



## FABRIC UPGRADE PROCEDURE TO V7.2

This document describes:

- How to upgrade Fabric to the present version: from the latest **V7.1.1** to **V7.2**.
- How to re-implement the modified product features.

Notes:

- This document does not cover the Fabric server topology changes, such as nodes addition, DCs, change of replication factor and consistency level.
- It is a must to perform the Fabric upgrade procedure in the testing environments prior to applying it on your production deployment.
- Sanity tests must be performed upon the completion of the upgrade procedure, such as performing a few GET commands and conducting other checks per the sanity procedure defined in your project.

## SOFTWARE UPGRADE PROCEDURE

### Preliminary Step

Download the latest release of Fabric V7.2 package and copy it to the server(s).

### Stop Fabric

Take the following steps in the specified order:

1. If your project has an iidFinder:

Stop the iidFinder on all nodes.

Wait until Kafka lags are zero in the relevant Kafka topics:

- Delta\_cluster\_<LU\_name> topics
- DeltaPriority\_cluster\_<LU\_name> topics

Run the following command to verify that the lag on the above topics is zero:

```
/opt/apps/k2view/apps/kafka/bin/kafka-consumer-groups --bootstrap-server <internal Kafka server IP> --group <Kafka interface group ID> --describe | awk '{if ($6>0) print $0}'
```

Investigate the remaining messages in the Delta tables and clean them, by taking the following steps per each LU:

- Run MIGRATE command on all distinct IIDs.
  - Check the results in order to decide how to proceed with the failed entity messages.
  - Clean the Delta table.
2. Stop Fabric on all nodes.

### Open the Package

Perform the following steps:

Rename the Fabric directory as shown, type the specific Fabric version in the indicated location:

```
cp -r config config_$(k2fabric -version | awk '{print $2}' | head -n1)
```



# FABRIC UPGRADE PROCEDURE

```
mv fabric $(k2fabric -version | awk '{print $2}' | head -n1)
```

Extract (un-TAR) the Fabric directories from the upgrade package (extract only the directories) as shown. The specific file name will depend on the specific Fabric version:

```
tar -zxvf k2fabric-server-fabric-<package_name>.tar.gz fabric apps
```

Adding **apps** after **fabric** installs the correct Java version and creates a soft link to Java.

## Run Upgrade Scripts (where applicable)

*When upgrading to a version whose number is not consecutive to the version you are using, run all the upgrade scripts of all versions in between. For example, when upgrading from the latest V6.4 to V7.0, run all the upgrade scripts in the following order: 6.4 -> 6.5 -> 7.0.*

*The versions that have the upgrade scripts between the latest 6.4 and 7.1 are: 6.5.8, 7.0, 7.1.*

*The upgrade scripts should run from one Fabric node only. After running the scripts, verify that all the changes have been applied.*

Run the upgrade script provided with the package of V7.1 - **update-cdc-templates.sh** – to support complex search templates as part of CDC for Elasticsearch V8.0.

## Verify Upgrade Success

Use the following command to verify that the upgrade completed successfully:

```
k2fabric -version
```

The result will display the Fabric package number, for example:

```
Tag fabric- 7.0.0_188 at revision = 3fc12ed1c6839614a9fd27e2894828bcd160988f
```

If there are local files on the server (such as local JARs), their names will be displayed here.

## Configuration Changes

*Prior to performing the changes, backup your project configuration files, such as config.ini, etc. Please see the below configuration changes summary per each version.*

### To Version 7.2

- The .NET Studio has moved to the new encryption mechanism **AES 256** for all project's passwords (used in the interfaces and environments). The encryption upgrade is done automatically upon opening of the project.
  - Automatic passwords upgrade on opening or importing a project can be disabled by setting the following property in the Studio app config to false:  
*AutoUpgradePasswordsOnProjectOpen.*
  - If the automatic upgrade is disabled, the password will be upgraded separately on each interface, when opening it in the Studio and saving it.



# FABRIC UPGRADE PROCEDURE

- Note that after upgrading the project's passwords encryption, an older version of Studio or Fabric won't be able to read/decrypt the passwords of the upgraded project. The passwords will have to be re-typed again, when opening a project in the older Studio.
- This feature can be disabled by setting the following property in the Studio app config to false: *EnableUpgradePasswordsEncryption*.

## Implementation Changes

*Please see the below implementation changes summary for each version.*

### To Version 7.2

N/A

## Start Fabric

Take the following steps in the specified order:

1. Start the first Fabric node and wait until it has been completed successfully.
2. Start all other nodes.
3. Deploy the project.
4. If your project has an iidFinder, re-start the iidFinder on all nodes.

Note that the above steps may vary per your project's runbook. For example, you may first re-start iidFinder on a single node only, verify the process has completed successfully and then proceed to re-starting it on all other nodes.