FABRIC UPGRADE PROCEDURE

FABRIC UPGRADE PROCEDURE TO V6.5.0

This document describes:

- How to upgrade Fabric to the present version: from the latest **V6.4.3** to **V6.5.0**.
- How to re-implement the modified product features.

Notes:

- This document does not cover the Fabric server topology changes, such as addition of nodes, DCs, change of replication factor or 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 test must be performed upon the completion of the upgrade procedure, such as performing a few GET commands and doing other checks per the sanity procedure defined in your project.

SOFTWARE UPGRADE PROCEDURE

Preliminary Step

Download the latest release of Fabric V6.5 package and copy it to the server(s):

Fabric Server
Fabric Studio

Stop Fabric

Do these steps in the specified order:

1. If your project has iidFinder:
   - Stop iidFinder on all nodes.
   - Wait until Kafka lags are zero in the relevant Kafka topics:
     a. Delta_cluster_<LU_name> topics
     b. DeltaPriority_cluster_<LU_name> topics
   - Run the following command to check the lag on those 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 doing the following steps per each LU:
     a. Run MIGRATE command on all distinct IIDs.
     b. Check the results to decide how to proceed with the failed entities messages.
     c. Clean the Delta table.
2. Stop Fabric on all nodes.

Open the Package

Do the following steps:

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

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

2. 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 (if relevant)

When upgrading to a version which is not immediately after the version you are using, run all the upgrade scripts of all versions in between. For example, when upgrading from the latest V6.2 to V6.5.0, run all the upgrade scripts in the following order: 6.3 -> 6.4 -> 6.5.

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

Upgrade scripts are included in the package only when the version has schema related changes. If there are no such changes, the upgrade script will not exist.

This step is not relevant for the upgrade from the latest V6.4.3 to V6.5.0.

Verify Upgrade Success

Use the following command to verify that the upgrade was successful:

```
k2fabric -version
```

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

```
Tag fabric-6.5.0_64 at revision = 3fc12ed1c6839614a9fd27e2894828bcd160988f
```

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

Re-Start Fabric

Do these steps in the specified order:

1. Backup your project configuration files: config.ini and iifConfig.ini.
2. Re-start Fabric on all nodes.
3. Deploy the project.
4. If your project has iidFinder, re-start iidFinder on all nodes.
5. Compare the new configuration files with your project to verify if there are changes which should be applied to the project configuration.

Note that the above steps may vary per your project's runbook. For example, you might first re-start iidFinder on one node only, verify the success and then proceed to re-start it on all other nodes.

IMPLEMENTATION CHANGES

To Version 6.5.0

- **Proactive Indicator** (which defines whether Proactive Sync should be enabled) was removed from the DB Interface definition and will now be supported on LU Population only.
  Important! Verify which LU Populations use this property and make sure to turn this indicator on the population itself.

- **iidFinder <cacheTableLuAlias>** property is removed from the IID Finder XML.
  If you were using <cacheTableLuAlias> property, manually remove it from the IID Finder XML.

- **iidFinder staging.xml** was removed and is not supported anymore. Currently only IID Finder XML from Fabric Studio is supported.
  If you were using starting.xml, copy the parts of staging.xml into the IID Finder XML per each Logical Unit.

- **Removed Jedis Functions** - the following deprecated Jedis functions were removed:
  - getJedisConnection
  - closeJedisConnection

  If you were using these functions, replace them by using Redis Interface type.

  If you are using the Fabric's Masking Library (the Java functions and not the Broadway Actors), then the Masking Library must be upgraded to its v6.5 version.