# FABRIC V7.2.0 RELEASE NOTES

These Release Notes describe the new features in Fabric release V7.2.0 and list bugs that have been fixed since the V7.1.1 release.

Certification of this Fabric release is based on:

- Cassandra version 4.0.3
- SQLite version 3.39.2
- Open JDK version jdk-17.0.3.1
- Kafka version 3.2.0
- Confluent Kafka version 7.2.1
- OrientDB - 3.1.3-tp3 and 3.2.4-tp3
- Elasticsearch - 7.16+ and 8.5.3
- AWS OpenSearch - 1.3.4

## MAIN FEATURES AND IMPROVEMENTS

### 1. Fabric Catalog (MVP)

Introducing the Fabric's Discovery and Catalog solution which provides an insight into the Fabric interfaces, starting with the RDBMS interface types in the MVP version.

- The Discovery process includes the following steps:
  - Auto-discovery of the data source's elements (schemas, tables, fields) and the existing FK relations between them, while modeling the data source in the *neo4j* GraphDB.
  - Enrichment of the data model by creating additional relations between the data source elements when the FK relations don't exist.
  - Auto-profiling of the data model by both the metadata and data to identify sensitive information (e.g. PII marking) and classify it based on pre-defined categories.
- The Discovery results are presented in a new Catalog application (part of the Fabric Web Framework) which allows to:
  - Display the Catalog data model as a tree and navigate through its elements.
  - View the Catalog's elements properties and manually edit them, if needed.
  - Support the multiple versions of a Catalog and the versions comparison.
  - Search any node by keywords or other advanced search parameters.
  - View and modify the profiling rules.
- The Logical Unit creation is now based on the discovered and enriched data model.
- The masking can now be based on the Catalog profiling results.

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

## 2. Project Versioning

As project implementation is an ongoing process, ensuring clarity of the deployed code on the server becomes essential. Fabric provides a robust solution through tagging of project versions, allowing easy identification of the deployed version/s on the server.

The version creation process is done along with GIT tagging capability and is structured into 3 key steps across the project lifecycle:

- Version Tagging: Tag a version efficiently within the Studio.
- Fabric Object Deployment: Deploy Fabric objects, that have been appropriately tagged with a version, to the server.
- Version Assessment: Gain valuable insights into the deployed project objects' versions through the Fabric console or terminal.

https://support.k2view.com/Academy/articles/16_deploy_fabric/04_project_versioning.html

## 3. Support SQLite and Postgres as Fabric Operational DB

Until this release only Cassandra could serve as Fabric operational DB (to store history of jobs execution, etc.). Starting V7.2, it is possible to use either Cassandra, SQLite, or PostgreSQL. SQLite support as operational DB enables creating a space in the Fabric Cloud without the Cassandra installation, when SQLite can serve as operational DB and, for example, S3 or GCS as storage. The settings are done as follows:

- The new [internal_db] section has been added to the config.ini and it holds the operational DB settings. By default, it is set to Cassandra.
- The existing DEFAULT_GLOBAL_STORAGE_TYPE parameter in the [fabric] section is now set to INTERNAL_DB.
- You can either update only the [internal_db] settings, impacting both the Storage and operational DB types together, or you can define each of them to have a different DB type.

## 4. MDB Export / Import

Fabric now supports exporting or importing the MicroDB data from SQLite to another DB type (PostgreSQL). This solution allows us to backup the Fabric data, share it with others or import data from external data sources into Fabric.

Two new Fabric commands have been introduced for this purpose:

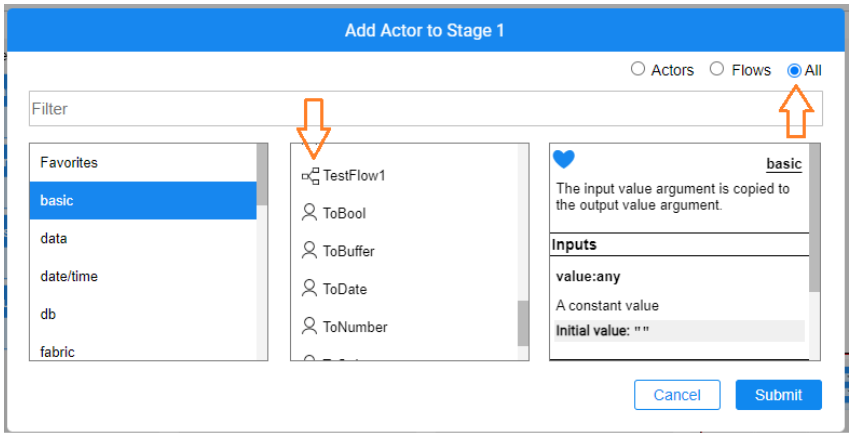
- MDB_IMPORT with IID
- MDB_EXPORT with or without IID

Both commands can optionally receive a list of tables to be excluded from the import or export process.

# 5. Broadway

## General

- The **Actor/Flow selection popup** has been enhanced with 'All' radio button. Each element has an indication (icon) to distinguish between the actors or flows.




- A new **com.k2view.regex plugin** has been added to the list of Broadway editors located in the `Server/fabric/staticWeb/editors` folder. It allows including a pre-defined list of frequently used regular expressions to any Actor, e.g. the **Regex** and **RandomRegexGenerator** Actors (see more details about the modified Actors further). Two separate lists are defined as part of this plugin:
  - List of regular expressions for **parsing** - it includes the expressions for parsing an input value, for example: Email, US phone number, US SSN, etc.
  - List of regular expressions for **generation** - it includes the expressions for generating a new value, for example: US EIN, US ITIN, US phone, Email.
  - Both lists' expressions can be edited on the project level.

## Modified Actors

- **Regex** and **RandomRegexGenerator** Actors include a **regex** input argument. The editor of this parameter has been updated to **com.k2view.regex** described earlier in this document.
  - The **regex** argument has become an editable drop-down list. Once the expression is selected, it can be edited in the Actor's input field.
  - The **Regex** Actor retrieves the list of the expressions for parsing. The **RandomRegexGenerator** Actor retrieves the list of the expressions for generation (by setting the editor's input argument isGenerating = true).

## New Actors

- **CatalogMasking** Actor - TBD

# 6. Miscellaneous

- **xxx**

## RESOLVED ISSUES

- Ticket xxx – xxx. The problem has been resolved.