


DOC 4.0.1 FP2 (and DBOS 3.3.0) Migration Guide

Migration Guide from DOC 4.0.1 FP1

Table of Contents

- [Migration Procedure](#)
- [Major or Breaking Changes](#)
 - [DBOS 3.2.2 3.3.0](#)
 - [Changes in Docker images](#)
 - [The new base image for the 'web' image](#)
 - [Custom Docker image for Keycloak](#)
 - [Predefined Docker base for CPLEX users](#)
 - [Changes in Docker Compose scripts](#)
 - [JDL String SQL Script](#)
 - [Change in JDL Generated Code \(gene-model\)](#)
 - [Breaking changes](#)
 - [Removal of Legacy Excel Import/Export](#)
 - [Changes needed](#)
 - [Removal of "entities-extended.jdl"](#)
 - [API Refactoring](#)
 - [Execution Service](#)
 - [Backend Service](#)
 - [Data Service](#)
 - [UI API](#)
 - [Changes in Build Scripts](#)
 - [Changes in the management of multiple occurrences of the project version](#)
 - [In "python-library.gradle" script](#)
 - [In module 'gene-model:gene-model-jpa'](#)
 - [In module 'extensions:backend-service-extension'](#)
 - [In module 'web'](#)
 - [Changes in Configuration \(configuration classes and application.yml files\)](#)
 - [All Services](#)
 - [Data Service](#)
 - [Backend Service](#)
 - [Gateway Service](#)
- [Changes in Dependencies](#)
 - [New Dependencies](#)
 - [Gradle plugins](#)
 - [Back-end](#)
 - [Web Front-end \(package.json\)](#)
 - [Version Changes](#)
 - [Code generation \(backend + frontend\)](#)
 - [Back-end](#)
 - [Web Front-end \(package.json\)](#)
 - [Removed Dependencies](#)

Migration Procedure

 **IMPORTANT:** The recommended way to perform a migration is described below.

Major or Breaking Changes

DBOS 3.2.2 3.3.0

This Gene release uses DBOS 3.3.0. 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/>

Note that these changes have slight impacts on the `application.yml` files of workers.

Changes in Docker images

The new base image for the 'web' image

The Docker image for the web front-end is now based on `nginx:1.19.8-alpine` (as opposed to `nginx:1.19.8` previously).

Custom Docker image for Keycloak

We switch back to a DecisionBrain-made image for Keycloak. The image is tagged after the release, that is, `REGISTRY/db-gene/keycloak:4.0.1-fp2`. It is based on `jboss/keycloak:8.0.1`.

As a consequence, the Gene theme for the Keycloak login page is now embedded in this custom Docker image. It no longer has to be mounted as a volume in the Docker Compose scripts. Furthermore, the theme tarball and scripts are no longer part of the scaffolding (you can remove them from your project if you did not migrate by re-scaffolding).

The realm is still mounted as a volume so that you can customize it. The changes in this file are:

- The Keycloak user associated with the service is renamed from `backend` to `backend-service`.
- The Keycloak user associated with the service is renamed from `execution` to `execution-service`.
- An SPI has been added as an event listener to enable session monitoring.

All these changes result in the following changes in the `infra` Docker Compose script.

```
keycloak:
  image: ${GENE_DOCKER_REGISTRY}/keycloak-theme:8.0.1
  networks:
    - <%= projectName %>
  container_name: <%= projectName %>-keycloak
  ports:
    - "9090:8080"
  expose:
    - 8080 # Keycloak web console
    - 9090 # Wildfly management console
  environment:
    - DB_VENDOR=postgres
    - DB_ADDR=postgres
    - DB_DATABASE=keycloak
    - DB_USER=keycloak
    - DB_PASSWORD=keycloak
    - KEYCLOAK_USER=admin
    - KEYCLOAK_PASSWORD=admin
    - KEYCLOAK_FRONTEND_URL=http://localhost:9090/auth # Enable backend services access f
    - KEYCLOAK_IMPORT=/tmp/realm.json
  volumes:
    - ./keycloak/unpack-gene-theme.sh:/opt/jboss/startup-scripts/unpack-gene-theme.sh
    - ./keycloak/gene-theme.tar.gz.b64:/tmp/gene-theme.tar.gz.b64
    - ./keycloak/realm.json:/tmp/realm.json
  command: [
    "-Dkeycloak.migration.action=import",
    "-Dkeycloak.migration.provider=singleFile",
    "-Dkeycloak.migration.file=/tmp/realm.json",
    "-Dkeycloak.migration.strategy=IGNORE_EXISTING"
  ]
  links:
    - postgres
  depends_on:
    - postgres
mongo:
  image: ${GENE_DOCKER_REGISTRY}/mongo/mongo:4.2.1
  container_name: <%= projectName %>-mongo
  networks:
keycloak:
  image: ${GENE_DOCKER_REGISTRY}/db-gene/keycloak:<%= dbGeneVersion %>
  networks:
    - <%= projectName %>
  container_name: <%= projectName %>-keycloak
  ports:
    - "9090:8080"
  expose:
    - 8080 # Keycloak web console
    - 9090 # Wildfly management console
  environment:
    - CLIENT_ID=gene-web
    - INCLUDE_GROUPS=GENE_USERS
    - RABBIT_USERNAME=guest
    - RABBIT_PASSWORD=guest
    - DB_VENDOR=postgres
    - DB_ADDR=postgres
    - DB_DATABASE=keycloak
    - DB_USER=keycloak
    - DB_PASSWORD=keycloak
    - KEYCLOAK_USER=admin
    - KEYCLOAK_PASSWORD=admin
    - KEYCLOAK_FRONTEND_URL=http://localhost:9090/auth # Enable backend
  volumes:
    - ./keycloak/realm.json:/tmp/realm.json
  command: [
    "-b", "0.0.0.0",
    "-Dkeycloak.migration.action=import",
    "-Dkeycloak.migration.provider=singleFile",
    "-Dkeycloak.migration.file=/tmp/realm.json",
    "-Dkeycloak.migration.strategy=IGNORE_EXISTING"
  ]
  links:
    - postgres
    - rabbitmq
  depends_on:
    - postgres
    - rabbitmq
```

Also, the `realm.json` `displayName` value has been updated to "IBM Decision Optimization Center".

This value is now used in the login title:

IBM Decision Optimization Center

Powered by DecisionBrain

Username or email

Password

Predefined Docker base for CPLEX users

CPLEX users can use predefined docker images to build their engine workers. Access to those images is restricted by the licensing terms of the products.

The following example shows how to build a worker that contains a jar compiled with CPLEX:

```
FROM cplex-registry.decisionbrain.cloud/cplex/cplex-cpo:12.10-jdk11.0.11-2.0.2

# copy the worker jar file
COPY *.jar app.jar

ENV SERVER_PORT 8080
EXPOSE $SERVER_PORT
ENTRYPOINT [ "java", "-jar", "app.jar" ]
```

This second example shows how to build a worker that contains a jar compiled with OPL:

```
FROM cplex-registry.decisionbrain.cloud/cplex/opl:12.10-jdk11.0.11-2.0.2

# copy the worker jar file
COPY *.jar app.jar

ENV SERVER_PORT 8080
EXPOSE $SERVER_PORT
ENTRYPOINT [ "java", "-jar", "app.jar" ]
```

Changes in Docker Compose scripts

As a consequence of the above DBOS upgrade, a few environment variables have changed in the Docker Compose scripts:

- SERVICES_DBOSMASTER_HOST and SERVICES_DBOSMASTER_PORT are replaced with OPTIMSERVER_URL
- MASTER_URL is replaced with OPTIMSERVER_URL
- MASTER_JWTKEY is replaced with OPTIMSERVER_JWT_JWTKEY

The scenario service needs an additional environment variable:

- SERVICES_SESSIONTRACKING_MONGODB_HOST=mongo

The MongoDB service also adds the following environment variables:

```
MONGODB_SESSION_TRACKING_USER=session-tracking
MONGODB_SESSION_TRACKING_PASSWORD=session-tracking
MONGODB_SESSION_TRACKING_DATABASE=session-tracking-db
```

Also, the following line is added to deployment/docker/infra/mongo/addUser.sh:

```
mongo -u $MONGO_INITDB_ROOT_USERNAME -p $MONGO_INITDB_ROOT_PASSWORD --
eval "db.getSiblingDB('$MONGODB_SESSION_TRACKING_DATABASE').createUser
({user: '$MONGODB_SESSION_TRACKING_USER', pwd:
'$MONGODB_SESSION_TRACKING_PASSWORD', roles: [{role: 'dbOwner', db:
'$MONGODB_SESSION_TRACKING_DATABASE' }]}),{w: 'majority',wtimeout:
5000});"
```

JDL String SQL Script

All Strings are now non-limited in size if it's not specified. Before a String was by default limited to 255 chars.

To keep the previous behavior, set the length of all the String fields in the JDL to 255.

In the SQL script, a String is now a "text" without size limitation, the limit will be now validated by the checkers.

Also, the entity internal id and scenario id was a VARCHAR(36) there are now "text".

Change in JDL Generated Code (gene-model)

Add "required" support for relations: Activity{ style required } to Style{ activities }

Annotations added:

- @UniqueConstraint
- @DefaultValue
- @Description

default.attribute.value is now deprecated consider the use of @DefaultValue

New types are supported:

- LocalDate
- LocalTime
- LocalDateTime
- Duration
- Binary

Breaking changes

Instant are now generated in `java.time.Instant` instead of `java.util.Date` in the DOM Collector

String have now infinite size, to have a limit use `id String length(32)`

Removal of Legacy Excel Import/Export

With 4.0.1-fp1, Gene had two Excel import/export libraries wrapped to the corresponding tasks. Starting with 4.0.1-fp2, the legacy OXM Excel library is no longer used.

Task Name	FP1 Library	FP2 Library
ExcellImportTask	Legacy OXM	Simple Excel API
SimpleExcellImportTask	Simple Excel API	Simple Excel API
ExcelExportTask	Legacy OXM	Simple Excel API
SimpleExcelExportTask	Simple Excel API	Simple Excel API

Changes needed

Since the date format is not handled in the same way between legacy Excel and new Excel processing, you may need to update your excel-mapping.yml file to specify the date format you are using (Only in case of date formatted in String). Please refer to the “gene-data-import-from-excel” documentation page.

Removal of “entities-extended.jdl”

Starting with this version, Gene no longer supports `xxxx-extended.jdl`. This file was stored in the `gene-model/spec` directory beside the project JDL file and was used to augment the data model with `Genelssue` and `GeneParameter` entities. This file has been moved inside Gene libraries and is no longer editable in the project.

If your project still has a `xxxx-extended.jdl` the following warning will be displayed at the startup of data-service:

```
2021-04-18 00:01:56.895 INFO 32314 --- [nio-8082-exec-9] c.d.g.d.s.metamodel.MetaModelService : Loading Application Meta-model from JDL file...
2021-04-18 00:01:59.796 WARN 32314 --- [nio-8082-exec-9] c.d.gene.data.service.jdl.JdlService : Extended JDL is no longer supported since 4.0.1-fp2, you need to use a single JDL File. 'entities-extended.jdl' JDL has been ignored.
2021-04-18 00:01:59.883 INFO 32314 --- [nio-8082-exec-9] c.d.g.d.s.metamodel.MetaModelService : Meta-model loaded
```

Similarly, if you try to scaffold a new project with `xxxx-extended.jdl` the following warning will also be displayed:

```
mi
*****
Gene CLI Project Generator
Copyright (c) 2020 DecisionBrain SAS. All Rights Reserved.
*****
WARNING: Extended JDL is no longer supported since 4.0.1-fp2, you need to use a single JDL File. './../jdl-templates/entities-extended.jdl' has been ignored.
force gene-sample/.gitattributes
force gene-sample/Jenkinsfile
```

The platform also now generates in addition to the project's Flyway script a second script **V0.1.000** which creates Gene internal tables.

```
2021-04-17 16:30:15.300 INFO 13005 --- [main] o.f.s.i.s.jdbcTableSchemaHistory : Creating Schema History table "public"."flyway_schema_history" ...
2021-04-17 16:30:15.410 INFO 13005 --- [main] o.f.core.internal.command.DbMigrate : Current version of schema "public": « Empty Schema »
2021-04-17 16:30:15.415 INFO 13005 --- [main] o.f.core.internal.command.DbMigrate : Migrating schema "public" to version 0.1.000 - CREATE GENE TABLES
2021-04-17 16:30:15.477 INFO 13005 --- [main] o.f.core.internal.command.DbMigrate : Migrating schema "public" to version 1.0.000 - CREATE TABLES AND CONSTRAINTS
2021-04-17 16:30:15.724 INFO 13005 --- [main] o.f.core.internal.command.DbMigrate : Successfully applied 2 migrations to schema "public" (execution time 00:00.315s)
```

As a consequence, the `gene-model/gene-model-jpa` module must have the following additional dependency:

```
api "com.decisionbrain.gene:data-model-jpa-base:${version.decisionbrain.dbgene}"
```

If this dependency is missing the following error will happen when building the project:

```
error: package com.decisionbrain.gene.data.jpa.repository does not exist
import com.decisionbrain.gene.data.jpa.repository.DataServerRepository;
```

API Refactoring

Execution Service

The `id` field of the `JobStatusChange` type has been renamed into `jobId`.

In class `ExecuteRoutineStatement`, the method `onBackendService()` is deprecated in favor of method `of()`.

Backend Service

The `com.decisionbrain.gene.execution.model.script.FileValue` class is now in package `com.decisionbrain.platform.run.types`. This may impact the imports of your routines, if they use it.

Routines (that is, their implementation classes) should now be annotated with `@Routine(name = ...)`. The provided name is to be used in the task script, namely in `ExecuteRoutineStatement.of()`. Note that if you do not annotate a routine, or annotate it with `@Routine` with no arguments, the fully-qualified name of its class is used as its name; this means that existing code still works.

Data Service

The transactional API with conflict detection no longer detects conflicts of type **ADDING_EXISTING_ROW**. Because of the soft data model adding rows with the same business key is now allowed, and is detected by schema checkers.

UI API

- Some enum has been renamed to include an underscore as word separator:
 - `TaskInputType.SCENARIO_ID`
 - `TaskInputType.WORKSPACE_ID`
 - `NotificationType.SCENARIOS_CHANGE`
 - `NotificationType.STEPS_CHANGE`
 - `NotificationType.STATUS_CHANGE`
- `GeneScenarioEventTypedeprecated` and replaced by `ScenarioNotificationType` (openapi generated)
- Icons: Gene now uses the new Clarity Icon API <https://clarity.design/foundation/icons/>. This means that all `<clr-icon></clr-icon>` should be now replaced by `<cds-icon></cds-icon>`. Not that all Clarity icons previously available remain available, so you don't need extra import.
- The classes from filter API have been moved from `@gene/widget-data` to `@gene/data` library, as a consequence the imports need to be updated.

```
// 4.0.1-fp1
import { GeneBaseObjectFilterBuilder, GeneContextBuilder,
GeneEntityAttributeFilterBuilder, GeneEntityFilterBuilder,
GeneEntityFilterCriteriaBuilder, GeneFilterClauseBuilder
GeneObjectFilterBuilder } from '@gene/widget-data';

// 4.0.1-fp2
import { GeneBaseObjectFilterBuilder, GeneContextBuilder,
GeneEntityAttributeFilterBuilder, GeneEntityFilterBuilder,
GeneEntityFilterCriteriaBuilder, GeneFilterClauseBuilder
GeneObjectFilterBuilder } from '@gene/data';
```

Changes in Build Scripts

Changes in the management of multiple occurrences of the project version

The version of the project is defined by the `version` property in the root `build.gradle` file. However, it may be required to replicate it at various locations. The build scripts that we provide take care of keeping the version properly updated in the `web/package.json` file and in the `processing/python-engine/setup.py` file.

For this, starting with 4.0.1-fp2, we use DecisionBrain's "code-replicate-plugin" to manage the multiple occurrences of the project version. This impacts:

- The root `build.gradle`: added the classpath for the plugin
- The script `versions.gradle`: added a version number for the plugin, removed functions `ext.toPythonVersion` and `ext.getProjectPythonVersion`
- The script `processing/python-engine/build.gradle`:
 - `toPythonVersion` has been replaced with `VersionUtils.javaVersionToPythonVersion` with the import: `import com.decisionbrain.gradle.plugin.codereplicate.VersionUtils.`
 - The following dependency has been removed: `tasks.checkPythonRequires.dependsOn tasks.alignPythonRequires`

In “python-library.gradle” script

Too many changes should be retrieved from a new scaffolding.

In module ‘gene-model:gene-model-jpa’

The `build.gradle` file has been updated to fix the build sequence. Here is the modified part (dependency for ‘processResources’ added):

```
// Un-comment the next line and comment the next one if you want to
generate SQL flyway migration script
// Un-comment the next two lines and comment the next ones if you want
to generate SQL flyway migration script
// at each compilation.
//compileJava.dependsOn tasks.generateJPAForce
//processResources.dependsOn tasks.generateJPAForce
compileJava.dependsOn tasks.generateJPA
processResources.dependsOn tasks.generateJPA
```

Also, in the dependencies `gene-model-base` and `data-service-base` have been replaced with `data-extension-support` and `data-model-jpa-base`.

In module ‘extensions:backend-service-extension’

A dependency has been added to process routine annotations:

```
annotationProcessor "com.decisionbrain.gene:execution-routine-
base:${versions.decisionbrain.dbgene}"
```

In module ‘web’

The `inputs.files` from `yarn_build` task has been updated to improve incremental builds.

```
yarn_build { // part of 'compile' step
    args = ['--prod']
    inputs.files(fileTree(".").exclude("**/dist/**", "**/*.log", "**/
coverage/**", "**/karma*", "**/node_modules/**"))
    outputs.dirs("./dist")
}
```

Changes in Configuration (configuration classes and application.yml files)

All Services

All services may activate logs to debug access control issues. The loggers to activate used to be named after the implementation classes. This has been simplified by giving implementation-independent names to loggers. As a consequence, these logs are activated by adding (or uncommenting) the following lines in the `application.yml` files of the micro-services of your interest:

```
logging.level:
  # Log operations related to permission evaluation
  com.decisionbrain.gene.security.DBG_PERMISSIONS: DEBUG
  # Log operations related to scoped identity
  com.decisionbrain.gene.security.DBG_IDENTITY: DEBUG
```

The same effect can be achieved with the following environment variables. This is usually more suited when using Docker images.

```
LOGGING_LEVEL_COM_DECISIONBRAIN_GENE_SECURITY_DBG_PERMISSIONS=DEBUG
LOGGING_LEVEL_COM_DECISIONBRAIN_GENE_SECURITY_DBG_IDENTITY=DEBUG
```

Data Service

The `spring.jpa.database-platform` entry has been updated to `com.decisionbrain.gene.data.jpa.GenePostgreSQLDialect`

If you have this entry in the `application.yml` file of your data service extension, do not forget to update it.

Backend Service

The `configuration/DbDomCollectorConfiguration` class has been updated. It used to contain two methods that created the beans `collectorFactory` and `entityRegister`. The creation of these beans has been moved elsewhere and is activated by the `@EnableXYZDom` annotation, where `XYZ` is the name of the collector as defined during the scaffolding.

Gateway Service

Content Security Policy has been enabled on generated applications. This is controlled by the `spring.cloud.gateway.filter.secure-headers.content-security-policy` property in `extensions/gateway-service-extension/.../application.yml`. It is configured with recommended values, which are described in comments in this file.

Changes in Dependencies

New Dependencies

Gradle plugins

- `com.decisionbrain:code-replicate-plugin:0.1.2`
- `com.jayway.jsonpath:json-path:2.5.0`

Back-end

- `"io.springfox:springfox-core": "3.0.0"`
- `"org.testcontainers:postgresql": "1.15.3"`
- `"org.testcontainers:junit-jupiter": "1.15.3"`
- `com.google.auto.service:auto-service:1.0`
- `com.google.auto.service:auto-service-annotations:1.0`

- com.google.testing.compile:compile-testing:0.19 (for tests only)
- "org.hibernate.validator:hibernate-validator":"6.1.5.Final" (needed in execution-service to load OptimServerClientProperties Configuration)
- "com.rabbitmq:amqp-client" : "5.12.0"
- "org.slf4j:slf4j-simple" : "1.7.30"
- "com.github.jengelman.gradle.plugins:shadow": "5.2.0"

Web Front-end (package.json)

- "@cds/angular": "5.3.0"
- "luxon": 1.27.0
- "@types/luxon": 1.26.5
- "html-duration-picker": 2.2.13

Version Changes

Code generation (backend + frontend)

- "org.openapitools:openapi-generator-gradle-plugin": "4.3.1" "5.1.0"

Back-end

- "org.apache.commons:commons-text": 1.6 1.9
- "org.apache.commons:commons-lang3" removed from versions.gradle, the version is defined by the Spring BOM 3.10
- "org.apache.poi:poi-ooxml": 4.0.0 5.0.0
- "com.univocity:univocity-parsers" (CSV library): 2.8.4 2.9.1
- "org.slf4j:slf4j-jdk14": 1.7.26 1.7.30
- "decisionbrain:dbos": 3.2.2 3.3.0-RC2

Web Front-end (package.json)

```
"@cds/core": "5.0.2" -> "5.3.0"
"@clr/angular": "5.0.2" -> "5.3.0"
"@clr/ui": "5.0.2" -> "5.3.0"
@angular-devkit/build-angular": "~0.1102.1" -> "0.1102.1" (fix frontend
build issue)
```

Removed Dependencies

Web Frontend (package.json)

- "@clr/icons": "5.0.2"