



FABRIC UPGRADE PROCEDURE TO V8.0.0

This document describes:

- How to upgrade Fabric to the present version: from the latest **V7.2.2** to **V8.0.0**.
- 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 V8.0.0 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 (e.g. deltaGroup)
- DeltaPriority_cluster_<LU_name> topics (e.g. IDfinderGroupId_Cluster)

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
```

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
```

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.

This step is not relevant for the upgrade from the latest **V7.2.2** to **V8.0.0**.

Verify Upgrade Success

Use the following command to verify that the upgrade procedure has been 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 read the below configuration changes summary per each version.

To Version 8.0.0

- **Fabric and project dependencies** – Fabric's dependencies are now isolated from the project's dependencies by default. This isolation allows a higher flexibility for the project to use different versions of 3rd party JARs than those used by Fabric.
 - During your project upgrade to V8.0, you might receive *ClassNotFoundException*.
 - To resolve these exceptions, find which JAR files contain the problematic Java classes and copy them to the project's **lib** folder.
 - The of 3rd party JARs frequently used by the projects are: **kafka-clients-3.4.0.jar**, **jsonassert-1.5.1.jar**, **xxx**.
 - To restart the Fabric server, do the following:
 - **XXXXX**



FABRIC UPGRADE PROCEDURE

- If you wish to disable the Fabric and project dependencies isolation and return to the Fabric 7.2.x behavior, do the following:
 - In the Desktop (.NET) Studio: open the **k2FabricStudio.exe.config** file under the Studio folder and update the **FabricFoldersClasspath** key by adding the **fabric\lib\fabric*** value.
 - In Web Studio: open the Workspace **settings.json** and update the **referencedLibraries** by adding **/opt/apps/fabric/workspace/fabric/lib/fabric/**.
 - In addition, in the Fabric server: update the config.ini by removing the value in the **PACKAGE_NAMES_CLASS_LOADING_FILTER** parameter.

Implementation Changes

Please see the below implementation changes summary for each version.

To Version 8.0.0

N/A

Start Fabric

Take the following steps in the specified order:

1. Start the first Fabric node and wait until it has been successfully completed.
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 restart the iidFinder on a single node only, verify that the process has been successfully completed and then proceed to restarting the iidFinder on all other nodes.