Release Notes
DB Gene 4.2.0

March 29th, 2024

Important Notes

*Please take into account the following important information when using the new version of **DB Gene 4.2.0**, released on March 29th, 2024.*

Note:

DB Gene 4.2.0 introduces several infrastructure updates and deprecations.
They are described in the DB Gene 4.2.0 Migration Guide, available on the DecisionBrain website.

The following information only focuses on the main changes in this release.

Updates

DB Gene 4.2.0 introduces the following infrastructure updates.

AG Grid Update

DB Gene 4.2.0 now uses AG Grid 31.0.0. It was formerly version 30.2.0.

PostgreSQL Update

DB Gene 4.2.0 now uses PostgreSQL 15.5. It was formerly version 15.2.

Java JDK Update

DB Gene 4.2.0 now uses Java JDK 17.0.9. It was formerly version 17.0.4.

Angular Update

DB Gene 4.2.0 now uses Angular 17.2.3. It was formerly version 17.0.7.

Keycloak Update

DB Gene 4.2.0 now uses Keycloak 23.0.4. It was formerly version 21.1.1.

Deprecations

- 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.

- Type `ScenarioDTO` has been renamed into `ScenarioCreationRequestDTO`.

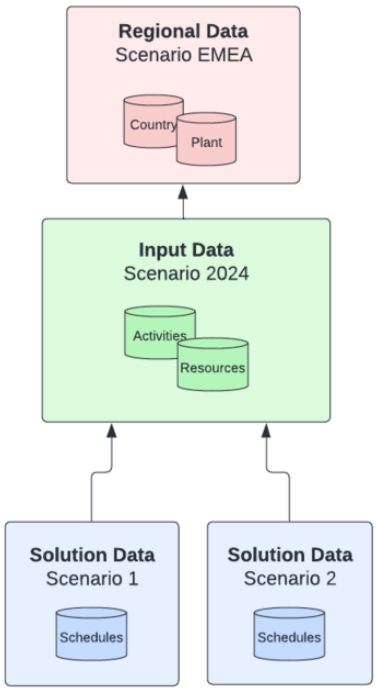
- The Navigation Button widget is deprecated and can no longer be added to a dashboard or view as its role can be fulfilled using the new Button widget.

End-User Features

DB Gene 4.2.0 introduces several end-user improvements with the new Composite Data Model feature, which affects the JDL definition as well as scenario use, import, locks, actions, and information display. Version 4.2.0 also replaces the Navigation and New Job Button widgets with a new one called Button.

New Composite Data Model

DB Gene 4.2.0 introduces the Composite Data Model (CDM) feature, which offers a new level of abstraction to the business data model definition. This allows sharing data between scenarios, which avoids duplicates, greatly improves application overall performance, and reduces resource usage.



The CDM helps solution designers define different scenario types in an application data model. Each scenario type defines a part of the model (tables) and can reference other scenario types.
This division of the application data model into several scenario types is reflected in the structure of scenarios: each scenario of the application has a scenario type, contains only the data defined in the corresponding tables of its scenario type, and references scenarios as expressed in the data model.

For example, an application may have three scenario types defined:

- *Regional Data*, which contains global shared data;
- *Input Data*, which contains optimization inputs on the one hand and references *Regional Data* on the other; and
- *Solution Data*, which contains optimization results and references *Input Data*.

This way, any dashboard displaying data for a *Solution Data* scenario can also display data from the referenced *Input Data* and *Regional Data* scenarios.

Updated JDL Syntax for the CDM

The Platform JDL syntax still accepts JDL files from version 4.1 and creates the same data model and database structure. However, it has been extended to support the description of a CDM.

A CDM can be spread across several JDL files, which are still looked for in the `gene-model/spec` directory of the project, but which can now also be located in subfolders of the `spec` directory.

Contrary to the classic model, in a composite data model, JDL files must contain:

1. The `application` block, as in version 4.1, which defines metadata about the model, such as the name of the collector class, and potential `include` statements linking it to other JDL files.

   Here is an example of a CDM `application` block :

```
Unset
application {
  // DOM  [java.collectorClass] : [CapacityPlanning]
  include "primary_data.jdl"
  include "delivery_data.jdl"
  include "transactional_data.jdl"
  include "plan_data.jdl"
}
```

   The paths of the files are relative to the directory where the main JDL file is stored;

2. Exactly one `scenarioType` block that indicates the name of the scenario type. There is no constraint between the name of the JDL file and the name of the scenario type. The latter must be a valid identifier. The convention is to use Pascal case for scenario type names (that is, attached words with initial letters in uppercase, including the very first letter, e.g. PrimaryData). The `scenarioType` block supports the `@Description` annotation:

```
Unset
@Description("Primary Data of the application")
scenarioType PrimaryData {
}
```

3. Some `entity` blocks that define the entities of the data model, which associates them with the scenario type described. Entity names must be unique across the application data model and use Pascal case; and

4. Some `relationship` blocks that define relations between two entities of the same scenario type. Relations from an entity in scenario type *ST1* to an entity in scenario type *ST2* are declared by including them in the JDL file that defines *ST1*. In addition, visibility on scenario type *ST2* must be declared by adding an `import` statement in the `scenarioType` block for *ST1*.

   The following block declares relations from entities of *TransactionalData* to entities of *PrimaryData*.

```
Unset
@Description("Data that yields a new plan")
scenarioType TransactionalData {
   import PrimaryData
}
```

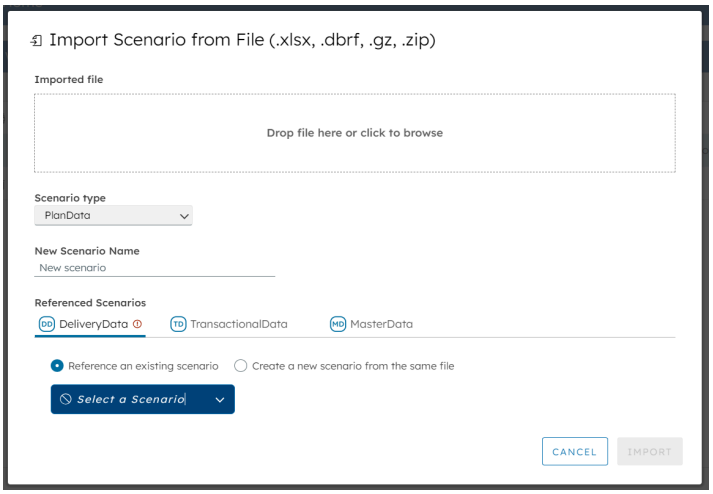
Note that no cycles are allowed in the graph of scenario type imports.

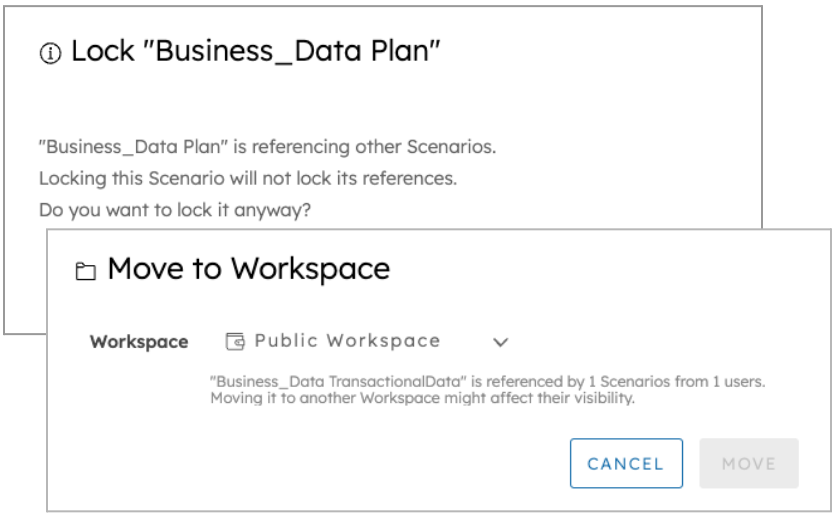
Also, an `import` statement can only refer to a scenario type that is defined in one of the JDL files mentioned in the `include` statements of the application block.

Updated Scenario Import

When using a Composite Data Model, as described above, users must now specify the type of scenario during its import and, if need be, reference other scenarios.

If a scenario to reference is unavailable, the option "Create a new scenario from the same file" can be used to create one from the imported file.





Improved Scenario Lock Mechanism

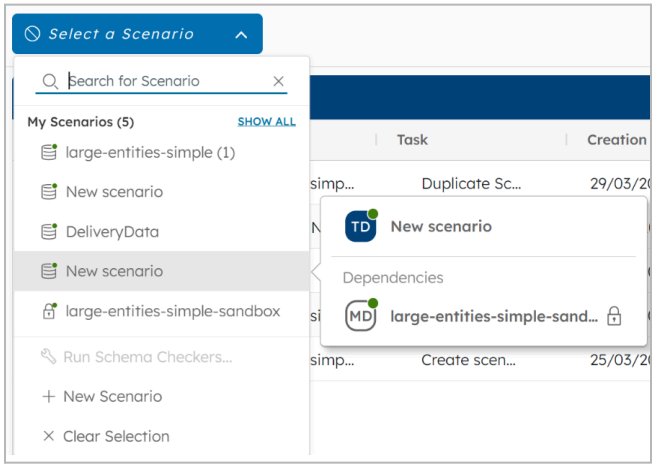
To avoid breaking data continuity, moving or locking a scenario with references from or to other scenarios now triggers a warning.

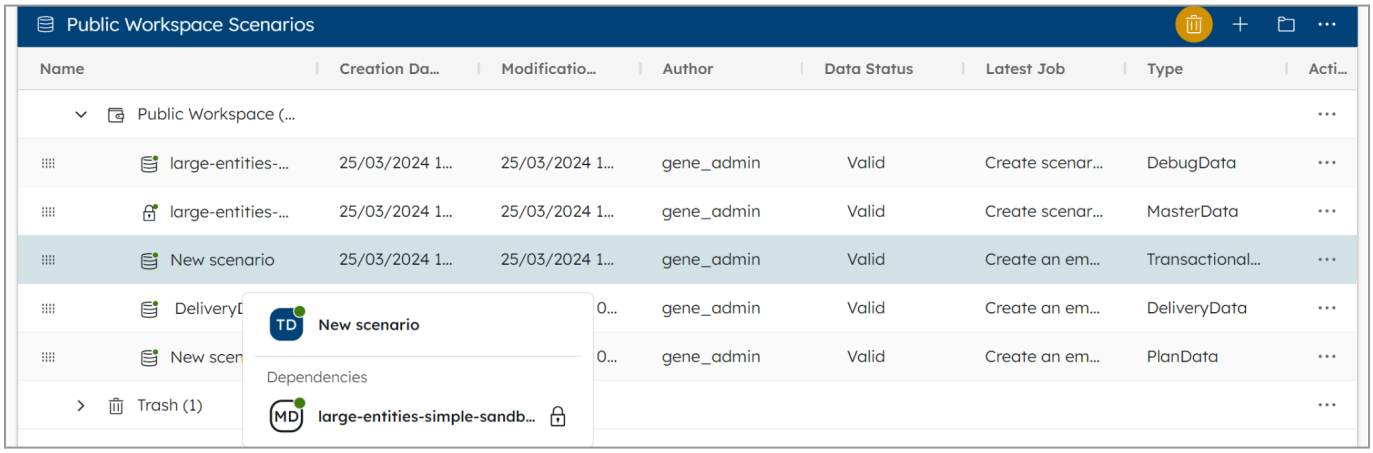
Note that, duplicating a scenario only duplicates the data it contains and its references. It does not copy the data in the referenced scenarios.

Improved Scenario Information Display

Users can now display the type and references of a scenario when hovering over the scenario, in the Scenario Selector.

In the Scenario List widget, users can also display a column for the scenario type. It can also be found, along with its references, using the option "Scenario details" in the Actions column, or when hovering over the scenario.





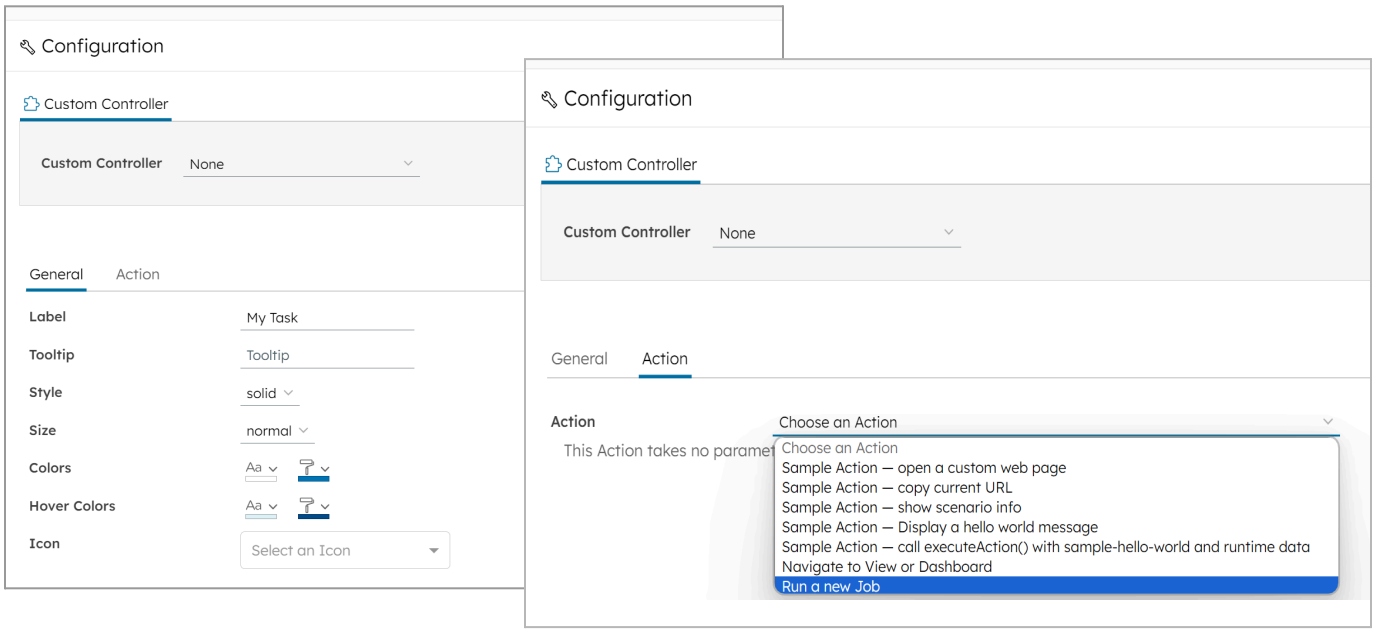
The tooltip displays the same information when hovering over a scenario in the Job List widget.

New Custom Actions in the Scenario List

Custom actions defined via the Action API, introduced in version 4.1.0, are now available in the Scenario List widget from the menu Actions.

New Button Widget

The Navigation Button and New Job Button widgets are now deprecated and make way for a new widget called Button. This new Button widget is easily customizable and relies on the Action API introduced in version 4.1.0.



Note that, even if they are not available as new widgets to create, existing Navigation Button and New Job Button widgets still work as expected.

Changelog

DB Gene 4.2.0 introduces several improvements and bugfixes listed in detail below.

Improvements

DB Gene 4.2.0 introduces the following improvements:

Application *General*	*DOC-177*	Users can now define a composite data model
	DBPF-6061	The JDL metamodel now supports the Composite Data Model feature
	DBPF-6062	The JDL syntax now supports the Composite Data Model feature
	DBPF-6063	Data integration now supports the Composite Data Model feature
	DBPF-6064	In a composite data model, a read-only lock can now be set on scenarios
	DBPF-6065	Permissions are now compatible with the Composite Data Model feature
	DBPF-6066	The UI now supports the Composite Data Model feature
	DBPF-6107	In a composite data model, generated DOMs, whether Python or Java, now include all entities of all JDL files
	DBPF-6112	Scenario data in Spring now supports the Composite Data Model feature
	DBPF-6113	Scenario metadata in MongoDB now supports the Composite Data Model feature

Dev *3rd-party Components*	*DBPF-5660*	The Platform now relies on AG Grid 31.0
	DBPF-5669	The Platform now relies on PostgreSQL 15.5
	DBPF-5670	The Platform now relies on Java JDK 17.0.9
	DBPF-5939	The Platform now relies on Angular 17.2
	DBPF-5941	The Platform now relies on Keycloak 23.0.3
UI *Scenario /Workspace List*	*DBPF-6046*	Users can now display the type and references of a scenario in the Scenario List widget columns
	DBPF-6047	In the Scenario List, the tooltip and Action menu option “Scenario details” now display the type and references of a scenario
	DBPF-5929	The Action API is now available through Custom Actions from the Scenario List widget
UI *Button*	*DBPF-5930*	The Button widget now replaces the New Job and Navigation Button widgets
UI *Job*	*DBPF-6051*	In the Scenario Selector, the tooltip now displays the type and references of a scenario
UI *Job*	*DBPF-6059*	In the Job List, the tooltip now displays the type and references of a scenario

Bugfixes

DB Gene 4.2.0 introduces the following bugfixes:

Application *Views & Dashboards*	*DBPF-6002*	Reverting the changes on a dashboard was triggering an error
Data *Data Integration Framework*	*DOC-770*	The CRF mapping was failing to load from DBM on Docker images
	DOC-769	The CRF datasource was failing to read lines ending with a blank value
DBOS *Master*	*DOC-534*	When too many events were stored, MongoDB was using 5GB of RAM and DBOS was lagging
DBOS *Worker*	*DOC-747*	DBOS jobs were remaining "SCHEDULED" if the related worker was packaged in a Docker image without Java
Dev *Deployment*	*DOC-808*	The file “postgresql.conf” was not taken into account in a “postgres” Kubernetes deployment
Dev *Security*	*DOC-731*	The Platform was blocking CORS preflight requests in microservices
UI *Extensibility*	*DOC-800*	Entering and exiting edition mode on a Table widget using a Custom controller was not calling the Custom controller “processColumns” callback
UI *Data Grid /Explorer*	*DOC-489*	When pasting values over a greater range than the one copied, the Platform was not properly filling the empty cells
	DOC-627	In some cases, copying and pasting values from a percentage field was not working properly
	DBPF-6137	When clicking on a row in a Data Grid, charts were not being filtered without refreshing the page

UI *Charts*	*DOC-819*	For some series configurations, the Chart widget was not representing “zero” values
UI *Tables*	*DBPF-5864*	The Scenario List widget was crashing when moving the current scenario to the trash bin
UI *Composite Widget*	*DOC-643*	Tab titles were overflowing the widget size
UI *Filter*	*DOC-626*	The Filter widget was not working properly for entities having only relations
	DBPF-6144	In a widget common configuration, the filter option “Applies to” was not working properly
	DBPF-6364	The Filter widget was not working properly when selecting a scenario before the import was complete
UI *Scenario Comparison*	*DBPF-6318*	The Platform was not displaying the message "There is too much data to display" when comparison was failing