DOC 4.0.1 FP3 (Release Notes and Migration Guide)

Migration Procedure

IMPORTANT: The recommended way to perform a migration is described below. Shortcuts are possible, but are at your own risks.

Migration Procedure from 4.0.1 FP2

If you are currently using DOC 4.0.1 FP2, you can follow the steps below to migrate to 4.0.1 FP3. Otherwise, please migrate first to 4.0.1 FP2 using the standard migration procedure, that is, re-scaffold and report the changes.

- *In gradle/templates/versions.gradle, line 23: change the DB Gene version to "4.0.1-fp2.1"*
- *In gradle/templates/versions.gradle, line 26: change the DBOS version to "3.3.1"*
- *In deployment/docker/dbos/.env, line 7: change the DBOS version to "3.3.1"*
- *In the root directory of your project, run ./gradlew updateCode to apply the DB Gene version change throughout the project*
- *If you deploy using Helm, follow these instructions:*

```
dbos:
    master:
        image:
            imageId: "dbos-master:3.3.1"
    console:
        image:
            imageId: "dbos-web-ui-dashboard:3.3.1"
    documentation:
        image:
            imageId: "dbos-documentation:3.3.1"
```

Changes in the Build Chain

Gradle version change from 5.6.2 to 7.1.1

Gradle Plugins versions

jacoco 0.8.3 0.8.7

sonarqube-gradle-plugin 3.1.1 3.3

com.moowork.gradle:gradle-node-plugin:1.3.1 com.github.node-gradle:gradle-node-plugin:3.1.0

openapi-generator-gradle-plugin 5.1.0 5.2.0

dom-generator-plugin 0.5.4 0.5.6

Gradle Build Scripts

The memory consumption for the build has been optimized from 4g to 2go.
You have to update the gradle.properties file with the following value:

```
# Default Xmx for the build
org.gradle.jvmargs=-Xmx2g
```

Jenkins pipeline “docker build”

If you use Jenkinsfile docker.build() steps ensure that the build context passed to docker is `<project>/build/docker`

for instance :

```
  docker.withRegistry('https://cplex-registry.decisionbrain.cloud',
'NEXUS_USER') {
        image = docker.build("engine-worker:$dockerTagVersion", "--
network=dockernet254 workers/engine-worker/build/docker")
    }
```

Changes in API

List by back-end µ-service (REST API, Java API), plus Typescript API

Model

- Method `DbDomCollector.getMetadata()` has been removed
- `byte[] saveSnapshot(outputStream, classes, format)` should now be called with DBRF_UNCOMPRESSED instead of DBRF to have the same behavior
- `Collection<Issue> loadSnapshot(inputStream, format)` should now be called with DBRF_UNCOMPRESSED instead of DBRF to have the same behavior

Scenario service

- `ScenarioService.lockScenario` has been removed, use `ScenarioLockService.lockScenario` instead
- `ScenarioService.systemLockScenario` has been removed, use `ScenarioLockService.systemLockScenario` instead

Execution service

- The `ExecutionServiceClient.launchJob()` method now returns a `ResponseEntity<String>` (it used to return a `String`).
- The `AskInputStatement.of(String inputName, boolean required, ParameterType inputType[, String description])` method used to take a `JobInputType` as its third argument.

Also, `JobInputType.NUMERIC` is deprecated in favor of `JobInputType.REAL`.

- Change in JobLog and JobLogRow persistence model:
  - JobLog mongo document is no longer used, and was removed from mongoDB
  - JobLogRow now has an indexed field jobId allowing to query all logs from a job

This change does not affect existing endpoints signature to retrieve logs for a job.

However this change means that jobs created in previous versions will lose their logs after migrating to 4.0.1 FP3.

If this is an issue, a suggested solution would be to retrieve the logs from mongoDB before migrating (find query on Document `jobLog`), and then adding the field jobId and persisting them in mongoDB after migration (insert query on Document `jobLogRow`). This is facilitated by the fact that we still use the same document to store logs: JobLogRow.

To fetch job logs from previous versions:

To return log rows for a given job id value JOB_ID , adding the JOB_ID value to the returned structure use the following query:

```
db.jobLog.find( { jobId : "JOB_ID" }, { "logRows.$": 1, _id: 0 } )
```

Note that the `._id : 0` is to avoid default behavior of returning _id field value for JobLog document, which is unnecessary in our case.

The returned log rows in json format should be altered to add the new field jobId, then the transformed json can be used to insert the log rows from into the new document collection JobLogRow.

After migrating an insert query can be used to populate JobLogRow collection. The query should look like the following for a given JOB_ID:

```
db.jobLogRow.insertMany( [
  { jobId : "JOB_ID", level: "INFO", sourceType: "DBOS", sourceName:
"My Task Name", timestamp: 1631797847934, stepIndex: 1, message: "First
log line"},
  { jobId : "JOB_ID", level: "INFO", sourceType: "DBOS", sourceName:
"My Task Name", timestamp: 1631797847935, stepIndex: 2, message:
"Secondlog line"},
  { jobId : "JOB_ID", level: "INFO", sourceType: "DBOS", sourceName:
"My Task Name", timestamp: 1631797847936, stepIndex: 3, message: "Third
log line"},
  { jobId : "JOB_ID", level: "INFO", sourceType: "DBOS", sourceName:
"My Task Name", timestamp: 1631797847937, stepIndex: 4, message:
"Fourth log line"},
  { jobId : "JOB_ID", level: "INFO", sourceType: "DBOS", sourceName:
"My Task Name", timestamp: 1631797847938, stepIndex: 5, message: "Fifth
log line"}
] )
```

Web

- The following interfaces`GeneWidgetConfigurationAware` ,`GeneDynamicWidgetEvent` and `GeneDynamicWidgetEventType` have been moved from `@gene/widget-data` library to the `@gene/widget-core`library. As consequence the imports of these classes must be updated:

```
// 4.0.1-fp2
import { GeneWidgetConfigurationAware, GeneDynamicWidgetEvent,
GeneDynamicWidgetEventType} from '@gene/widget-data';

// 4.0.1-fp3
import { GeneWidgetConfigurationAware, GeneDynamicWidgetEvent,
GeneDynamicWidgetEventType} from '@gene/widget-core';
```

- `GeneBaseDataWidgetComponent` method `canLoadData(): boolean | Observable<boolean>` has a different return type. Existing code remains retro compatible.
- package.json requires updating the following dependency versions

```
"ag-grid-angular": "26.0.0", (update)
"ag-grid-community": "26.0.0", (update)
"ag-grid-enterprise": "26.0.0", (new)
```

Changes in User-facing Features

…

GeneTable

The default behavior of the GeneTable columns changes a little bit with the 4.0.1-fp3.

Default Columns

Default Columns before 4.0.1-FP3

- *Readonly mode*
- All Scalar columns (text, numbers, dates)
- For each relation a relation with one of its nested scalar column, eg: Plant (Name) for Plant column
- *optional additional columns on relation nested fields*

Edition mode

- All scalar columns
- One column for each relation displaying the BK field values
- no additional columns on relation nested fields

Default Columns with 4.0.1-FP3

Readonly mode

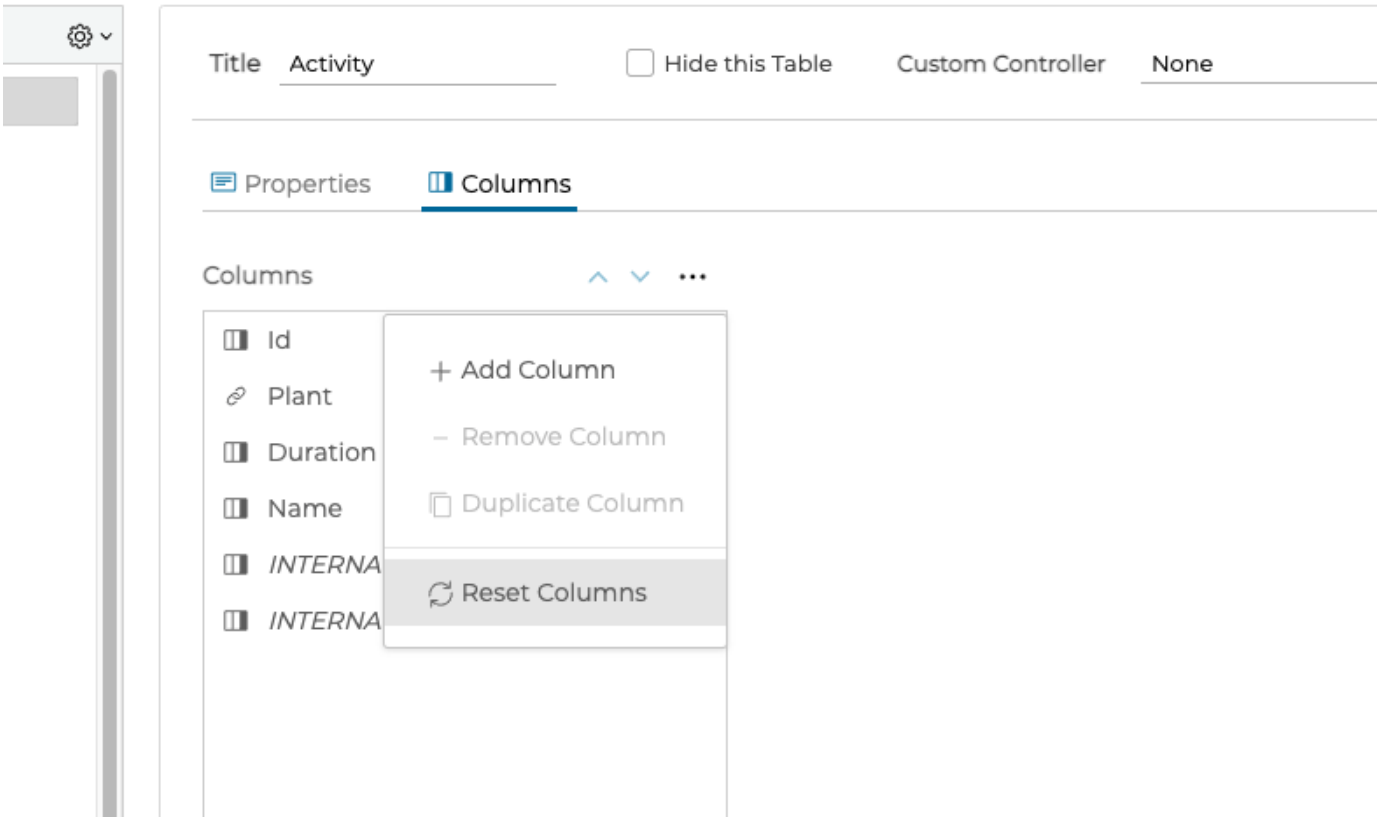
- All Scalar columns (text, numbers, dates)
- One column for each relation displaying the BK field values
- optional additional columns on relation nested fields

Edition mode

- All scalar columns
- One column for each relation displaying the BK field values
- no additional columns on relation nested fields

This change brings a better user experience because the columns (and their order) no longer change when users switch between the two modes.

To leverage this new mode you can reset the column configuration of the data grid through the configurator:



Changes in Deployment

Changes to Docker images, to Docker Compose scripts/config, move from OKD to K8s and Helm Charts

- The base Docker image for java micro-services has changed to `index.docker.io/library/adoptopenjdk:11.0.11_9-jre-hotspot`
- The Docker image for Keycloak micro-service has changed to `${DOCKER_REGISTRY}/decisionbrain/keycloak:14.0.0`
- The Keycloak provisioning has changed (unified realm + some change in microservices user and passwords).
  - You need to clean the `keycloak` schema
  - You have to take integrate the new realm.json in your docker-compose and helm deployments

Changes in Dependencies

Web:

```
"karma-junit-reporter": "2.0.1",
```

DBOS 3.3.0 3.4.1

This Gene release uses DBOS 3.4.1. This means that your workers will be compiled and run against this DBOS version. See the DBOS migration guide at https://dbos-documentation.public.decisionbrain.cloud/migration/.

Other Dependency Changes

New Dependencies

- "com.querydsl:querydsl-jpa": 4.4.0
- `"ag-grid-enterprise": "26.0.0"`

Version Changes

- Keycloak 8.0.1 14.0.0
- keycloak-js 8.0.1 14.0.0
- keycloak-angular 8.4.0
- `"ag-grid-community": "26.1.0"`
- `"ag-grid-angular": "26.1.0"`

```
- "@cds/angular": "5.5.0"
  "@clr/angular": "5.5.0"
  "@clr/ui": "5.5.0",
  "@clr/icons: "5.5.0"
  "@cds/core": "5.5.0"
```

- Spring boot 2.5.3 (latest)
- Spring cloud 2020.0.3 (latest)
- Kotlin 1.5.21 (latest)
- OWASP dependency-check 6.2.2 (latest)

Dependencies Removed