



Migration Guide

DB Gene 4.2.0

March 29th, 2024

Copyright © 2012-2024 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 4.2.0 Migration Guide

- Application Changes..... 4**
 - General Application Changes..... 4**
 - New Composite Data Model Feature.....4
 - Workspaces & Scenarios..... 4**
 - New Scenario Types.....4

- Data Changes..... 5**
 - JDL..... 5**
 - New JDL Syntax for the Composite Data Model..... 5
 - Built-in Import/Export..... 5**
 - Updated Import Mechanism.....5
 - Data Service..... 6**
 - Removed Data Service Classes..... 6
 - Updated SchemaChecker API.....6
 - Execution Service..... 7**
 - New Dependency.....7

- Dev Changes..... 8**
 - REST API..... 8**
 - Deprecated "GET" for "scenariosIdsAndNames"8
 - Removed REST API Elements.....8
 - Updated REST API Elements.....9
 - 3rd-Party Components..... 9**
 - Updated Angular Dependencies.....9
 - Updated Jackson Dependency..... 10
 - Jackson mappings are used for REST payload serialization and deserialization, for example..... 10
 - Updated Java Dependency.....10
 - New Java Dependency.....10

- Scripted Tasks Changes..... 11**
 - Routines..... 11**
 - Updated Genelssue.....11

- UI Changes..... 12**
 - Tables..... 12**
 - Removed Deprecated Elements..... 12
 - Updated Libraries..... 12
 - Workspace and Scenario Widgets..... 13**
 - Updated Scenario List Widget..... 13
 - Updated Gene Table Component..... 13
 - Button Widget..... 14**
 - Deprecated Navigation Button Widget..... 14
 - Updated New Job Button Widget..... 14
 - New Button Handler..... 14

- Deprecated Features and APIs Scheduled for Removal in 4.3.0..... 15**

- *For more details, please refer to the DB Gene Documentation available **here**.*

- *The order of the sections and subsections is based on the DB Gene Release Notes Changelog, available **here**.*

- *Content in each subsection is structured as follows: Removed, Deprecated, Updated, and New.*

- *Changes made to the DB Gene REST API are listed in Section **Dev Changes > REST API**.*

- *Changes made to the DB Gene dependencies are listed in Section **Dev Changes > 3rd-Party Components**.*

- *Deprecations for removal in 4.3.0 are listed in Section **Deprecated Features and APIs Scheduled for Removal in 4.3.0**.*

Application Changes

This part lists all changes related to:

- *general features on application element management, including events and hooks, in Section **General Application Changes**.*
 - *the workspaces and scenarios lifecycle, features, or APIs, in Section **Workspaces & Scenarios**.*
-

General Application Changes

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

New Composite Data Model Feature

The Composite Data Model feature now allows defining scenario types and references between scenarios to share their data with each other.

Workspaces & Scenarios

This section lists all changes related to the lifecycle, the features, or the APIs of workspaces and scenarios.

New Scenario Types

In the context of the Composite Data Model feature, scenarios are now associated with a type, and they can reference or be referenced by other scenarios.

Data Changes

This part lists all changes related to:

- the language definition and parsing, in Section **JDL**.
 - the import or export mechanism provided by the product, in Section **Built-in Import/Export**.
 - the Data Service API, in Section **Data Service**.
 - the Execution Service API, in Section **Execution Service**.
-

JDL

This section lists all changes related to the language definition and parsing.

New JDL Syntax for the Composite Data Model

The `JdlApplication` class and the associated ones have been extended to reflect the Composite Data Model features. In particular, when you build a `JdlApplication` instance, you must provide a list of `JdlScenarioTypes`; if your JDL is still in the style of the 4.1.0 release, that is, without scenario types, then you can pass a singleton list of `JdlScenarioType.default()`. Similarly, instances of `JdlEntity` must be given a scenario type name (the name of the default scenario type is ""). Finally, if you are using the `JdlApplicationLoader` class, its current `readJdl` methods have been replaced with one that takes a `JdlStreamResolver`.

Built-in Import/Export

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

Updated Import Mechanism

With the new Composite Data Model feature, when importing a scenario, users can now select its type and reference other scenarios to share data.

Data Service

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

Removed Data Service Classes

The following classes have been deprecated:

- `ReflexionGetService`. Instead, use `DatabaseGetService`.
- `ReflexionInsertService`. Instead, use `DatabaseInsertService`.
- `ReflexionUpdateService`. Instead, use `DatabaseUpdateService`.

Updated SchemaChecker API

The interface `SchemaChecker` has been updated to support the Composite Data Model (CDM) features. This implies the following things:

- Update the method signatures of your `SchemaChecker`, if you implemented any.
- Adapt your `SchemaChecker` to support the CDM.

The following methods have been updated:

```
Unset
public interface SchemaChecker<T extends SchemaCheck> {
    List<T> createSubjects(ScenarioReferenceGraphAware scenario);
    List<String> findInvalidInternalIds(T checker, ScenarioReferenceGraphAware
scenario);
}
```

The parameter `ScenarioReferenceGraphAware scenario` has been added. This object contains all the CDM; it represents the scenario for which the checker is called and contains all the CDM information of the scenario.

In general, you will need to have JDL representation to adapt the implementation of your checker—a bean of type `JdlApplication` is available in the Spring context. Now, for instance, to get the list of entities contained by the scenario, you can do `scenario.getProperEntities(jdlApplication)`.

Execution Service

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

New Dependency

In the `execution-service` module, the `build.gradle` file now declares a new dependency to the `:gene-model:gene-model-desc` module.

```
Unset
dependencies {
    // Execution service application
    implementation
    "com.decisionbrain.gene:execution-service-application:${versions.decisionbrain.dbgene
}"

    // Extension
    implementation project(":extensions:execution-service-extension")

    // Description of the application data model
    implementation project(":gene-model:gene-model-desc")

    // =====
    // = There should not be any additional dependencies here. =
    // = Please customize the extension instead.               =
    // =====
}
```

Dev Changes

This part lists all changes related to:

- the DB Gene REST API, except for the **Data** and **Execution** Services API, in Section **REST API**.
 - the DB Gene external dependencies and libraries, such as Angular or Keycloak, in Section **3rd-Party Components**.
-

REST API

This section lists all changes related to the DB Gene REST API, except for the Data Integration Core, Data Service, Execution Service, and Scenario Service API.

Deprecated “GET” for “scenariosIdsAndNames”

For the `/api/scenario/scenariosIdsAndNames` endpoint, the GET method is now deprecated. Use the POST method instead.

Removed REST API Elements

- In class `GeneContextService`, methods `setScenarioIds()` and `addScenarioId()` have been deprecated since June 2020 and are now removed. Instead, use `setScenarioSelection()` and `addToScenarioSelection()`, respectively.
- The type and constant `GeneScenarioEventType` have been deprecated since 4.0.1-fp2 and are now removed. Instead, use `ScenarioNotificationType`.
- In class `GeneSettingsService`, methods `registerDefaultSettings()` and `resetSettings()` have been deprecated since April 2020 and have been removed. Instead, use `registerDefaultApplicationSettings()` and `resetApplicationSettings()`, respectively.
- In interface `GeneWidgetHeaderConfiguration`, member `showMenu` has been deprecated since October 2021 and is now removed. Instead, use `GeneMenuItemsProvider`.

- In interface `GeneModalDialogButton`, the member `shortcut` and its associated type `GeneDialogButtonShortcut` have been deprecated since February 2021 and are now removed.
- In class `ExecuteOptimizationServerTaskStatement`, the variant of method `withOutputScenario()` that takes a format as argument has been deprecated since 4.0.0-fp4 and is now removed.
Instead, use the other variant of this method as only the CSV format is supported.
- In type `JobInputType`, constant `NUMERIC` and method `numeric()` have been deprecated since 4.0.1-fp3 and are now removed.
Instead, use `REAL` and `real()`, respectively.

Updated REST API Elements

The type `ScenarioDTO` has been renamed into `ScenarioCreationRequestDTO`.

3rd-Party Components

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

Updated Angular Dependencies

```
Unset
// ----- Updated Angular Dependencies
@fullcalendar/angular: 6.1.8 -> 6.1.10
@fullcalendar/core: 6.1.8 -> 6.1.10
@fullcalendar/daygrid: 6.1.8 -> 6.1.10
@fullcalendar/interaction: 6.1.8 -> 6.1.10
@fullcalendar/list: 6.1.8 -> 6.1.10
@fullcalendar/timegrid: 6.1.8 -> 6.1.10

@angular/common: 17.0.7 -> 17.2.3
@angular/core: 17.0.7 -> 17.2.3
@angular/forms: 17.0.7 -> 17.2.3
@angular/router: 17.0.7 -> 17.2.3
@angular/animations: 17.0.7 -> 17.2.3
@angular/cdk: 17.0.4 -> 17.2.1
@angular/compiler: 17.0.7 -> 17.2.3
@angular/core: 17.0.7 -> 17.2.3
```

```
@angular/forms: 17.0.7 -> 17.2.3
@angular/google-maps: 17.0.4 ->17.2.1
@angular/localize: 17.0.7 -> 17.2.3
@angular/platform-browser: 17.0.7 -> 17.2.3
@angular/platform-browser-dynamic: 17.0.7 -> 17.2.3
@angular-devkit/build-angular: 17.0.7 -> 17.2.1
@angular/cli: 17.0.7 -> 17.2.1
@angular/compiler-cli: 17.0.7 -> 17.2.3
@angular/language-service: 17.0.7 -> 17.2.3

@ag-grid-angular : -> 31.0.0
@ag-grid-community : -> 31.0.0
@ag-grid-enterprise : -> 31.0.0
@echarts : -> 5.4.3
@ngx-echarts : -> 17.1.0
@ngx-quill : -> 24.0.4
```

Updated Java Dependency

Keycloak has been upgraded to version 23.0.4.

Unset

```
// ----- Updated Java Dependency
com.decisionbrain.db-keycloak 21.1.0 → 23.0.4
```

New Java Dependency

JGraphT is introduced with version 1.5.2.

Unset

```
// ----- New Java Dependency
@JGraphT 1.5.2
```

Updated Jackson Dependencies

The libraries `object-mapper-base` that provide a common configuration for Jackson mappings do not include the Kotlin configuration by default. However, it still can be added at the project customization level.

Jackson mappings are used for REST payload serialization and deserialization, for example. The update of Keycloak brings an incompatible version of Jackson.

A constraint has to be added to ensure a consistent version of Jackson libraries across the application.

Edit the file `gradle/templates/spring-bom.gradle` and add the following content in the `configurations.configureEach` block.

```
Unset
configurations {
    configureEach {
        // ... Existing configuration ...
        resolutionStrategy.eachDependency { DependencyResolveDetails details ->
            if (details.requested.group.startsWith('com.fasterxml.jackson')) {
                details.useVersion '2.14.3'
                details.because '25/03/2024: Force Jackson to the version provided by
Spring Boot BOM'
            }
        }
    }
}
```

Scripted Tasks Changes

This part lists all changes related to:

- routines in Java & Python, in Section **Routines**.
-

Routines

This section lists all changes related to routines in Python or Java.

Updated GeneIssue

When creating a `GeneIssue` in a Composite Data Model for an `InstanceClass`, the `GeneIssue` must be saved on the scenario owning the `InstanceClass` (ie: the `InstanceClass` must be a proper entity of the scenario).

If the `GeneIssue` is not associated with the right scenario, the UI might not display it correctly.

UI Changes

This part lists all 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, in Section **Tables**.
 - specific UI widgets, in Sections **Workspace and Scenario** and **Button** widgets — if a change impacts more than one widget, it is listed in the most relevant section and referred to in the others.
-

General UI Changes

This section lists all changes related to all widgets, regardless of their type, or non-widget elements of the UI, such as the sidebar, menus, scenario selector, or views and dashboards, except for the Access Control elements, such as permissions and API keys.

Deprecated Navigation Button Widget

The Navigation Button widget is deprecated and can no longer be added to a dashboard or view, as its role can be fulfilled using the new Button widget.

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 Deprecated Elements

Class `GeneScenarioListParams` has been removed.
Instead, use `GeneScenarioListConfiguration` if needed.

Property `comparisonState` has been removed from the class `GeneTableWidgetState`.

Updated Libraries

The `gene-ag-grid-utils.ts` that exports methods `setGeneContext`, `getGeneContext`, `getGeneTranslateService`, and `setGeneTranslateService` has been moved to library `@gene/widget-core`. If you use these methods in your Web UI code, you should update the imports.

Workspace and Scenario Widgets

This section lists all changes related to the Workspace List, Scenario List, and Scenario Timeline widget.

Updated Scenario List Widget

Custom Actions now use the new Action API.

- `GeneScenarioListConfiguration#customTasks` is now deprecated in favor of `GeneScenarioListConfiguration#GeneScenarioListConfiguration`.
- `GeneScenarioListCustomActionTask` is now deprecated in favor of `GeneScenarioListCustomAction`.

Updated Gene Table Component

Some breaking changes related to the AG Grid v31 upgrade should be noted:

- `gridOptions` are now set to read-only during table initialization. Setting a property on the `gridOptions` object may not be detected by the grid. Instead, `api.setGridOption(property, value)` should be used in any custom code that alters the grid configuration.
 - Function `getRowId` can only be set once. Updating after the `ag-grid` component has been initialized is no longer needed.
 - For the reasons described above, function `GeneTableController.customizeGridOption` is now called when the widget configuration and/or custom controller is assigned to the widget. This should not require changes to the custom controller code. It was previously called upon the initialization of the `ag-grid` component.
 - `columnApi` has been deprecated in AG Grid. Functions of `columnApi` have been moved to `gridApi` and the calls in the custom code should be changed accordingly
 - If changes to the `gridOptions` are made only through `GeneTableController.customizeGridOption`, there should be no change required on the custom code unless it requires the grid to be initialized. If the custom code somehow mutates the `gridOptions` after it is set to the `ag-grid` component, the corresponding code should be replaced by a call to `api.setGridOption`
-

Button Widget

This section lists all changes related to the Button widget.

Updated New Job Button Widget

The New Job Button widget is renamed to `Button` and can now use the Action API. Existing New Job Button widgets will still work as expected.

New Button Handler

`GeneNewJobButtonController` has a new optional method `actionHandler` to handle clicks on the button.

Unset

```
/**
 * Implementing this method allows to override the click behavior of the button.
 * @param event The native pointer event
 * @param context The GeneContext when the click occurred
 * @param action The action configuration of the button
 */
actionHandler?: (event: PointerEvent, context: GeneContext, action: GeneAction<any>)
=> void;
```

Deprecated Features and APIs Scheduled for Removal in 4.3.0

This section lists all removals planned to come in DB Gene 4.3.0.

- No change
-