



# FABRIC RELEASE NOTES

## FABRIC V8.4.4 RELEASE NOTES

These Release Notes describe the new features in Fabric release V8.4.4 and list bugs that have been fixed since the latest V8.4.3 release.

Certification of this Fabric release is based on:

3 <sup>rd</sup> Party Name	Included with Fabric	3 <sup>rd</sup> Party Server
Cassandra	java-driver-core-4.17.0.jar	4.1.11
Elasticsearch	8.13.1	8.5.3
Kafka	3.9.1	confluent-8.1.1
Neo4j Enterprise	5.26.26	-
OpenJDK Runtime Environment	openjdk-21.0.11+10	openjdk-21.0.11+10
OpenSearch	3.4.0	AWS 1.3.4
PostgreSQL	postgresql-42.7.8.jar	17.5
SQLite	3.46.1	-

### MAIN FEATURES AND IMPROVEMENTS

- Ticket #49532 (CR) – Fabric now supports exposing attributes retrieved from the SAML response as claims in the session JWT generated by Fabric. This allows these attributes to be accessed and utilized within implementation-specific functional code.
- **Studio AI – Issue Resolver Agent** – introduces a new AI agent that diagnoses coding errors and runtime failures. The agent is accessible through Studio AI chat or directly from error notifications via the new **Ask AI** button (for example, on failed test connections). It integrates with the **Coder Agent**, enabling suggested fixes to be automatically applied for faster issue resolution.

### RESOLVED ISSUES

- Ticket #49576 – resolve sync failures during Fabric upgrades from version 6.5.9 to 8.4.3.
- Ticket #49727 – fix DBLoad actor to correctly handle table names containing the special character '@'.
- Fix Fabric to work correctly with Azure-managed PostgreSQL.
- Ticket #49613 – fix Sync on Demand to prevent failures when processing standalone LU tables without parent tables.
- Ticket #49750 – add support for running the SourceDBQuery actor inside a loop.



# FABRIC RELEASE NOTES

- Ticket #49561 – fix missing command parameter logging when running Fabric in audit mode.
- Catalog:
  - Build artifact progress was not displayed correctly when monitoring a previously deleted data platform.
  - An array of strings inside a complex structure was parsed incorrectly as the UNKNOWN data type, instead of Collection (STRINGS).
  - When running discovery on a JSON Schema containing ad-hoc objects nested two or more levels inside array items (e.g., root → Groups[].Transactions[].ST.ST\_01), fields of these deeply nested objects were duplicated in the metadata output — appearing once correctly under their own class, and once incorrectly under the root dataset class.
  - Ticket #49641 - Plugins that use snapshot data stopped working on complex fields containing XML.
- Resolved 3rd party security vulnerabilities:
  - Upgrade tomcat to 10.1.55.
  - Upgrade OpenJDK to 21.0.11+10.
  - Upgrade Theia to 1.72.0.