FABRIC V7.0 RELEASE NOTES

These Release Notes describe the new features in Fabric release V7.0 and list bugs that have been fixed since the V6.5.9 release.

Certification of this Fabric release is based on:

- Cassandra version 3.11.14 and 4.0.3
- SQLite version 3.39.2
- Open JDK version jdk-8u322 and jdk-17.0.3.1
- Kafka version 3.2.0
- Confluent Kafka version 7.2.1
- OrientDB - 3.1.3-tp3 & 3.2.4-tp3
- Elasticsearch - 7.14.1 & 7.8.1
- AWS OpenSearch - 1.3.4

MAIN FEATURES AND IMPROVEMENTS

1. Fabric PubSub

The following changes were done to the Fabric PubSub mechanism:

- PubSub abstraction layer has been introduced for defining a unified Fabric connection to a message provider (such as Apache Kafka or JMS), which would be used by various Fabric processes connected to it. The PubSub abstraction layer is utilized by the Broadway Pub/Sub actors, Common DB and CDC.

- This unified configuration is achieved by both creating a new **PubSub Configuration** interface type and defining all the connection settings in the **[default_pubsub]** section of the config.ini.

- Memory mode enables running PubSub in-memory, using an internal queue that runs on a localhost. It can only be used for either development or debug purpose.

- The setting of the transaction mode determines how the publisher handles the transactions. The 3 supported modes are: **ASYNC**, **BROKER**, **IGNORE**.

- New **publisher** and **subscriber UserCode** methods were exposed to publish and subscribe messages via the PubSub Configuration interface. They can be invoked as follows:

    try (IoSession session= pubsub("section_name")) {

    session.publisher(Util.map()).send(msg,Util.map("TOPIC", topic));

    }

https://support.k2view.com/Academy/articles/24_non_DB_interfaces/02a_pubsub_config.html

2. Sync On Demand

The purpose of a new **Sync On Demand** mode is to reduce the time spent by Fabric performing the LUI sync by synchronizing the relevant data only. In this mode, the sync is performed only on tables that are a part of the executed SELECT statement and on their parent tables. The sync follows the standard Sync mechanism rules based on a LU predefined sync method and sync mode.

The Sync On Demand mode can have either one of the following settings:

- **Sync On Demand = TRUE**: Perform GET LUI and then select from the LUI table(s). Fabric checks whether sync should be performed. If it should, the SELECT triggers a sync of the selected table and its parent tables according to the LU Schema. Once running an additional SELECT, which follows some of the parent tables' synchronization for this GET, the selected tables will not be synchronized again – even if their source tables were meanwhile changed. The GET command is what resets the table's sync state.

- **Sync On Demand = ALWAYS**: Perform GET LUI and then select from the LUI table(s). Fabric checks whether sync should be performed. If it should, the selected table and its parent tables are always synchronized, regardless of whether they were already synchronized for this GET or not.

- **Sync On Demand = FALSE**. This is a default setting which is configurable via the config.ini.

https://support.k2view.com/Academy/articles/14_sync_LU_instance/02_sync_modes.html

3. Broadway

Broadway UI Modernization
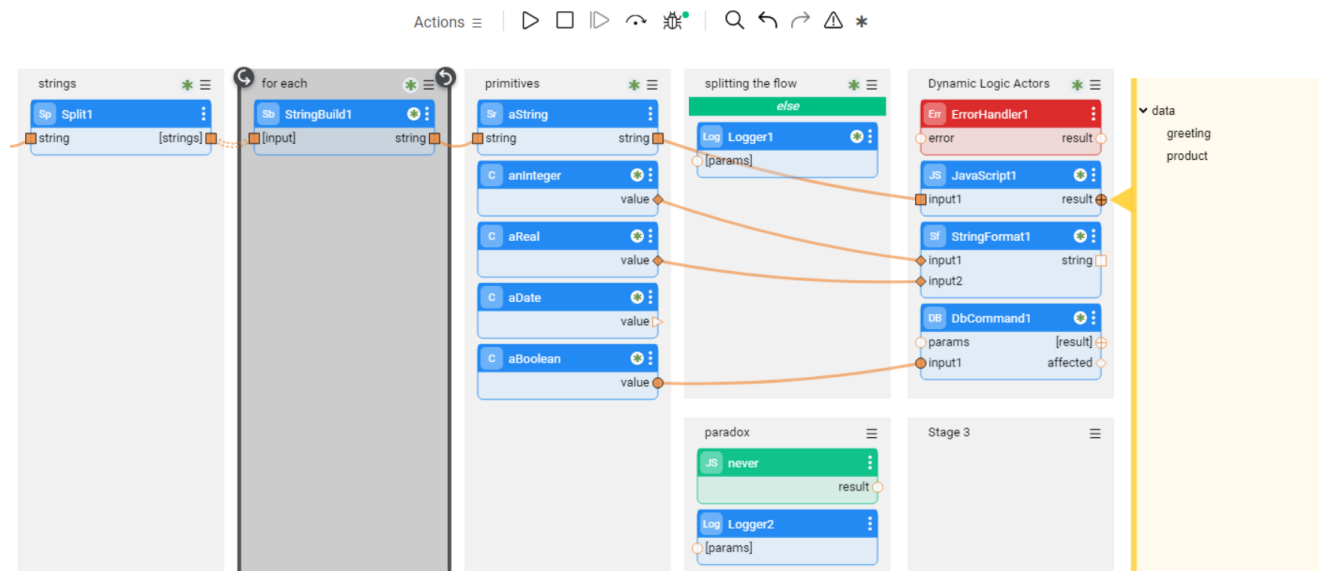
Broadway UI has undergone a major facelift, introducing a new up-to-date coloring scheme.

Broadway command result structure

The Broadway command now enables defining the format of the flow output by setting the new command parameter: RESULT_STRUCTURE=<ROW/COLUMN>. Default is configurable via config.ini. The parameter's values are:

- **COLUMN** (default) –The outputs are returned, each in a column. For example:

```
||result |date          |
|+------+---------------+
||15      |2022-07-19    |
|
```

- **ROW** – The Broadway flow outputs are returned, each in a row. For example:

```
||column |value         |
|+-------+------------+
||result  |15          |
||date    |2022-07-19|
```

- Note that until Fabric 7.0, the result structure was ROW. Refer to the Fabric Upgrade Procedure document regarding how to continue working on your existing project.

Custom editors

Broadway now allows creating custom editors (plug-ins) on a project level. The implementation of your custom editor should be placed in a new folder under the `Server/fabric/staticWeb/editors` folder, in the **index.html** file. Using the Actor Editor, the editor can then be used by any actor, attaching it to the actor's input argument.

https://support.k2view.com/Academy/articles/19_Broadway/28_actor_editor.html

Search enhancements

The Broadway Flow SEARCH was enhanced and now any object in a flow can be searched: stage, actor, parameter (input/output) or a schema element (in the yellow segment). The Search Results field indicates the number of occurrences found in a flow and allows to navigate between them using the Prev/Next buttons.

Copy schema element

Broadway now enables copying a schema element and pasting it in another actor's input or output schema. When the clicked element is complex (it has additional nested levels), the whole complex schema is copied.

New Actors

Filter and **Mapper** – to filter / transform a collection of values based on JavaScript or Inner Flow logic and return the filtered list / list of transformed values. When Inner Flow is used, it must have a **value** input variable. Any single result of the flow or the result 'value' is used as the flow's return value.

Examples of how to use these actors can be found in the **data-transform** example flow.

4. GraphIt

GraphIt's capabilities were improved as follows:

- The GraphIt files can now be defined under categories, at a new GraphIt folder, located in each LU. It is recommended to start using this new GraphIt folder rather than Java/Resources.

- The GraphIt files, located in the new GraphIt folder at Web Services LU, support the following new features:

  - WS versioning and WS Verbs settings (via file naming convention).

  - Ability to decide whether a GraphIt file will be exposed as a Web Service.

  - Additional Permissions and Require Authentication properties.

The existing GraphIt files can support the above features once they are moved under the LU's GraphIt folder and category. This is optional. If not moved, the GraphIt files will continue working as before V7.0.

- All GraphIt files (new and existing) now support the following:

  - Ability to strict the output format, by setting the top node format property.

  - Ability to set parameters as either mandatory or non-mandatory.

  - 2 new node types - "Get" and "Broadway" with appropriate helper functions - for simplifying the creation of GET and BROADWAY commands.

  - Ability to define a description of a WS and of each parameter (via an Open API format).

5. CDC

The Fabric CDC process has been simplified and improved in order to prevent the messages duplication issue that occasionally occurred due to Kafka rebalancing. Fabric now relies on Kafka transactions rather than handling them in an internal queue.

6. iidFinder

LU Alias

- A new property has been introduced to iidFinder.xml on LU level: **alias**.

- This property should be manually added to the XML when the setting of an LU alias, different from the default, is required. The default alias is the first 3 letters of the LU name, and it is not present in the XML.

Broadway population for the iidFinder

- It is now possible to define a Broadway population flow to be used by the iidFinder. This can be done by using the new input arguments on the SourceDbQuery actor. To use SourceDbQuery actor for iidFinder population:

- o Set the new **use_mdb_finder** argument to true.
- o Use the additional new arguments: **luid_column_name**, **is_cross_iid** and **stored**.

7. Miscellaneous

- Starting from Fabric 7.0, the Fabric product code is encrypted. Any project, which invokes the core product functions that are not described in the Fabric documentation, will have to modify the implementation accordingly.

- The **JDK version** has been upgraded from 8 to 17. However, the Fabric Studio still builds the code using the JDK 8. If you wish to use the new features of Java 17 in Studio, do the following:
  - o Update the **JavacArgs** from 8 to 17 in the **k2FabricStudio.exe.config** file under the `Studio` folder.
  - o Handle the compilation errors in your project. For example, *yield* has become a restricted identifier and should be replaced with *UserCode.yield*.

- **DATETIME/DATE/TIME data types** are now supported in Fabric. They are reflected in all Fabric modules: Common and LU table's new data type, Broadway flows and populations, GraphIt, Web Services, Java functions, CDC messages, etc. These data types are saved in Fabric as the following Java classes:
  - o DATETIME - java.sql.Timestamp (supports the time till nano-seconds)
  - o DATE - java.sql.Date
  - o TIME - java.sql.Time

- **AWS Keyspaces and Astra** can now be used as Fabric's operational layer and storage.

- **Google Cloud Storage** is introduced as an additional LU Storage type.

- A new **SET** command has been added: **SET BUFFER_RESULT_SET** (true/false).
  - o When set to true, all the following SELECT statements from Fabric will be uploaded to memory. To inactivate it, set it back to false.
  - o The purpose is to avoid the problem of a locked MicroDB SQLite file, which can happen in a use case where a GET & SELECT from LU1 is followed by a loop on a GET & SELECT from LU2.

- **Web Server self-signed scripts** have been added to enable running Fabric on a secured port.

  https://support.k2view.com/Academy/articles/99_fabric_infras/devops/03_fabric_api_and_ui_hardening.html

- Fabric Open APIs spec - https://<fabirc-endpoint>/api - are now accessible for logged-in users only. An unauthenticated call would fail.

- **WAL Mode for MDB** (Experimental) allows accessing the LUI data during the transaction. For example, during the LU sync, the previous LUI data remains accessible until the transaction is completed.

- **REST API error handling** has been improved. Now, when you throw the *ApiServletException* with the message, only the message is displayed in the response without a full stack trace. Note that the full stack trace can still be found in the log.

- **Fabric JDBC driver** is now secured by default.

- Now, in **CommonDB**, the old snapshots are not automatically deleted. If needed, set DELETE_OLD_SNAPSHOTS_ON_SYNC to true in the [common_area_config] section of the config.ini. Note that the default snapshot table TTL remains unchanged and is set to 7 days.

- You can now define the LU Metadata files storage using the new parameter in the config.ini - **DEFAULT_LU_FILES_STORAGE_TYPE**. By default, it is set to CASSANDRA.

RESOLVED ISSUES

- Issue - when viewing the Cassandra keyspaces/tables, there is no display of the newly added or dropped keyspace/table. The problem has now been fixed.

- Ticket 27781 – Search creates wrong type for IID. The problem has now been fixed.

- Ticket 28666 – Documentation of the SourceDbQuery actor was improved.

- Ticket 28896 – IIDF not able to process Kafka messages if table/field names appear in lowercase. The problem has now been fixed.

- Ticket 29037 – IID Finder does not delete deltas. The problem has now been fixed.

- Ticket 31272 – reset.bat does not work. The problem has now been fixed.

- Ticket 31409 – Test connection shows success even when it fails. The problem has now been fixed.

- Ticket 31317 – IID Finder cache populated with wrong IID. The problem has now been fixed.

- Ticket 31781 – Studio closes modified files without a prompt. The problem has now been fixed.

- Ticket 31953 – Reset drop fails to drop keyspaces. The problem has now been fixed.

- Ticket 31705 – SftpException exception is thrown when the ErrorHandler is set on suppress. The problem has now been fixed.

- Ticket 31966 – Multi-source on different LU tables with IID Finder. The problem has now been fixed.