



Migrate a DOC Application from 4.0.1 to 4.0.1 FP1

March 23, 2020

Porting from 4.0.1 - FP4 to 4.0.1-FP1	2
Libraries and Version Changes	2
Major Changes	3
Web UI	3
Naming Changes	3
Application Configuration changes :	4
Gradle changes:	5
Soft Data Model	5
Excel Import/Export	5
Breaking Changes	6
Permissions	6
Application Startup - Default Configuration File:	6
Application Initialization API	7
Collector API	7

Porting from 4.0.1 - FP4 to 4.0.1-FP1

This document describes how to port a 4.0.0 FP4 application into a 4.0.1 application.

Libraries and Version Changes

New Dependencies Web (to be added in web/package.json)

- angular-split 5.0.0

Dependency Updates Web (to be updated in web/package.json)

```

"@angular/animations": "9.1.0" -> "11.2.1"
"@angular/cdk": "9.1.0" -> "11.2.1"
"@angular/common": "9.1.0" -> "11.2.1"
"@angular/compiler": "9.1.0" -> "11.2.1"
"@angular/core": "9.1.0" -> "11.2.1"
"@angular/forms": "9.1.0" -> "11.2.1"
"@angular/google-maps": "9.1.0" -> "11.2.1"
"@angular/localize": "9.1.0" -> "11.2.1"
"@angular/platform-browser": "9.1.0" -> "11.2.1"
"@angular/platform-browser-dynamic": "9.1.0" -> "11.2.1"
"@angular/router": "9.1.0" -> "11.2.1"

"@clr/core": "3.1.0" -> "@cds/core": "5.0.2" (NAME CHANGED)
"@clr/angular": "3.1.0" -> "5.0.2"
"@clr/icons": "3.1.0" -> "5.0.2"
"@clr/ui": "3.1.0" -> "5.0.2"

"@ng-select/ng-select": "4.0.4" -> "6.1.0",

"@ngx-translate/core": "12.1.2" -> "13.0.0",
"@ngx-translate/http-loader": "4.0.0" -> "6.0.0",

```

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

"graphql": "14.2.0" -> "15.5.0"	24
"moment": "2.27.0" -> "2.29.1",	25
	26
"ngx-color-picker": "9.1.0" -> "11.0.0",	27
"ngx-echarts": "4.2.1" -> "4.2.2",	28
"ngx-quill": "11.0.0" -> "13.2.0",	29
	30
"rxjs": "6.5.5" -> "6.6.3",	31
	32
"tslib": "1.11.1" -> "2.1.0",	33
"uuid": "7.0.1" -> "8.3.2",	34
"zone.js": "0.10.3" -> "0.11.4",	35
"angular-split": "5.0.0" (NEW)	

The report of node dependencies after angular migration is available here:

<https://nexus.decisionbrain.cloud/repository/dbgene-dependency-reports/4.0.1-fp1-rc7/dependencies/node-dependencies.txt>

Engine Worker Dockerfile

The engine worker default Dockerfile has changed: instead of relying on an ubuntu base, we are now using the adoptopenjdk:11-jre-hotspot base.

Major Changes

Web UI

Naming Changes

GeneIssue class was moved from @gene/common-widget library to @gene/data

In web/src/app/gene-lib/routing/gene-routing.config.ts :

GeneNoWorkspaceGuardGuard component was renamed to
GeneNoWorkspaceGuard

In web/src/app/gene-lib/routing/gene-routing-wrappers.module.ts :

GeneErrorPagesModule imported from library @gene/layout was renamed to GeneErrorModule

GeneChartFilterService was removed :

- instead of filterFieldsToFilterValues use this code on the GeneChartConfiguration :

```

1 // equivalent to filterFieldsToFilterValues()
2
3 const filterValues = configuration.filterFields.map(filter => {
4
5     return {
6
7         ...filter,
8
9         value: null,
10
11         allValues: []
12
13     };
14
15 });

```

- instead of getFilterValues(...) use the GeneDataService's .findDistinctValues()

```

1 // instead of
2
3 // this.filterService.getFilterValues(context, this._widgetConfiguration.dataType, f.fieldName, 100)
4
5 // use this, note maxItems parameter does not exist on this method
6
7 this.dataApi.findDistinctValues(context, this._widgetConfiguration.dataType, f.fieldName)
8
9

```

Application Configuration changes :

You have to update configurations files stored under web/src/environments/

there were changes in some endpoints, and a new endpoint was added so the up-to date values are :

1

```
// ...

scenarioServerAppConfigExport: '/api/scenario/application-configuration',
scenarioServerAppConfigImport: '/api/scenario/application-configuration',
scenarioServerManagementEndpoint: '/api/scenario/permissions',

// ...

importDbrfEndpoint: '/api/data/dbrf-import',

// ...

exportDbrfEndpoint: '/api/data/dbrf-export',

// ...

excelTemplateEndpoint: '/api/data/excel/excel-template', // NEW

// ...
```

Gradle changes:

In gene-model/gene-model-dto/build.gradle

Replace com.decisionbrain.gene:data-service-dto-base by
com.decisionbrain.gene:data-service-dto

Soft Data Model

Caution: There is no mode data consistency check until the introduction of the checker in the next release.

Excel Import/Export

New Simple Excel file processing has replaced the previous one (aka legacy), through the default Import/Export Excel tasks: SimpleExcelImportTask / SimpleExcelExportTask.

It is faster and has a low memory footprint. It is designed to process large amount of data.

- By doing a migration you will keep the legacy Excel processing. To use the new one edit the default Tasks through the Application preferences, and change the following settings:
 - DEFAULT_EXCEL_IMPORT_TASK_ID: SimpleExcelImportTask
 - DEFAULT_EXCEL_EXPORT_TASK_ID: SimpleExcelExportTask
- By scaffolding a new project, the new Excel processing Tasks are now the default configuration.

It is strongly encouraged to move from the legacy processing to the new one. ExcelExportTask, ExcelImportTask are now deprecated and will no longer use the legacy API in a coming version.

To know the change between the two Excel formats please refer to the Gene documentation in the “Getting started” section “Excel import”.

Breaking Changes

Permissions

The permission system has been simplified, with changes that break the compatibility of its API, and changes to the persistence format. The impact on an application migrating from version 4.0.1 are:

- If the code of the application uses the Java or REST API of the permission system, it may no longer compile or run properly.
- The permission rules stored in the application database cannot be reused. You will have to remove them and redefine them entirely.
- Restoring a 4.0.1 backup of the application configuration will ignore the permission rules in the backup.

Given the changes in the permission system, if you defined complex permission rules, you may want to contact the development team for help in setting things up in FP1.

Application Startup - Default Configuration File:

At application startup, a check is made on the existence of a configuration file as defined in application.yml of scenario-service-base module.

The configuration file is pointed to using the property scenario-management.app-config-file.

By default the property is set to a new file name: app-config.json in 4.0.1-fp1 instead of app-config-file.json previously.

It is possible to override the value of the property scenario-management.app-config-file by adding a new entry to application.yml of scenario-service-extension module.

When migrating from 4.0.1, permission rules are not imported because their format has changed too much. As a consequence, the rules that allow universal access to “Public Workspace” must be redefined by hand.

Application Initialization API

The signatures of the application initialization API has evolved in an incompatible way. This concerns:

- the WorkspaceAndPermissionRulesInitializer interface;
- the ApplicationElementCreationHandler interface.

Collector API

Previously deprecated API have been removed:

```

1 void loadSnapshot(InputStream inputStream, boolean useDbrfFormat) throws IOException, ParseException;
2
3 void loadSnapshot(byte[] zippedBytes, boolean useDbrfFormat) throws IOException, ParseException;
4
5 byte[] saveSnapshot(List<Class<? extends DbDomObject>> classes, boolean writeRelationDbGeneInternalId,
6 boolean useDbrfFormat) throws IOException;
7
8 byte[] saveSnapshot(List<Class<? extends DbDomObject>> classes, boolean writeRelationDbGeneInternalId,
9 DbDomCollectorSerializerFormat format) throws IOException;
10
11 byte[] saveSnapshotFromInterfaceSimpleName(List<String> names, boolean writeRelationDbGeneInternalId,
12 boolean useDbrfFormat) throws IOException;
13
14 void saveSnapshot(OutputStream outputStream, List<Class<? extends DbDomObject>> classes, boolean
15 writeRe

```