



# DB Gene 4.8.0 Migration Guide

December 19th, 2025

Copyright © 2012-2025 DecisionBrain S.A.S. All rights reserved.

All specifications and information regarding the products in this document are subject to change without notice and should not be construed as a commitment by DecisionBrain. DecisionBrain assumes no responsibility or liability for any mistakes or inaccuracies that may appear in this document. All statements and recommendations in this document are believed to be accurate but are presented without warranty. Users must take full responsibility for their application of any product.

# DB Gene 4.8.0 Migration Guide

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| <b>Preliminary notes.....</b>                                            | <b>5</b>  |
| Version 4.8.0.....                                                       | 5         |
| Version 4.7.2.....                                                       | 5         |
| Version 4.7.1.....                                                       | 5         |
| <b>Application Changes.....</b>                                          | <b>6</b>  |
| <b>General Application Changes.....</b>                                  | <b>6</b>  |
| Removed Support for Tableau Integration.....                             | 6         |
| <b>Data Changes.....</b>                                                 | <b>7</b>  |
| <b>Built-in Import/Export.....</b>                                       | <b>7</b>  |
| Improved Scenario Export (4.7.2).....                                    | 7         |
| Improved Exception Management for Scenario Import.....                   | 7         |
| <b>Scenario Service.....</b>                                             | <b>8</b>  |
| Removed REST API in the Scenario Service.....                            | 8         |
| <b>Data Service.....</b>                                                 | <b>8</b>  |
| Improved Schema Checkers (4.7.1).....                                    | 8         |
| <b>Execution Service.....</b>                                            | <b>9</b>  |
| Removed Expressions in the Execution Service.....                        | 9         |
| Deprecated Methods in the Execution Service.....                         | 9         |
| Improved Scenario Data Management in the Execution Service.....          | 9         |
| <b>Backend Service.....</b>                                              | <b>10</b> |
| Deprecated Scenario Data DTO in the Backend Service (Internal).....      | 10        |
| Improved Scenario Data Management in the Backend Service (Internal)..... | 10        |
| <b>DBOS Changes.....</b>                                                 | <b>11</b> |
| <b>Master.....</b>                                                       | <b>11</b> |
| Removed Parameters in the DBOS REST API.....                             | 11        |
| <b>Workers (Java).....</b>                                               | <b>12</b> |
| Improved Interface for DBOS Tasks.....                                   | 12        |
| <b>Dev Changes.....</b>                                                  | <b>13</b> |
| <b>REST API.....</b>                                                     | <b>13</b> |
| Gateway now provides rate limit protection.....                          | 13        |
| <b>3rd-Party Components.....</b>                                         | <b>14</b> |

|                                                                                 |           |
|---------------------------------------------------------------------------------|-----------|
| Updated Dependencies.....                                                       | 14        |
| Updated Angular Dependencies.....                                               | 18        |
| Updated Java JDK Dependencies.....                                              | 21        |
| Updated Gradle Dependencies.....                                                | 21        |
| <b>Security.....</b>                                                            | <b>21</b> |
| Removed Methods (Internal).....                                                 | 21        |
| Improved Base Images.....                                                       | 21        |
| Improved Configuration of the Content Security Policy.....                      | 21        |
| <b>Build.....</b>                                                               | <b>22</b> |
| Updated Requirements for GitLab CI Builds (Internal).....                       | 22        |
| Improved Gradle Files.....                                                      | 22        |
| Improved DSL for the Gradle Plug-In CodeReplica (Internal).....                 | 22        |
| <b>Deployment.....</b>                                                          | <b>23</b> |
| Improved Endpoint Probes for Kubernetes.....                                    | 23        |
| <b>JupyterLab.....</b>                                                          | <b>23</b> |
| New Jupyter Notebook Sample Using the HiGHS Solver (4.7.1).....                 | 23        |
| <b>Scripted Tasks Changes.....</b>                                              | <b>24</b> |
| <b>Definition.....</b>                                                          | <b>24</b> |
| Removed Methods and Task (4.7.2).....                                           | 24        |
| <b>UI Changes.....</b>                                                          | <b>25</b> |
| <b>Extensibility.....</b>                                                       | <b>25</b> |
| Improved Modals API.....                                                        | 25        |
| <b>Tables.....</b>                                                              | <b>27</b> |
| Removed Method (4.7.2).....                                                     | 27        |
| <b>Data Grid/Explorer Widgets.....</b>                                          | <b>27</b> |
| Removed Pagination in the Data Grid/Explorer Widgets (4.7.2).....               | 27        |
| Improved Documentation in the Data Grid/Explorer Widgets Custom Controller..... | 27        |
| <b>Chart Widgets.....</b>                                                       | <b>28</b> |
| Removed Option for Pie Charts (4.7.2).....                                      | 28        |
| <b>Gantt Chart Widget.....</b>                                                  | <b>28</b> |
| Improved Sample Custom Controller for the Gantt Chart Widget (4.7.2).....       | 28        |
| Improved Tooltip Service (Internal).....                                        | 28        |
| <b>Filter Widget.....</b>                                                       | <b>28</b> |
| Removed Property in the Filter Widget Custom Controller (4.7.2).....            | 28        |
| <b>Pivot Table Widget.....</b>                                                  | <b>29</b> |

---

|                                                            |           |
|------------------------------------------------------------|-----------|
| Improved Custom Controller for the Pivot Table Widget..... | 29        |
| New Value Formatting for the Pivot Table Widget.....       | 29        |
| <b>Deprecated Features and APIs</b>                        |           |
| <b>Scheduled for Removal in 4.9.0.....</b>                 | <b>30</b> |

- Each section is structured as follows: Removed, Deprecated, New, Improved, and Updated.
- Dependency updates are listed in Section [Dev Changes](#) > [3rd-Party Components](#).
- Future removals are listed in Section [Deprecated Features and APIs Scheduled for Removal in 4.9.0](#).

For more details, please refer to the [DB Gene Documentation and Release Notes](#).

---

# Preliminary notes

---

*This part lists any attention points for migrations from previous versions.*

---

## Version 4.8.0

- [DBPF-9535](#) MongoDB v8.2.2 contains a high-severity CVE. **Fixed in 4.8.1.**
- [DBPF-9184](#) PostgreSQL updated from 15.4 to 18.1.



To preserve scenario data, follow the migration process for PostgreSQL described in Section [Dev Changes](#) > [3rd-Party Components](#).

## Version 4.7.2

- [DBPF-9480](#) The Keycloak configuration might trigger an infinite loop in the user/workspace initialization process. **Fixed in 4.8.0.**

## Version 4.7.1

- [DBPF-9093](#) [Regression] The CLEAR button in the column filters of the Data Grid does not work. **Fixed in 4.7.2.**
  - [DOC-1317](#) [Regression] Formatting values in the Data Grid causes values to become NULL. **Fixed in 4.7.2.**
  - [DBPF-9200](#) [Regression] Charts comparison throws an error in non-CDM environments. **Fixed in 4.7.2.**
-

---

# Application Changes

---

*This part lists all changes related to:*

- *General features on application element management, including events and hooks, in Section General Application Changes.*
- 

## General Application Changes

*This section lists all changes related to general features of the application element management, including events and hooks.*

### Removed Support for Tableau Integration

#### [DBPF-9167](#)

Tableau integration has been removed from DB Gene. Users can no longer synchronize DB Gene data with a Tableau Server or display Tableau visualizations within DB Gene dashboards. If Tableau remains part of your workflow, you can still export DB Gene data and manage it manually within Tableau.

---

---

# Data Changes

---

*This part lists all changes related to:*

- The import or export mechanism provided by the product, in Section **Built-in Import/Export**.
  - The Data Service API, in Section **Data Service**.
  - The Scenario Service API, in Section **Scenario Service**.
  - The Execution Service API, in Section **Execution Service**.
  - The Backend Service API, in Section **Backend Service**.
- 

## Built-in Import/Export

*This section lists all changes related to the import or export mechanism provided by the product.*

### Improved Scenario Export (4.7.2)

[DOC-1060](#) ([DBPF-9043](#))

The scenario export dialog has been reworked to use the scenario name by default.

### Improved Exception Management for Scenario Import

[DOC-1379](#) ([DBPF-9306](#))

The import service now stops immediately when encountering connection errors, for instance, a PostgreSQL database being unavailable (e.g., down or unreachable).

Before, instead of throwing an immediate connection error, the service attempted to process every row of the table. This could generate continuous, high-volume error logging, making it difficult to identify the original root cause (the database being offline).

Now, there is only one log related to the error, and the job completes with the **ALERTING** status.

To filter which table to import into the Data Service, **DataClient#importScenarios** now has a new filter parameter; if **null** or empty, all tables will be imported.

---

## Scenario Service

*This section lists all changes related to the Scenario Service API.*

### Removed REST API in the Scenario Service

[DBPF-9279](#)

**GET /scenario/scenariosIdsAndNames** API in the Scenario Service was deprecated in 4.1.0 and has been removed. **POST /scenario/scenariosIdsAndNames** API should be used instead.

---

## Data Service

*This section lists all changes related to the Data Service API.*

### Improved Schema Checkers (4.7.1)

[DBPF-9082](#)

The following schema checkers' severity can now be configured through properties or environment variables:

*Min Value*

**services.data.schema-checkers.bounds.min.severity: ERROR (default)**

*Max Value*

**services.data.schema-checkers.bounds.max.severity: ERROR (default)**

*Max Length*

**services.data.schema-checkers.length.string-max-length.severity: ERROR (default)**

One common use case of changing those severities is to reduce their severity from **ERROR** to **WARNING** so that affected scenarios can still be used with the scenario comparison feature.

---

## Execution Service

*This section lists all changes related to the Execution Service API.*

### Removed Expressions in the Execution Service

#### [DBPF-9040](#)

The following expressions have been removed from the Execution Service:

- `StringExpression#idOfNewScenarioWithMetadata(Expression file, Expression scenarioName, Expression workspaceId, Expression folderId)`
- `StringExpression#idOfNewScenarioWithMetadata(Expression file, Expression scenarioName, Expression workspaceId, Expression folderId, Expression lockIt)`
- `StringExpression#idOfNewScenarioWithFolder(Expression scenarioName, Expression workspaceId, Expression folderId)`

### Deprecated Methods in the Execution Service

#### [DBPF-9280](#)

The following methods have been marked as deprecated in the Execution Service:

- **ScenarioData**
  - `getRawData` -- `asFile` or `asInputStream` can be used instead
  - `setRawData` -- `asOutputStream` can be used instead
  - `getIssues` – no replacement
  - `resetModifiedTables` – no replacement
  - `hasErrorsOrWarnings` – no replacement
- **ScenarioDataExpression**
  - `hasErrorsOrWarnings` – no replacement
  - `issues` – no replacement

### Improved Scenario Data Management in the Execution Service

#### [DOC-1122](#) ([DBPF-9062](#), [DBPF-9063](#)), [DBPF-9047](#)

To minimize memory consumption, the Execution Service now:

- Streams scenario data from the **Data Service** when creating a **DbosJobInput** for DBOS.
- Passes scenario references instead of full data to the Backend Service.
- Uses new DBOS REST APIs to reduce DBOS Master memory consumption.

Also, **ScenarioData** is now an interface, not a class.

## Backend Service

*This section lists all changes related to the Backend Service API.*

### Deprecated Scenario Data DTO in the Backend Service (Internal)

[DBPF-9064](#)

The **ScenarioData** DTO is now **@Deprecated**.

### Improved Scenario Data Management in the Backend Service (Internal)

[DBPF-9064](#)

Data is now streamed and processed more efficiently, enabling even larger scenarios to be handled reliably without causing application instability.

To minimize memory consumption when processing scenarios, the collector objects (the manipulated scenario) are now stored in temporary files when exchanging data with the **Data Service**.

This was handled via the **data** field in the **ScenarioData** DTO, which is now marked as **@Deprecated**. For backward compatibility, its getter should use **RoutineExecutionContext#getCollector** to trigger on-demand data loading if called, serving the data without storing it in the DTO instance memory.

When **RoutineExecutionContext#markModified** is called, the modified collector object is written back to the temporary files.

Upon completion of the routine, the scenario data from the temporary files is streamed to the **Data Service** to be saved, but only if the scenario has been modified.

The temporary files are deleted after the operation is complete (both on success and failure), and a "job finished" notification is sent back to the **Execution Service** with an empty **data** field in the response DTO.

---

---

# DBOS Changes

---

*This part lists all changes related to:*

- The DBOS worker registration, job scheduling & monitoring, and communication with DB Gene, in Section **Master**.
  - The DBOS Java task definition and worker library, in Section **Worker (Java)**.
- 

## Master

*This section lists all changes related to the DBOS worker registration, job scheduling & monitoring, and communication with DB Gene.*

### Removed Parameters in the DBOS REST API

[DBPF-7778](#), [DBPF-5605](#)

The following deprecated endpoints have been removed from the DBOS REST API:

- **POST /files**. Use **POST /datastorage/temporary** instead
- **GET /buckets/{bucketId}**. Use **GET /buckets/{bucketId}/definition** and **GET /buckets/{bucketId}/content** instead

The following fields have been removed; all replacement fields can be filled with the **dataStorageUri** provided by **POST /datastorage/temporary** endpoint:

- **BucketCreation.data** and **BucketCreation.fileId**. Use **BucketCreation.dataStorageUri** instead.
- **JobParameter.data** and **JobParameter.uri**. Use **JobParameter.dataStorageUri** instead.
- **BlobDefinition.uri**. Use **BlobDefinition.dataStorageUri** instead.

In addition, the **ExecutionContext.getBucket** has been replaced by **ExecutionContext.getBucketData**.

Finally, the deprecated parameters **inputs** and **outputs** (type **Parameter**) have been removed from the DBOS **Task** interface.

---

## Workers (Java)

*This section lists all changes related to the DBOS Java task definition and worker library.*

### Improved Interface for DBOS Tasks

#### [DBPF-5605](#)

The method **Task.execute**, used to create a DBOS worker task, has been modified to remove deprecated parameters:

None

```
- void execute(Parameter input, Parameter output, ExecutionContext context);  
  
+ void execute(ExecutionContext context);
```

---

# Dev Changes

---

*This part lists all changes related to:*

- The REST API, except for the Data Integration Framework, Data Service, and Scenario Service API, in Section **REST API**.
  - The external dependencies and libraries, such as Spring, Angular, or Keycloak, in Section **3rd-Party Components**.
  - The security issues, in Section **Security**.
  - The Scaffolding and Gradle scripts, in Section **Build**.
  - The integration to Docker or Helm, in Section **Deployment**.
  - The integration of JupyterLab, in Section **JupyterLab**.
- 

## REST API

*This section lists all changes related to the REST API, excluding the Data Integration Core, Data Service, and Scenario Service APIs.*

### Gateway now provides rate limit protection

#### [DBPF-7357](#)

The Gateway Service supports rate limitation to protect back-end services from excessive or abusive traffic. Rate limitation can be configured to control the number of requests allowed per service within a given time window, and to set burst capacities for short spikes.

Configuration is performed via environment variables or YAML files and can be customized per service or globally using a default profile.

For more details, see the rate limit [configuration guide](#).

---

## 3rd-Party Components

This section lists all changes related to the external dependencies and libraries, such as Spring, Angular, or Keycloak.

### Updated Dependencies

- [DBPF-9180](#) Spring Boot updated from 3.5.5 to 3.5.7, and Spring Framework from 6.2.10 to 6.2.12

 There are likely no changes in the properties for a minor update of Spring Boot. For example, there was no change in DB Gene and DBOS for the default properties. However, **we still recommend running your application with Spring Boot properties migrator once** to ensure that your custom Spring configuration is compatible with this new version of Spring Boot.

- [DBPF-9182](#) Apache Commons Text updated from 1.13.1 to 1.14.0
- [DBPF-9182](#) Apache Commons POI updated from 5.4.1 to 5.5.0
- [DBPF-9185](#) RabbitMQ updated from 4.1.3 to 4.2.1
- [DBPF-9191](#) NGINX updated from 1.28.0 to 1.29.2
- [DBPF-9188](#) AG Grid updated from 34.1.0 to 34.3.0
- [DBPF-8444](#) Keycloak updated from 26.3.1 to 26.4.6
- [DBPF-9183](#) MongoDB updated from 8.0.12 to 8.2.2
- [DBPF-9184](#) PostgreSQL updated from 15.14 to 18.1

 **DB Gene 4.8.0 relies on PostgreSQL 18. Formerly, it was version 15.**

The new directory structure for PostgreSQL is as follows:

**Mount Point:** `/var/lib/postgresql`

**Data Directory (PGDATA):** `/var/lib/postgresql/18/docker`

PostgreSQL does not support automatic in-place upgrades between major versions. See [PostgreSQL Docker Docs](#) for more details.

Make sure to have sufficient disk space on your computer to store the database backups and **upgrade PostgreSQL using Docker Compose or Kubernetes** as follows:

1. Stop the service that connects to the database (data-service and Keycloak)
2. Back up the required schema(s)
3. Deploy the new application (with PostgreSQL v18)
4. Restore the backup
5. Restart the services
6. Clean the files
7. Rollback if needed

### Kubernetes:

1. Stop the service that connects to the database (Data Service and Keycloak).

Shell

```
# Stop the services that connect to the database, but leave the database container running.
kubectl scale deployment data-service --replicas=0
kubectl scale deployment keycloak --replicas=0
```

2. Back up the required schema(s).

Shell

```
# Identify the name of your PostgreSQL 15 pod:
# Example output: postgres-15-abcdef-12345
kubectl get pods | grep postgres

# Use this pod name in the following command:
kubectl exec -it <your-postgres-15-pod> -- bash -c
'PGPASSWORD="$POSTGRES_dataServer_password" pg_dump -Fc --clean --if-exists -U
"$POSTGRES_dataServer_username" --schema=public -d
"$POSTGRES_dataServer_database"' >data_service_backup.dump

kubectl exec -it <your-postgres-15-pod> -- bash -c
'PGPASSWORD="$POSTGRES_keycloak_password" pg_dump -Fc --clean --if-exists -U
"$POSTGRES_keycloak_username" --schema=public -d "$POSTGRES_keycloak_database"'
>keycloak_backup.dump
```

3. Deploy the new application (with PostgreSQL v18).

Shell

```
# Make sure services are down beforehand
kubectl scale deployment data-service --replicas=0
kubectl scale deployment keycloak --replicas=0
```

#### 4. Restore the backup.

Shell

##### # Copy the backup to the volume

```
kubectl cp data_service_backup.dump  
<your-postgres-18-pod>:/tmp/data_service_backup.dump  
kubectl cp keycloak_backup.dump <your-postgres-18-pod>:/tmp/keycloak_backup.dump
```

##### # Restore the backup

```
kubectl exec -it <your-postgres-18-pod> -- bash -c  
'PGPASSWORD="$POSTGRESQL_dataServer_password" pg_restore -d  
"$POSTGRESQL_dataServer_database" -U "$POSTGRESQL_dataServer_username"  
/tmp/data_service_backup.dump'  
  
kubectl exec -it <your-postgres-18-pod> -- bash -c  
'PGPASSWORD="$POSTGRESQL_keycloak_password" pg_restore -d  
"$POSTGRESQL_keycloak_database" -U "$POSTGRESQL_keycloak_username"  
/tmp/keycloak_backup.dump'
```

#### 5. Restart the services.

Shell

##### # Restart services

```
kubectl scale deployment data-service --replicas=1  
kubectl scale deployment keycloak --replicas=1
```

#### 6. Clean the files.

Shell

##### # Clean the backup files

```
kubectl exec -it <your-postgres-18-pod> -- bash -c 'rm -f /tmp/*_backup.dump'
```

#### 7. Rollback if needed:

- Stop everything (including the Data Service).
- Redeploy the previous application version (with PostgreSQL v15) and the old volume.
- Start the application.

**Docker:**

1. Stop the service that connects to the database (Data Service and Keycloak).

Shell

```
# On the docker folder ("deployment/docker")
cd deployment/docker

# Stop the services that connect to the database, but leave the database container
running.
docker compose -f ./app/docker-compose.yml stop data-service
docker compose -f ./infra/docker-compose.yml stop keycloak
```

2. Back up the required schema(s).

Shell

```
# Back up the databases
docker compose exec -T postgres bash -c 'PGPASSWORD="data_server" pg_dump -Fc --clean
--if-exists -U data_server --schema=public -d data_server' > data_service_backup.dump

docker compose exec -T postgres bash -c 'PGPASSWORD="keycloak" pg_dump -Fc --clean
--if-exists -U keycloak --schema=public -d keycloak' > keycloak_backup.dump
```

3. Deploy the new application (with PostgreSQL v18).

Shell

```
# Make sure your Docker compose has been migrated correctly beforehand
postgres:
  volumes:
    - pgdata:/var/lib/postgresql
  environment:
    - PGDATA=/var/lib/postgresql/18/docker
```

Only run the PostgreSQL container.

Shell

```
# Still on the docker folder ("deployment/docker")
# Stop postgres 15 and remove its volume.
docker compose -f ./infra/docker-compose.yml down -v postgres

# Start the new Postgres 18
docker compose -f ./infra/docker-compose.yml up -d postgres
```

#### 4. Restore the backup.

Shell

##### # Copy the backup files

```
docker compose cp data_service_backup.dump postgres:/tmp/data_service_backup.dump
docker compose cp keycloak_backup.dump postgres:/tmp/keycloak_backup.dump
```

##### # Restore the backup

```
docker compose exec -T postgres bash -c 'PGPASSWORD="data_server" pg_restore -d
data_server -U data_server /tmp/data_service_backup.dump'
docker compose exec -T postgres bash -c 'PGPASSWORD="keycloak" pg_restore -d keycloak
-U keycloak /tmp/keycloak_backup.dump'
```

#### 5. Restart the services.

Shell

##### # Start the application

```
docker compose -f ./infra/docker-compose.yml up -d
docker compose -f ./dbos/docker-compose.yml up -d
docker compose -f ./app/docker-compose.yml up -d
```

#### 6. Clean the files.

Shell

##### # Clean copied files

```
docker compose exec -T postgres rm -f '/tmp/*_backup.dump'
```

#### 7. Rollback if needed:

- Stop everything (including the Data Service).
- Redeploy the previous application version (with PostgreSQL v15) and the old volume.
- Start the application.

## Updated Angular Dependencies

[DBPF-9186](#), [DBPF-9187](#), [DBPF-9188](#), [DBPF-8205](#) (4.7.2), [DBPF-9016](#)

Angular has been upgraded to 21.0.3 and NodeJS to 22.21.1

None

### Updated dependencies:

|                                         |                           |
|-----------------------------------------|---------------------------|
| @angular/animations:                    | from "20.1.7" to "21.0.3" |
| @angular/cdk:                           | from "20.2.0" to "21.0.2" |
| @angular/common:                        | from "20.1.7" to "21.0.3" |
| @angular/compiler:                      | from "20.1.7" to "21.0.3" |
| @angular/core:                          | from "20.1.7" to "21.0.3" |
| @angular/forms:                         | from "20.1.7" to "21.0.3" |
| @angular/localize:                      | from "20.1.7" to "21.0.3" |
| @angular/platform-browser:              | from "20.1.7" to "21.0.3" |
| @angular/platform-browser-dynamic:      | from "20.1.7" to "21.0.3" |
| @angular/router:                        | from "20.1.7" to "21.0.3" |
| @danielmoncada/angular-datetime-picker: | from "20.0.0" to "20.0.1" |
| angular-oauth2-oidc:                    | from "19.0.0" to "20.0.2" |
| ngx-toastr:                             | from "14.2.1" to "19.1.0" |
| @angular-devkit/build-angular:          | from "20.1.6" to "21.0.2" |
| @angular/cli:                           | from "20.1.6" to "21.0.2" |
| @angular/compiler-cli:                  | from "20.1.7" to "21.0.3" |
| @angular/language-service:              | from "20.1.7" to "21.0.3" |
| typescript:                             | from "5.8.3" to "5.9.3"   |

|               |                     |
|---------------|---------------------|
| luxon:        | from 3.3.0 to 3.7.2 |
| @types/luxon: | from 3.2.0 to 3.7.1 |

|                           |                       |
|---------------------------|-----------------------|
| @fullcalendar/angular     | from 6.1.17 to 6.1.19 |
| @fullcalendar/core        | from 6.1.17 to 6.1.19 |
| @fullcalendar/daygrid     | from 6.1.17 to 6.1.19 |
| @fullcalendar/interaction | from 6.1.17 to 6.1.19 |
| @fullcalendar/list        | from 6.1.17 to 6.1.19 |
| @fullcalendar/timegrid    | from 6.1.17 to 6.1.19 |

|                    |                       |
|--------------------|-----------------------|
| ag-grid-angular    | from to 34.1.0 34.3.0 |
| ag-grid-community  | from to 34.1.0 34.3.0 |
| ag-grid-enterprise | from to 34.1.0 34.3.0 |

### New dependencies:

|                |         |
|----------------|---------|
| @types/lodash: | 4.17.20 |
|----------------|---------|

### Removed dependencies:

zone.js

⚠ With the migration to Angular 21, DB Gene switched to zoneless change detection, which allows for a smaller bundle size, faster load times, more predictable change detection cycles, and easier debugging (since there is no more **zone.js** stacktrace).

Going zoneless slightly changes the timing of the change detection mechanism, and it may uncover issues in your **widgets** that were previously masked by **zone.js**. You may have to fix some **ExpressionChangedAfterItHasBeenCheckedError** as a result.

In your unit tests, you cannot use **waitForAsync**, **fakeAsync**, and the **tick** function anymore, as they relied on **zone.js**.

**waitForAsync** can be safely removed if you ensure that the function it was wrapping returns the Promise **waitForAsync** was used for.

None

```
// Example:
beforeEach(waitForAsync(() => {
  TestBed.configureTestingModule({
    imports: [
      // imports...
    ],
  }).compileComponents()); // <- compileComponents returns a Promise
}));
// would become:
beforeEach(() => {
  TestBed.configureTestingModule({
    imports: [
      // imports...
    ],
  }).compileComponents(); // <- compileComponents returns a Promise
});
```

**fakeAsync** can safely be removed. However, **tick(x)** must be replaced by **jasmine.clock().tick(x)**; and the Jasmine mock clock must be set up before each test:

None

```
beforeEach(() => {
  jasmine.clock().install();
});

afterEach(() => {
  jasmine.clock().uninstall();
});
```

## Updated Java JDK Dependencies

[DBPF-8831](#)

Java 25 JDK is now required for development and runtime.

## Updated Gradle Dependencies

[DBPF-8830](#)

- Gradle updated from 7.6.4 to 9.2.1
- Jacoco updated 0.8.7 to 0.8.14
- SonarQube Gradle plugin updated from 3.3 to 7.0.1.6134

---

## Security

*This section lists all changes related to any security issues.*

### Removed Methods (Internal)

[DBPF-8149](#)

**GeneScopedIdentityService** deprecated methods have been removed across the application.

### Improved Base Images

[DBPF-8828](#), [DBPF-8504](#)

DBOS and DB Gene UBI base images for Java services and workers have been replaced with BellSoft images, which are smaller and have a better vulnerability surface.

In addition, several CVEs related to the base image upgrade of Java services to **ubi8/openjdk-17** have been fixed.

### Improved Configuration of the Content Security Policy

[DBPF-8507](#), [DBPF-8508](#)

For better security, the Content Security Policy configuration now defines default, more restrictive values for the following directives:

**script-src-elem**, **script-src-attr**, **style-src-elem**, **style-src-attr**, **connect-src**, **base-uri**, **frame-ancestors**, **form-action**, **worker-src**, **child-src**, **media-src**, **manifest-src**, **fenced-frame-src**, and **upgrade-insecure-requests**.

---

## Build

*This section lists all changes related to the Scaffolding and Gradle scripts.*

### Updated Requirements for GitLab CI Builds (Internal)

#### [DBPF-8831](#)

When building a project with Gradle 9 and JDK 25 using a Gitlab CI pipeline, create/update the file named **.gitlab-ci.yml** in the root folder of the project with the following content:

```
None
include:
  - project: 'decisionbrain/devops/ci-templates'
    ref: tags/0.5.0
    file: '/templates/gitlab-ci-default.yml'
variables:
  BUILD_IMAGE: build-images.decisionbrain.cloud/gene-project-cloud:2.0.0
```

### Improved Gradle Files

#### [DBPF-8830](#)

Following the update to Gradle 9:

- The code replicate plugin is now applied only in the root project (affects multiple **.gradle** files). You can keep configuring **codeReplicas {...}** in any Gradle project, but note that the **codeReplicasUpdate** task is available only in the root project (it will apply the **codeReplicas {...}** of all projects at the same time).
- The **modelGeneration {...}** extension has been removed; the **modelGenerations {...}** extension should be used instead. Also, a lot of **dependsOn** and source sets configuration have been removed for the tasks that generate code (DTO/JDL/DOM/JPA). Instead, most of the time you can just configure the **targetSourceSets** in the **modelGenerations {...}** extension.
- Some minor changes have been implemented in **.gradle** files to adapt to Gradle 9.

### Improved DSL for the Gradle Plug-In CodeReplica (Internal)

#### [DBPF-9339](#)

Following a Gradle/Groovy upgrade, an error could occur due to ambiguous method signatures with **java.util.function.Supplier** and **org.gradle.api.provider.Provider**. To fix this, the DSL is now simpler to use. The **withValue** method was updated. It now directly accepts **Provider**, **Supplier**, or **Replica** types. This change eliminates the ambiguity and the need for explicit casting. Users must now replace **withValueFrom** with the new **withValue**.

---

## Deployment

*This section lists all changes related to the integration with Docker and Helm.*

### Improved Endpoint Probes for Kubernetes

[DOC-726](#) ([DBPF-7667](#))

In Kubernetes, live and readiness probes have been enabled for the Gateway, Web Client, JupyterLab, and Documentation Chatbot endpoints.

---

## JupyterLab

*This section lists all changes related to the JupyterLab integration, except for **Workers (Python)** and **Routines**.*

### New Jupyter Notebook Sample Using the HiGHS Solver (4.7.1)

[DBPF-7644](#)

A new sample Jupyter Notebook using HiGHS (High-performance parallel linear optimization software) library is now available in the **samples** folder.

This sample is supported for all scaffoldings (internal and external) in both the DB Gene and DB Gene Online environments.

---

---

# Scripted Tasks Changes

---

*This part lists all changes related to:*

- *The task definition and script language, in Section **Definition**.*
- 

## Definition

*This section lists all changes related to the task definition and script language.*

### Removed Methods and Task (4.7.2)

[DBPF-9040](#), [DBPF-9036](#)

Several deprecated methods from **idOfNewScenarioWithFolder** and **idOfNewScenarioWithMetadata** have been removed.

Also, the deprecated **DbrfImportTask** has been removed.

---

# UI Changes

---

*This part lists all changes related to:*

- The custom renderers and controllers, in Section **Extensibility**.
  - The table-based widgets, regardless of their type, such as the Data Explorer, Data Grid, Job List, Scenario List, Workspace List, Issue List, and Issue Details widgets, in Section **Tables**,
  - The default widgets, in Section **Data Grid/Explorer, Chart, Gantt, Filter, or Pivot Table Widgets** — if a change impacts more than one widget, it is listed in the most relevant section and referred to in the others.
- 

## Extensibility

*This section lists all changes related to the custom renderers and controllers.*

### Improved Modals API

#### [DBPF-9263](#)

The **GeneModalComponent** has been reworked to:

- Deprecate the usage of the **modalConfiguration** field in favour of the **GeneModalConfigurationToken** injection token. A warning will be logged in the console if the deprecated field is detected.
- Introduce a new injection token **GeneModalConfigurationToken** to inject the modal configuration instead.
- Update all gene modals to use the injection token.
- Remove the **InsertionDirective** that was used to access the **ViewContainerRef** of an **HTMLElement**. The **ViewContainerRef** is accessed directly instead.

The **modalConfiguration** field is still working. When used, a warning message is displayed in the console to explain that **GeneModalConfigurationToken** should be used instead.

To migrate your classes implementing this interface, you can remove the **modalConfiguration** field and replace it with **myConfig: GeneModalConfiguration<MyModalConfig> = inject(GeneModalConfigurationToken);**

## Examples of migration:

None

```
// Before
class MyModal implements GeneModalComponent {
    modalConfiguration: GeneModalConfiguration;
}

//After
class MyModal implements GeneModalComponent {
    myConfig: GeneModalConfiguration = inject(GeneModalConfigurationToken);
}
```

None

```
// Before
class MyModal implements GeneModalComponent {
    set modalConfiguration(modalConfiguration:
GeneModalConfiguration<MyModalConfiguration>) {
        this.myValue = modalConfiguration.data.myValue;
        this.requestLoadData();
    }
    myValue: any;
}

//After
class MyModal implements GeneModalComponent {
    myConfig: GeneModalConfiguration<MyModalConfiguration> =
inject(GeneModalConfigurationToken);
    myValue = this.myConfig.data.myValue;
    constructor() {
        this.requestLoadData();
    }
}
```

**i** The field name for the injected **GeneModalConfiguration** is arbitrary and can be customised to any preferred identifier. Although the previous use of a configuration field named **modalConfiguration** is deprecated and will log a console warning, using this name for the field holding the injected token will not trigger the deprecation warning, as in:

None

```
class MyModal implements GeneModalComponent {
    modalConfiguration: GeneModalConfiguration = inject(GeneModalConfigurationToken);
}
```

---

## Tables

*This section lists the changes related to all table-based widgets, regardless of their type, such as the Data Explorer, Data Grid, Job List, Scenario List, Workspace List, Issue List, and Issue Details widgets.*

### Removed Method (4.7.2)

#### [DBPF-9038](#)

The deprecated method `tableApi.getPressedKeyCodes()` has been removed. Use `tableApi.getLastUserInteractionEvent` instead when you need to know if modifier keys are pressed.

---

## Data Grid/Explorer Widgets

*This section lists all changes related to the Data Grid and Data Explorer widget.*

### Removed Pagination in the Data Grid/Explorer Widgets (4.7.2)

#### [DBPF-8623](#)

The deprecated pagination option in the Data Grid/Explorer widgets has been removed. They now use the AG Grid *Server-Side Row Model*.

### Improved Documentation in the Data Grid/Explorer Widgets Custom Controller

#### [DOC-1341](#) ([DBPF-9150](#))

The Data Grid/Explorer widgets custom controller has been improved in the `gene-table-controller.model.ts`.

---

## Chart Widgets

*This section lists all changes related to the Chart widget.*

### Removed Option for Pie Charts (4.7.2)

[DBPF-9106](#)

The option *Show legend entries*, which applies to series with a *Split by*, has been removed.

It allowed displaying a single legend entry for the series, or an entry for each split value. Since the legend shows colors for the categories, not the series, this does not make sense for *Pie* charts.

---

## Gantt Chart Widget

*This section lists all changes related to the Gantt Chart widget.*

### Improved Sample Custom Controller for the Gantt Chart Widget (4.7.2)

[DBPF-9144](#)

The Sample Gantt controller *Sample Controller Load Data*, which implements a custom data model building, now allows the configuration of a series to be optional.

### Improved Tooltip Service (Internal)

[DBPF-9450](#)

The offset between the cursor and the tooltip is no longer passed by the **MouseOverInteractor** but managed by the **TooltipService** itself. If the low-level **MouseOverInteractor** has been customized, which is unlikely, this may require attention.

---

## Filter Widget

*This section lists all changes related to the Filter widget.*

### Removed Property in the Filter Widget Custom Controller (4.7.2)

[DBPF-8733](#)

The deprecated **fromSelection** property has been removed from **GeneEntityFilter**.

---

## Pivot Table Widget

This section lists all changes related to the Pivot Table widget.

### Improved Custom Controller for the Pivot Table Widget

#### [DBPF-9261](#)

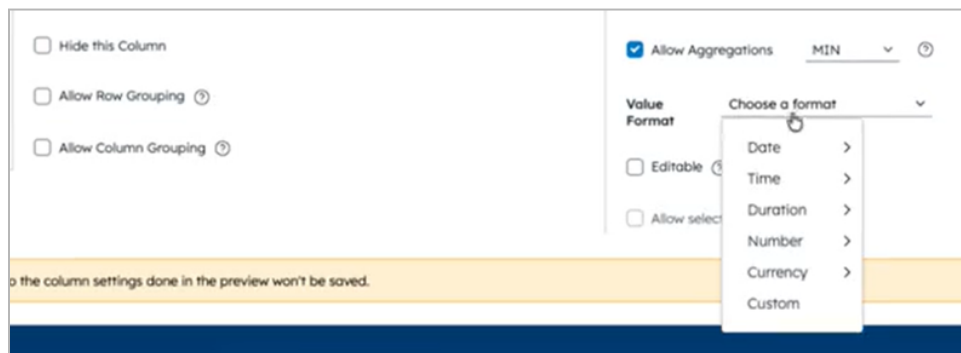
The **PivotTableComponent** custom controller has been reworked to:

- Add a new component **InnerPivotTableComponent**, responsible for handling the AG Grid pivot table.
- Make **PivotTableComponent** only responsible for handling the **GeneWidget** and supervising the subcomponents **InnerPivotTableComponent** and **AggregatedRowsViewComponent**.
- Add **JSDoc** nearly everywhere with breakdown for complex functions.
- Make **PivotTableController** methods optional.
- Add a new class, **PivotTableDatasourceConfigService**, for handling the pivot table datasource configuration.
- Add a new **GeneDynamicHeaderClassWidget** introducing the **headerCssClass\$**, allowing widgets to set the widget header CSS classes dynamically.
- Make **InnerPivotTableComponent**, **PivotTableComponent**, and **AggregatedRowsViewComponent** standalone.

### New Value Formatting for the Pivot Table Widget

#### [DOC-1168](#) ([DBPF-8749](#))

In the Pivot Table widget configurator, the new *Value Format* option appears as a dropdown selector.



# Deprecated Features and APIs Scheduled for Removal in 4.9.0

The following deprecated elements will no longer be supported from version 4.9.0.

*This section lists all removals that are planned to come in 4.9.0.*

- The methods **hasErrorsOrWarnings** and **issues** in **ScenarioDataExpression**.
- The following methods in **ScenarioData**:
  - ScenarioData#getRawData** and its implementation.
  - ScenarioData#setRawData** and its implementation.
  - ScenarioData#getIssues** and its implementation.
  - getModifiedTables** only in the interface.
  - resetModifiedTables** and its implementation.
  - addModifiedTables** only in the interface.
  - isSchemaCheckRequested** only in the interface.
  - setSchemaCheckRequested** only in the interface.
  - hasErrorsOrWarnings** only in the interface.
- Deprecated methods in Collector load/save snapshot.
- Deprecated method **com.decisionbrain.gene.data.service.graphql.jpa.service.ComparisonStatusChecker#checkScenarios(com.decisionbrain.gene.application.model.GeneContext, boolean)**.
- **buildChart** in **web/web-frontend-base/projects/gene/common-widget/src/lib/gene-chart/model/gene-chart-controller.ts**.
- **GeneTaskConfiguration.file** in **web/web-frontend-base/projects/gene/data/src/lib/modules/open-api/execution/model/gene-job-input.model.ts**.
- The **com.decisionbrain.gene.data.dto.PageDT0ForServices** class.