FABRIC V8.3.0 RELEASE NOTES

These Release Notes describe the new features in Fabric release V8.3.0 and list bugs that have been fixed since the latest V8.2 release.

Certification of this Fabric release is based on:

- Cassandra version 4.1.3
- SQLite version 3.46.1
- OpenJDK Runtime Environment 21.0.4+7
- Confluent Kafka version 7.9.2
- Neo4j enterprise 5.26.10
- Elasticsearch 8.5.3
- AWS OpenSearch 1.3.4
- PostgreSQL 17.5

MAIN FEATURES AND IMPROVEMENTS

1. Fabric Catalog

The Discovery & Catalog solution has been enhanced with several powerful features, such as:

- ✓ The ability to create a Catalog from either files or HTTP requests.
- ✓ The ability to create a bulk of Catalog nodes and edit this bulk (e.g., add new property to all bulk entries simultaneously).
- ✓ Additional Catalog plugins with new capabilities, such as the creation of logical relations between datasets, analysis of source data and calculation of various metrics.

Below is a detailed list of the new features:

- **File Cataloging**: Catalog can now be built based on files or HTTP requests. Discovery can be performed by using either the file's metadata definition, sample data or both.
    - The Crawler framework used for file cataloging employs a generic mechanism that is independent of specific file format. It expects to get an input in a predefined format, achieved through the invocation of the 3 predefined Broadway flows: *Get Metadata*, *Get Files List* and *Get File Data*. These flows should be created in the project's implementation and attached to the project interface.
    - The *File Cataloging - Demo* extension (available in K2exchange) provides an illustration of the E2E process of file cataloging and masking the files data. The extension includes the example flows for CSV, XML, JSON, Avro and HTTP formats.

  https://support.k2view.com/Academy/articles/39_fabric_catalog/05_cataloging_of_files.html

- **Bulk creation & edit:** The Catalog Application now supports new manual activities: bulk creation and editing of the Catalog entities. These capabilities allow users to efficiently update multiple Catalog entities, by adding or editing various properties simultaneously (for example, assigning a *Sequence Name* property to multiple fields).
    - Bulk creation is initiated by selecting the required Catalog entities (such as fields) from the Search Catalog screen. The selected bulk is stored in the session storage and remains available until the browser session is closed.
    - Bulk editing is enabled only when the Catalog is in edit mode. However, bulk creation and viewing are possible in any Catalog mode.

  https://support.k2view.com/Academy/articles/39_fabric_catalog/catalog_app/14_1_bulk_creation.html

  https://support.k2view.com/Academy/articles/39_fabric_catalog/catalog_app/14_2_bulk_edit.html

- **Logical relations plugins** were enhanced to provide more ways to identify possible logical relations between the datasets.
    - **Reference by Query Analysis** is a new plugin that analyzes a list of SQL queries provided as an input file. By examining the JOIN operations in the queries, the plugin can create foreign key (FK) relationships between datasets. Query parsing and analysis are performed using either an internal Fabric parser or an LLM.
    - **Reference by Data Comparison** is a new plugin that examines data within the data source fields to identify correlations using the probabilistic Bloom filter algorithm. Based on the analysis results, the plugin can establish FK relationships between datasets.
    - The existing Metadata Logical Reference plugin has been renamed to **Reference by Name Comparison**. The rule for creating links between two non-PK fields across datasets has been removed.

  https://support.k2view.com/Academy/articles/39_fabric_catalog/plugins/05_relations_creation.html

- **Source data analysis plugins** were enhanced to provide more statistical and analytical data that can be used to improve the quality of masking and synthetic data generation:
    - **Data Quality Metrics** is an existing plugin that has been enhanced with additional statistical metrics calculations, such as min, max, average and standard deviation.
    - **Option Set Analyzer** is a new plugin that identifies fields with a limited number of distinct values (in the data sample). The identified distinct values are then saved into a *catalog_field_option_set___<dataPlatform>_<schema>* MTable.

  https://support.k2view.com/Academy/articles/39_fabric_catalog/plugins/04_source_data_metrics.html

- **Manual override of the relations** has been enhanced with the following:
    - It is now possible to create cross-schema / cross-data platform relations manually (when the Catalog is in edit mode).
    - It is now possible to create a relation from any field of the parent table, and not necessarily from the primary key.

  https://support.k2view.com/Academy/articles/39_fabric_catalog/catalog_app/07_manual_overrides.html

- **Catalog artifact** has been enhanced with the following:
    - The artifacts of the **relations** can now be created when needed. This feature is only available via the */api/catalog/{version}/build-catalog-artifacts* API, by setting the input parameter *refersTo=true*. The relations artifacts are created as a new **catalog_relations_info MTable**.
    - The artifacts (both fields and relations) are now **split by default** into separate files, per data platform and schema, to reduce the time required to create each artifact. This is done when SPLIT_CATALOG_ARTIFACTS = ON, which is now the default parameter setting (in the config.ini file). Click here to get more details related to this feature, when upgrading your project to Fabric V8.3.

  https://support.k2view.com/Academy/articles/39_fabric_catalog/catalog_app/09_build_artifacts.html

  https://support.k2view.com/Academy/articles/39_fabric_catalog/20_catalog_APIs.html
- Databases with 2 or 4 levels of hierarchy:
    - The Catalog hierarchy has 3 levels: data platform, schema and dataset.
    - Some databases might either have no schema level or have an additional hierarchy level (called a catalog or a database) that can group several schemas.
    - Catalog supports these data sources in the following way:
        - When the schema level is missing or empty (e.g. SpannerDB), a schema is created with the name “__default” (containing 2 underscores). Note that for SQLite databases, the Catalog keeps “main” as a schema name.
        - When there is an additional hierarchy level (e.g. Snowflake DB), it is reflected by a new property **catalogName**, created on a schema node.

2. Native Support for NoSQL Document Storage

Fabric can now provide native E2E support for NoSQL Document Storage (such as MongoDB or CouchBase), which includes the following:

- A Discovery job can run on interfaces such as MongoDB or CouchBase, once the respective K2exchange connector has been installed in the project. The Catalog is then created based on the discovered document hierarchy. (This feature was already supported prior to V8.3).
- The Web Studio’s Interface explorer can now present the Document’s complex structures such as nested hierarchy levels and arrays of primitives.
- Logical Units can now be created based on the Document’s metadata retrieved from the Catalog, after discovery has run on it.
    - The nested hierarchy levels are then created as LU tables with a referential link to their respective parent level.
    - When a complex structure on any level has a non-unique name, the names of the parent levels are concatenated to it, after 3 underscores. E.g. *emailInfo___emergencyContacts*.
- LU tables are created with the following system-generated fields:

    - *_docId* is a unique ID added to each LU table. Its purpose is to uniquely identify the row of an instance, when splitting the document and composing it back.
    - *_parentDocId* is added to all LU tables except for the root table, and it is used for creating the referential link from the nested structure to its parent structure.
    - *_value* is only added to LU tables that represent an array of primitives, to keep the element's value.
    - *_docHints* is a field used internally by Fabric to manage the composition of the original Document.
- Root table population is the only population that reads data from the source document.
- The populations created for such LU tables include a dedicated DocumentQuery Actor that is responsible for generating unique IDs to maintain the relations between hierarchy levels of the Document. In addition, the Actor takes the respective part of the original Document and breaks it into fields that are then populated in the LU table.
- The TDM V9.4 templates have been enhanced to support LUs that are generated based on NoSQL documents and enable loading the masked documents to the target environment:
    - Adding the TDM setup to these LUs.
    - The TDM delete flow is enhanced to delete these documents from the target system.
- The TDM load flow is enhanced to assemble the document structure of a LU representing a document (complex type) using the DocumentAssemble Actor and load the document to the target environment.

https://support.k2view.com/Academy/articles/03_logical_units/22_native_support_for_NoSQL.html

3. Broadway

New Actors

- **JsonSchemaToMetadata** is a new Actor that can be used to transform a JSON schema file into the Catalog metadata format. The Actors can be used when creating the "Get Metadata" flow as part of **File Cataloging** solution.
    - The Actor identifies the "title" keyword to use as the dataset name and processes the "properties" keyword to extract dataset fields. Currently, it does not support advanced features such as "$ref" for references, "$defs" for reusable components, or other complex schema structures. These capabilities are planned to be supported in the future.
- **ObjectSerialize** is a new Actor, that builds a binary stream from objects, using either Kryo or the Java Object Stream serialize system.
- **ObjectDeserialize** is a new Actor, that parses of a binary object stream serialized by the ObjectSerialize actor.

Existing Actors

- The **CatalogMasking** Actors default mapping behavior has been updated as follows:

  - If the catalog_field_info MTable does not exist, the value is passed ‘as is’ without masking.
  - If the catalog_field_info MTable exists but the field is missing, its value is passed ‘as is’ without masking.
- The **XmlParser** Actor has been updated as follows:
  - A new input parameter **trim** has been added, to define whether whitespace should be removed from XML document’s values.
  - The existing input parameters **valueField** and **attributesField** of the **XmlParser** Actor and **XmlStringify** Actors are now defined as a drop-down lists, and include the following values:
    - The **valueField** parameter includes “___value___”, a standard value wrapper used by the Catalog and Catalog masking process.
    - The **attributesField** parameter includes “___attributes___”, a standard attributes wrapper used by the Catalog and Catalog masking process.
- The useEnvironment, useInstanceId, and useExecutionId fields are hidden as input parameters in the Catalog Masking Actors to avoid confusion, as the Actor’s internal logic retrieves these values from the Catalog.
- Removed Uniform range and Normal distribution options from the RandomDistribution Actor when generating random strings, as these distribution types are not applicable to string generation and were causing user confusion.
- **Enhanced SSH Actor Error Handling in Broadway Automation.** The SSH Actor in Broadway has been enhanced to execute shell commands more accurately. Previously, any output to STDERR would cause the Actor to raise an exception, even if the command was completed successfully (exit code 0). This behavior led to false positives, particularly in automation workflows that rely on diagnostic messages being sent to STDERR. Exceptions are now raised only if the SSH command returns a non-zero exit code.

4. Web Studio

- The Interface Explorer tab displays databases with 2 or 4 hierarchy levels as follows:
  - When the schema level is missing or empty (e.g. SpannerDB), a dummy schema name “__default” (containing 2 underscores) is displayed. Note that for SQLite, “main” is displayed.
  - When there is an additional hierarchy level (e.g. Snowflake DB), it is only reflected by a tooltip over the schema node, displaying the Catalog Name.

5. Miscellaneous

- A new feature has been added; it validates during the sync process whether a user is authorized to view LU instance data based on their role, security profile and defined root table view, and throws an unauthorized exception if access is denied. The Search command has been enhanced to return only the IID when the user is unauthorized. The Web Service/GraphIt functionality has been enhanced to consider security profile definitions.
- **BATCH command** has been enhanced with the support of **binding parameters**. You can utilize the BATCH command by setting up one of the following new input parameters:
  - **BIND_PARAMS** — a list of binding parameters used in the SELECT query on the source interface.
  - **BIND_PARAMS_SEPARATOR** — a separator between parameters in BIND_PARAMS (optional). The default separator is ','.
  - **BIND_PARAMS_JSON** — a JSON-format array with binding parameters used in the SELECT query on the source interface.
- **SET IS_STUDIO** is a new Fabric command that determines whether the Fabric instance is Dev (includes Studio) or None-Dev (without Studio). The command is read-only and returns either true or false.
- **LIST CONFIG_OVERRIDES, LIST CONFIG_OVERRIDES_HISTORY** are new Fabric commands that list the user overrides of the Fabric configuration.
  - **LIST CONFIG_OVERRIDES** presents a list of current overrides.
- **LIST CONFIG_OVERRIDES_HISTORY** presents the history of overrides, including the version, date of change, user, node ID and the new value.
- **MDB Export and Import to/from file**: A new enhancement to the **existing** MDB_EXPORT and MDB_IMPORT **commands** now enables exporting an LUI to an SQLite file instead of a database. While the current capability supports moving LUIs between environments via Postgres interfaces, this new enhancement enables export and import using file-type interfaces (e.g., S3, local FS, Azure, GCS, SFTP). This enhancement allows teams to ‘freeze’ an LUI for regression testing or debugging, eliminating the need to depend on a live source database or worry about data changes. The command syntax is similar to the existing syntax.
- **Secret Managers**
  - Fabric now enables specifying secret parameters as a query string. This removes the need to follow the naming convention, which Fabric assumes, for some secret managers.
- Fabric now supports “secret zero” mode for any secret in the config.ini file. This mode expands the capability introduced in Fabric V8.2.2 release, which previously supported only specific secrets.

- **Oracle JDBC driver** was upgraded from 19.3.0.0 to 23.8.0.25.04.
- The SFTP interface port is now fully integrated with the Secret Management service.
- Resolved 3rd party security vulnerabilities.

6. Auditing Capability (New for K2cloud SaaS Customers)

A new auditing feature is now available for K2cloud SaaS customers, providing enhanced visibility into user and system activities within Fabric. This feature enables organizations that leverage K2view's K2cloud SaaS services to capture a detailed log of operational actions—supporting security, compliance, and operational analysis objectives.

Key capabilities:

- Captures logins, user and role changes, deployments, web service calls, and executed Fabric commands.
- Provides actionable traceability through a structured audit record, including timestamps, protocols, parameters, user identities, session IDs, and execution results.
- Supports export of audit logs in CSV or text format directly from the Fabric Monitoring UI.
- Logs are filterable in real time using the Fabric Monitor search interface (AUDIT keyword).
- Tailored for production use; auditing is not recommended in development environments, especially when using Web Studio as it performs actions on behalf of users.

Note:

This capability is available only in **K2Cloud SaaS environments** and is not supported in self-hosted installations.

For a complete list of audited actions and detailed examples, refer to the K2View Auditing Overview.

RESOLVED ISSUES

- Ticket #42080 — it was previously possible to add 'pii' and 'classification' properties incorrectly (e.g., in capital letters), which caused confusion as they appeared similar in the UI but were not recognized by the system; manual entries in the Catalog PII Edit section are **now** prevented and if a user **now** types 'pii' and 'classification' in any case (capital or camel case), they are automatically converted to lowercase.
- Ticket #42148 — Fabric Instance Group - the number of instances was not presented correctly.
- Ticket #42540 — irrelevant log messages appear when running Discovery with the MetadataFileAnalyzer plugin.
- Ticket #42733 — MTable selection is not sorted. MTables are **now** arranged in lexicographical order to make searching easier.

- Ticket #43072 — the new FabricSetRead Actor does not work when the LU is provided using a global name.
- Ticket #43192 — “Reveal in Explorer” issues have been fixed.
- Ticket #44058 — in Broadway, when selecting an Actor that includes many inputs, the focus automatically shifts to the middle of the Actor pane, whereas it should remain at the top.
- Ticket #44415 — when saving the Catalog setting failed for any reason, no error message was shown in the UI, leaving the user unaware of the failure.
- The buildAndDeployArtifacts script was enhanced to return a 401 Unauthorized error code when incorrect user credentials are provided, instead of returning error code 500.
- Ticket #42911 — in K2exchange, extension upgrades sometimes fail to work properly, requiring users to uninstall and reinstall the extensions.
- Ticket #43557 — the Discovery process that runs on Cassandra DB does not identify the data types (such as timestamp) correctly; it maps all data types to BYTES.
- Ticket #43643 — it is not possible to insert a string into a TEXT field.
- Ticket #43653 – SET Globals is not working for k2_ws. The issue has been fixed.
- Ticket #44293 – Issue with missing Globals.class when deploying LU from the Web Studio. The issue has been fixed.
- Ticket #44066 – In the Web Studio, new interface filter should only show categories with valid matches.
- Ticket #44422 – BigQuey Interface is not showed in DB command Dropdown list. The issue has been fixed.