



FABRIC V8.5.0 RELEASE NOTES

These Release Notes describe the new features in Fabric release V8.5.0 and list the bugs that have been fixed since the latest V8.4 release.

Certification of this Fabric release is based on:

3 rd Party Name	Included with Fabric	3 rd Party Server
Cassandra	Cassandra-driver-core-3.11.5	4.1.8
Elasticsearch	8.13.1	8.5.3
Kafka	3.9.2	Confluent-8.1.1
Neo4j Enterprise	5.26.29	-
OpenJDK Runtime Environment	openjdk-21.0.11+10	openjdk-21.0.11+10
OpenSearch	3.4.0	AWS 1.3.4
PostgreSQL	postgresql-42.7.13.jar	17.5
SQLite	3.46.1	-

LLM connectors (like OpenAI and Anthropic): use version 2.X

MAIN FEATURES AND IMPROVEMENTS

1. Studio AI

- k2Studio's embedded AI assistant handles Fabric project work end-to-end: creating and editing Data Products, building and debugging Broadway flows and actors, interfaces, web services, Graphit, MTables, and Globals, plus answering project and runtime questions and troubleshooting errors.
- It is fully integrated with the Fabric runtime, so it deploys, tests, and verifies its own work while crafting artifacts and analyzing issues.
- It is backed by a set of specialized skills, reference docs, and LLM-wiki files, along with K2View's AskMe vector DB service, covering the full support site (articles and Q&A), Fabric command syntax, implementation Javadocs, and more.
- The k2assistant is an open, extensible agent framework:
 - Teams can fully review built-in agents and skills alongside with defining their own project/organization custom agents and reusable skills.
 - Every request and response are fully visible through a dedicated history log and token-consumption dashboard.



FABRIC RELEASE NOTES

- Various capabilities can be set per chat session where session history is fully available to review and revive.
- The LLM provider and model is configurable per agent, where custom LLM providers can be set, including local, on-premises.

→ To enable these AI capabilities, install the *Studio AI Core Artifacts* extension

2. Fabric Catalog

- Large Catalog navigation:
 - Until V8.5, navigating large schemas in the Catalog App was slow and sometimes unresponsive. This was because expanding a schema with thousands of datasets, fields, and relations required retrieving and rendering a large volume of data on the client side, which sometimes caused the browser to crash.
 - To address this issue, the Catalog App introduces the **Schema Dataset List view**: for large schemas, datasets are displayed in a filterable list instead of the tree view, reducing rendering time and improving navigation.
 - All actions available in the standard tree view — such as viewing and editing datasets, fields, properties and relations — are fully supported in the list view.

https://support.k2view.com/Academy/articles/39_fabric_catalog/catalog_app/18_large_schema_navigation.html

- File Cataloging — OpenAPI Support:
 - The File Cataloging framework has been extended to support discovery of OpenAPI interfaces (versions 3.0 and 3.1).
 - A new **OpenApiToMetadata** Broadway actor is added to parse OpenAPI specifications and transform them into Catalog's standard metadata format, enabling the standard Discovery pipeline — including classification, PII detection, and more — to run in the same way as for any other data source.
 - API endpoints (paths) are mapped as Catalog datasets, with schemas derived from OpenAPI tags and field sets merged across all HTTP methods defined on each path.

https://support.k2view.com/Academy/articles/39_fabric_catalog/06_open_api_support.html

- Search Catalog enhancement:
 - The Catalog Advanced search has been enhanced with a new 'Limit by hierarchy' search option, which allows users to limit the search scope by selecting a Data Platform, or both a Data Platform and a Schema.

https://support.k2view.com/Academy/articles/39_fabric_catalog/catalog_app/08_search_catalog.html

- Data Quality Metrics plugin:
 - Enhanced with additional metric calculations and input parameters.
 - Added a restriction related to PII fields: when a field is marked as PII, the minimum and maximum values are not calculated for it to avoid revealing sensitive data.



FABRIC RELEASE NOTES

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

- Reference by Name Comparison plugin:
 - Introducing a new rule - *fieldNameCompare* - based on the Jaro-Winkler similarity algorithm, which compares two field name strings and produces a score between 0 (completely different) and 1 (identical). Unlike the other matching rules, this rule does not perform a PK check of the compared fields and can create a *refersTo* relation between any two fields based on name similarity alone.

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

- Reference by LLM plugin:
 - This new plugin identifies possible FK references between datasets by analyzing dataset pairs using an LLM and creates corresponding *refersTo* relations.

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

- Class nodes - naming convention improvement:
 - Until now, when an ad-hoc class was created in the Catalog — either from a complex field parsed from data (e.g., JSON, XML) or from a JSON Schema ad-hoc class definition — the parent class names used to be concatenated to the class name to prevent name ambiguity, e.g.:
 - AddressClass;PrimaryContactClass;CityClass
 - AddressClass;ShippingInfoClass;CityClass
 - In this version, this logic was changed to align with the Web Studio naming convention used during the LU Schema creation. The class name is now kept as is, and uniqueness is achieved by appending 3 underscores followed by the parent class name. If the parent's name alone doesn't provide uniqueness, additional ancestor levels (grandparent and beyond) are appended, separated by a single underscore, e.g.:
 - CityClass___AddressClass_PrimaryContactClass
 - CityClass___AddressClass_ShippingInfoClass
 - Note that all classes sharing the same name receive the same number of appended ancestor names. When the root is reached, the @root@ annotation is used as the parent's name.
- DISCOVERY_JDBC_LIMIT_BY_QUERY parameter refactoring:
 - The DISCOVERY_JDBC_LIMIT_BY_QUERY parameter has been removed from the config.ini and replaced by another parameter — **Limit** — available in the Discovery Pipeline settings screen, under the Crawler section.
 - When Limit is set to **true**, the query limit is applied directly in the SQL query rather than via the JDBC driver's setMaxRows method.
 - By default, it's set to **false**.
- Catalog Settings enhancements:



FABRIC RELEASE NOTES

- The **Classifier Regex**, **PII & Masking**, and **Sequences** tabs in Catalog > Settings now offer a smoother filtering experience: the Classification filter is always expanded and focused, so a user can simply start typing to filter the list, without first having to click and select the filter field.
- The product regular expressions in the **Classifier Regex** have been improved, by removing the regexes which caused a high number of false positives (e.g. DATE_OF_BIRTH, DRIVER_LICENSE, ZIP_CODE) and adding some more specific regexes (e.g. CREDIT_CARD, STATE).
- The product settings of the **PII & Masking** have been updated by adding default generators for some classifications that previously had no generator.
- The **Classifier Regex** tab now allows disabling a classification rule instead of deleting it, making it easier to manage rules that are temporarily unused. The disabled rules are skipped by the *Metadata Regex Classifier* and *Data Regex Classifier* classification plugins.

https://support.k2view.com/Academy/articles/39_fabric_catalog/catalog_app/10_catalog_settings.html

3. Security Improvements

- **API Key security:** Strengthen API Key-based authentication by giving a stronger, higher-entropy key option. API Keys now support three explicit types, selectable via CREATE TOKEN command or the Web Admin UI:
 - Access - high-entropy key, when sending API key as *Authorization: Bearer* header. This is the recommended type and is now the default in the Admin UI.
 - Legacy - the existing name-as-key behavior, preserved for backward compatibility and thus also set as default.
 - Signing JWT - the client signs its own short-lived JWT using a Fabric-issued signing key (replacing the previous “Signed by client” / “Secured” option).
- Both the Access and Signing JWT keys are never displayed or retrievable again after creation.
- **Authenticate API mechanism:** The /authenticate endpoint now distinguishes end-user logins from server-to-server API calls. End-user sessions continue to receive their JWT as a secure cookie tied to their identity. Server-to-server callers now explicitly request the JWT in the response body, keeping API access decoupled from browser session cookies. Audit logs now clearly attribute each login to the resolved user or, for pure API-key access, to the API Key itself (in both cases, only a partial identifier is written to the audit log, for security).

4. IID Finder as a Fabric Job

- The **IID Finder** is no longer executed as a standalone process; it now runs as a **Fabric job**.



FABRIC RELEASE NOTES

- Its logs and JMX metrics are merged into Fabric, the start and stop scripts and the **iifConfig.ini** file are removed, and its settings are relocated to **config.ini**. Existing Broadway flows are preserved.
- For more information, see the [Fabric 8.5 IID Finder Upgrade to Fabric Job](#) document.

5. Other Improvements and Changes

- The **SET INSTANCE_TTL** command has been enhanced to support Time To Live (TTL) management for MicroDBs stored in **Google Cloud Storage (GCS)**, **Azure Blob Storage**, and **Amazon S3**.

https://support.k2view.com/Academy/articles/02_fabric_architecture/04_fabric_commands.html

- Hybrid MicroDB cache:
 - A new CACHE_TYPE value, **HYBRID_NO_CACHE**, is available. This option is designed for data products that contain a mix of small and large MicroDBs, optimizing both performance and memory utilization.
 - When CACHE_TYPE is set to **HYBRID_NO_CACHE**, Fabric determines where to store each MicroDB cache, in memory or on disk, based on its size. The size threshold is configured using the **HYBRID_MEMORY_THRESHOLD** parameter in the [fabriccdb] section.

https://support.k2view.com/Academy/articles/32_LU_storage/05_in_memory_lu_storage.html

- Affinity Management:
 - Fabric now provides centralized Affinity Management, designed for Kubernetes elastic scaling. As cluster nodes are added or removed, the affinity engine automatically ensures that affinity roles are assigned according to the configured priority rules. Rules can be centrally managed for an entire cluster or per data center through the Admin UI or the new **set_global affinity_rules** command.
 - A dedicated Affinity Management Job (one per DC) automatically distributes and maintains the settings on all nodes.
 - The feature also introduces node groups (colors) and new commands for setting, viewing and monitoring node affinities.

https://support.k2view.com/Academy/articles/20_jobs_and_batch_services/19_affinity_management.html

- Retry during parallel download from S3 / Azure:
 - If a BLOB is modified and saved during a parallel download from S3 or Azure, Fabric automatically retries the download to retrieve the updated BLOB.
- Support of S3 versioning during parallel download:
 - A new parameter **IGNORE_VERSION_CHANGE_DURING_PARALLEL_DOWNLOAD** has been added to the config.ini and by default it is set to **true**. This means that if a new BLOB version is saved while a download is in progress, the download is retried, up to **MDB_LOAD_ATTEMPTS** attempts.



FABRIC RELEASE NOTES

- When the parameter is set to **false**, the download of the previous version will continue. This requires S3 object versioning to be enabled.
- Batch fetch size:
 - A new parameter **BATCH_FETCH_SIZE** has been added to the [batch_process] section of the config.ini (default: **1000**). It defines the fetch size to be used when batch population is based on an SQL query. Its purpose is to improve the performance of the Batch process IID generation step.
- Wrapping Fabric-generated queries with quotes:
 - When the parameter **ALWAYS_QUOTES_INTERNAL_QUERIES** is set to **true**, all fields in Fabric-generated queries are wrapped with quotes.
 - By default, it is set to **false**.
- The **DROP LUTYPE** Fabric command has been enhanced to support dropping only the data. A new **data_only** parameter has been added; set it to Y to drop only the data while preserving the LUTYPE definition.
- List of schemas per interface:
 - The existing */api/interface_schema/schemas* REST API that retrieves a list of interface schemas has been enhanced to return all interfaces with an indication whether each schema is retrieved from Catalog or from the source. For that purpose, a new *schemaSource* input parameter has been added and it can get one of the following values:
 - **<empty>** (default) - if the interface is in the Catalog, bring schemas from Catalog; otherwise - bring them from the source. The same behavior as before V8.5.
 - **source** - bring schemas from source only.
 - **catalog** - bring schemas from Catalog only.
 - **both** - bring schemas from Catalog and from the source.
- Fabric no longer requires a restart when **AUDIT=ON** is set in config.ini.
- **Improved MDB export performance** by using the PostgreSQL COPY command when exporting LU instances to PostgreSQL.
- Graceful Web Server shutdown:
 - Enhanced the Web Server shutdown process to support graceful termination of active requests. During shutdown, the server stops accepting new connections, allowing load balancers to redirect incoming traffic to other nodes, while continuing to process existing requests for a configurable grace period.
 - The grace period is set by default to **30 seconds** and is controlled by the **WEB_SERVICE_SHUTDOWN_TIMEOUT_SECONDS** parameter.



FABRIC RELEASE NOTES

- Once all active requests are completed, shutdown proceeds immediately without waiting for the full timeout. Requests that exceed the configured grace period are terminated, and any unfinished request-processing threads are logged with warning messages and stack traces to assist with troubleshooting.
- Setting the timeout to **0** disables the grace period and performs an immediate shutdown. This enhancement minimizes customer impact during planned restarts and improves high-availability behavior in load-balanced environments.
- The **IO_JDBC_PREPARED_STATEMENT_SET_NULL** configuration parameter is now enabled by default (**true**) in the config.ini file.

RESOLVED ISSUES

- The discovery LLM Description plugin was triggering a data snapshot step, even when its sample size was set to 0 and all other data-related plugins were inactive.
- Ticket #45272 – JobReconcile actor got stuck on quitting and didn't execute any more jobs. It was caused by the InnerFlowAsync actor in one of the Broadway jobs.
- Ticket #48140 - Version tag is now affected also at Studio deployment.
- Ticket #48292 – Broadway iterated over the result set of a disabled DbCommand actor.
- Ticket #49345 – In the Admin web app, cluster time showed UTC time zone instead of the local one.
- Ticket #49704 – In Broadway, creating links between actors using the Link Manager didn't work.
- Ticket #50396 – The discovery pipeline filter definition didn't match the source in terms of case sensitivity. Now, after the fix - if the source is case insensitive, the **dataset** name setting in the discovery pipeline filter might also be case **insensitive**.
- Ticket #50261 – The Web Framework Admin menu remained visible despite *hidden=true* in apps.json. Now apps.json properties (like "hidden") defined at the project level **override** the base apps.json properties.
- Ticket #50447 – Catalog parser didn't recognize CLOB field in Oracle DB with XML inside as complex.
- Ticket #50721 – When running Discovery on a table containing a CLOB field with a JSON structure in each row, an inner JSON level that sometimes contained an object and sometimes was *null* was not parsed as complex field.
- Ticket #50809 – Catalog Schema details are not expanded due to invalid retrieve of nodes from the Catalog tree.
- The following 3rd Party tools were upgraded:
 - Tomcat 11.0.22
 - Neo4j 5.26.29
 - MCP core 1.1.1



FABRIC RELEASE NOTES

- Masterkey Fortanix log4j-core-2.25.4
 - google-oauth-client-1.39.0
 - netty-codec-haproxy 4.1.136
 - postgresql 42.7.13
- The following 3rd Party tool was removed:
 - rhino-1.7.7.2.jar