pre-configured settings.
- Source Database Only mode: select a full schema (all tables added) or pick individual tables; source and target table names are assumed to be identical.
- Pairing mode: explicitly pair a source table with a different target table, including tables from different schemas.
- Newly added table pairs default to 10 partitions (defined in a new Advanced Setting tab), which can be adjusted per pair during task creation.

### Settings Page

- A new Settings page provides a centralized interface to view, add, edit, and delete table comparison configurations, replacing the previous `k2verify_config` MTable.
- Each configuration record defines the source and target interfaces and schemas for a pair of tables.
- Advanced Edit enables per-table configuration of column mapping, transformation functions, PII columns, excluded columns, comparison keys, partition count, partition method, ignore-null columns, and excluded rows SQL filter.

- Users can download the current settings as a CSV file, or upload a CSV to add new records or update existing ones in bulk.

### Automatic Partitions count setting

- Verify now supports automatic partition count calculation per table pair when no explicit partition value is configured (partitions = null or 0), based on the number of records in the table divided by the default PARTITION\_SIZE defined in the Advanced Settings tab.
- The partitions count is resolved using the following priority order:
  - Cached value from k2verify\_table\_count
  - DB catalog statistics (e.g., pg\_class.reltuples for PostgreSQL, DBA\_TABLES.NUM\_ROWS for Oracle, SYSCAT.TABLES.CARD for DB2)
  - COUNT(\*) on the source table
  - Default of 1.
- Calculated values are stored in a new k2verify\_table\_count cache table, keyed by source interface, schema, and table, to optimise subsequent executions.
- The COUNT(\*) path has a 1-minute timeout; if not completed in time, the system falls back to a partitions count of 1.

### Error Handling

- A new error handling mechanism introduces a two-tier error model:
  - Physical Errors (infrastructure failures such as insert failures or connectivity issues) immediately stop partition execution
  - Data/Comparison Errors (e.g., datatype mismatches, null handling violations) are evaluated against configurable thresholds before stopping.
- Three threshold parameters, configurable in the Advanced section of the task screen, control whether execution continues after data errors:
  - MIN\_PROCESSED\_THRESHOLD\_PCT - minimum percentage of records that must be processed before thresholds are evaluated.
  - MAX\_COMPARISON\_FAILURE\_PCT - maximum allowed percentage of records where the comparison could not be completed.
  - MAX\_RECORD\_MISMATCH\_PCT - maximum allowed percentage of records where comparison completed but field values differ.

### Retry Mechanism

- A new Retry button in the Task History view allows users to re-execute only the failed partitions of a previous execution, without creating a new task execution.
- Retry reuses the existing execution context: the task\_execution row is updated to In Progress (no new row is inserted), failed partition records are upserted, and the failed batch ID is fetched and re-executed via batch\_retry.
- Successfully completed partitions from the original run are preserved; their results are combined with the retry results in the final report and summary.
- The Retry button is enabled only when the task has at least one previous execution with a failed partition and a valid batch ID exists. It is disabled when an execution is In Progress, all partitions succeeded, or the batch has been purged.

### Pre/Post Execution Processes

- Verify tasks can now be extended with configurable Broadway flows that run before (Pre-Execution) and after (Post-Execution) the core verification logic, without modifying the verification engine.
- Flows are tagged with `verify_pre_execution` or `verify_post_execution` in their Flow Properties; only tagged flows appear in the respective selection list during task creation.
- Multiple flows can be selected. Execution order is user-defined; flows assigned the same order value run in parallel within that group.
- If a Pre-Execution flow fails, the main verification task does not start.
- A Post-Execution flow failure does not prevent the verification report from being generated.

### CSV File Comparison

- Verify now supports comparing CSV files using Local File System, SFTP, S3 File System, Azure Blob File System, and GCS File System, loading them into LUI partitions and reusing the existing table-to-table comparison flow.
- CSV rules can be configured in Settings in the same way as a JDBC connection, defining the source and target file paths, comparison keys, PII columns, excluded columns, transformation flows, and partition count.
- Large files are split across multiple partitions for parallel processing; results from all partitions are aggregated into a single execution report.
- A wildcard pattern (e.g., `customerFile*.csv`) is supported to select multiple matching files in a single task from the source and target paths; files are matched for comparison by name, so each source file must have a corresponding file with the same name in the target path.

### TDM and API Integration

- Verify tasks can be triggered automatically as a TDM post-execution process via the `bwVerifyAsTDMPostExec` flow. Two options are available: run a pre-defined Verify task (set `EXECUTE_VERIFY_TASK_ID` to an existing task ID), or have a new task created automatically per interface execution table (set `EXECUTE_VERIFY_TASK_ID` to `NEW_TASK`).
- Three API endpoints are available to manage Verify tasks programmatically: `wsK2VerifyStartTask` to initiate execution, `wsK2VerifyMonitorTask` to track progress using the Execution ID, and `wsK2VerifyAgentDownloadResource` to retrieve the completed report.

### Extended Datatype Support

- Verify now supports BLOB and CLOB field comparison across all databases available in Verify

### New Advanced Tab

- A new Advanced Settings tab is available under Settings, providing global parameters that control error handling and partition management across all tasks.
- The following parameters are configurable in this tab:
  - `MIN_PROCESSED_THRESHOLD_PCT` (Error Handling) - Minimum percentage of records that must be processed before evaluating other thresholds.
  - `MAX_COMPARISON_FAILURE_PCT` (Error Handling) - Maximum allowed percentage of records where the comparison between source and target could not be completed, for example due to missing records or processing errors.
  - `MAX_RECORD_MISMATCH_PCT` (Error Handling) - Maximum allowed percentage of records where the comparison succeeded but field values differ between source and target.

- PARTITION\_SIZE (Partition Management) - Partition size used to determine the partition count when Partitions is set to 0 or null. Verify estimates the source table row count and divides it by this value to determine how many comparison partitions to create. Range: 0-1,000,000.
- DEFAULT\_PARTITION\_COUNT (Partition Management) - Default number of partitions used for comparison when partitioning is not explicitly defined. If set to Auto, Verify automatically determines the optimal partition count based on data size and system characteristics. A numeric value overrides automatic calculation and enforces a fixed number of partitions. Range: Auto or 1–1,000,000.