DOC 4.4.0 Release Notes

December 11th, 2024

DOC 4.4.0 Release Notes

Note:

DB Gene 4.4.0 introduces several infrastructure updates and deprecations.
They are described in DOC 4.4.0 Migration Guide, available on the DecisionBrain website.

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

Important Notes

*The following information pertains to **DOC 4.4.0**, released on December 11th, 2024.*
This version evolutions are detailed in DOC 4.4.0 Migration Guide, available on the DecisionBrain website.

Updates

DOC 4.4.0 introduces the following infrastructure updates.

Spring Update
DOC 4.4.0 now uses SpringBoot 3.0.5 and Spring Cloud 2023.0.3, Formerly, it was version 3.0.12 and version 2022.0.3, respectively.

PostgreSQL Update
DOC 4.4.0 now uses PostgreSQL 15.5. It was formerly version 15.2.

Java JDK Update
DOC 4.4.0 now uses Java JDK 17.0.9. It was formerly version 17.0.4.

Gradle Update
DOC 4.4.0 now uses Gradle 7.6.4. Formerly, it was version 7.4.2.

Keycloak Update
DOC 4.4.0 now uses Keycloak 26.0.5. Formerly, it was version 21.1.1.

Angular Update
DOC 4.4.0 now uses Angular 18.2.10. Formerly, it was version 17.0.7.

AG Grid Update
DOC 4.4.0 now uses AG Grid 32.3.1. Formerly, it was version 30.2.0.

Apache Update
DOC 4.4.0 now uses Apache Commons Compress 1.26.1, Apache Commons Text 1.12.0, and Apache POI 5.2.5. Formerly, it was version 1.21, version 1.10, and version 5.2.3.

RabbitMQ Update
DOC 4.4.0 now uses RabbitMQ 4.0.2. Formerly, it was version 3.11.2.

NGINX Update
DOC 4.4.0 now uses NGINX 1.26. Formerly, it was version 1.22.

CPLEX Update
DOC 4.4.0 now uses CPLEX 22.1.2. Formerly, it was version 22.1.1.

Python Update
DOC 4.4.0 now requires Python 3.12.x. Formerly, it was version 3.11.x.

Deprecations

Version 4.4.0 brings the following deprecations:

Web Client

- **Pivot Table Widget**: A newer version has replaced the Pivot Table widget.
- **Navigation Button Widget**: Deprecated and can no longer be added to a dashboard or view. Its role is now fulfilled by the new Button widget.
- **Filter Widget Controller**:
  - Method `GeneEntityFilter.fromSelection` is deprecated in favor of `GeneEntityFilter.fromConfig`. Use `getFromConfigComparator` from `@gene/components` to compare configurations.
  - For the `buildFilters` function in `@gene/components`, the parameter `contextSelectionKey: string` has been replaced by `filterIdentifier: GeneFilterIdentifier`, which consists of a `typeName` and the paths leading to it.
- **Gantt Chart Widget Controller**:
  - Methods `getResourceQueryName()` and `getEventsQueryName()` in `DbGanttBuildermethods` are deprecated and will be removed in a future version.
  - Methods `loadResources` and `loadEvents` in `GanttController` are deprecated and will be removed in a future version.
- **Web Client Library**:
  - In the class `GeneApplicationService`, the method `setCustomApplicationController(..)` is deprecated and replaced by `registerApplicationController(..)`.
  - In the class `GeneContextService`, methods `setScenarioIds()` and `addScenarioId()` (deprecated since June 2020) are now removed. Use `setScenarioSelection()` and `addToScenarioSelection()` instead.
  - In the class `GeneScenarioEvent`, The type and constant `GeneScenarioEventType` (deprecated since 4.0.1-fp2) are now removed. Use `ScenarioNotificationType` instead.
  - In the class `GeneSettingsService`, methods `registerDefaultSettings()` and `resetSettings()` (deprecated since April 2020) are now removed. Use `registerDefaultApplicationSettings()` and `resetApplicationSettings()` instead.
  - In the interface `GeneWidgetHeaderConfiguration`, the member `showMenu` (deprecated since October 2021) is now removed. Use `GeneMenuItemsProvider` instead.
  - In the interface `GeneModalDialogButton`, the member `shortcut` and its associated type `GeneDialogButtonShortcut` (deprecated since February 2021) are now removed.

Data Service

- **Java Library (data-service-base):**
  - In `BatchCollectorService`, the method:
    `saveItems(String scenarioId, List<T> entityDTOs)`
    is deprecated in favor of:
    `saveEntities(String scenarioId, List<T> entityDTOs)`
  - In `ScenarioUpdateService`, the method:
    `saveItems(String, List<DataServerEntityDTO> items, SchemaCheckersRunOptions)`
    is deprecated in favor of:
    `saveEntities(String, List<DataServerEntityDTO> entities, S`
- **REST API**:
  - The endpoint `/data/simple-excel-export` is replaced by `/data/excel-export`.
  - The endpoint `/data/simple-excel-import` is replaced by `/data/excel-import`.
  - The parameter `sortColumns` has been removed from the Excel Export API.

Scenario Service

- **Java Library**:
  - The type `ScenarioDTO` has been renamed to `ScenarioCreationRequestDTO`.
- **GraphQL API**:
  - The property `scenarioReferenceGraph` has been removed from the definition of object `Path`. The REST API continues to expose this property.
  -
- **Code Replication Plugin**:
  - The old syntax of `code-replicate-plugin DSL` is deprecated in favor of the new one. Migrate the content of `codeUpdates { }` blocks to `codeReplicas { }`.

Execution Service

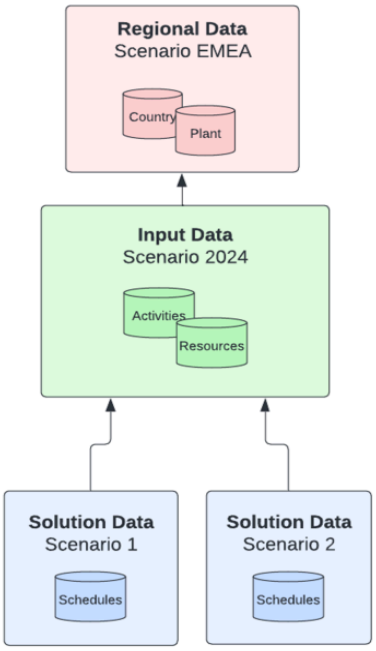
- **Java Library**:
  - The CSV and DBRF file formats are deprecated in the method `ScenarioDataExpression`.
  - In class `ExecuteOptimizationServerTaskStatement`, the variant of method `withOutputScenario()` that takes a format as an argument has been deprecated since 4.0.0-fp4 and is now removed. Use 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. Use `REAL` and `real()`, respectively.

End-User Features

DOC 4.4.0 introduces several end-user improvements with:

- *the new Composite Data Model feature, which extends JDL definition capabilities as well as scenario use, import, locks, actions, and information display;*
- *the new Code Editor widget, which can be used to edit free text or as a ChatGPT terminal;*
- *the new Rule Script Editor widget, which allows executing operations on scenario data, where these operations are coded as Drools rules;*
- *JupyterLab Integration, allowing the import of data from Jupyter notebooks;*
- *Extended web client APIs to add actions, toolbar buttons, and a status bar to widgets;*
- *improved Data Grid/Explorer and Gantt Chart widgets, a refactored Pivot Table widget, and a new Button widget that replaces the Navigation and New Job Button widgets;*
- *intuitive filtering functionalities, enhanced tasks, permissions, application controllers, and more.*

New Composite Data Model



DOC 4.4.0 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.

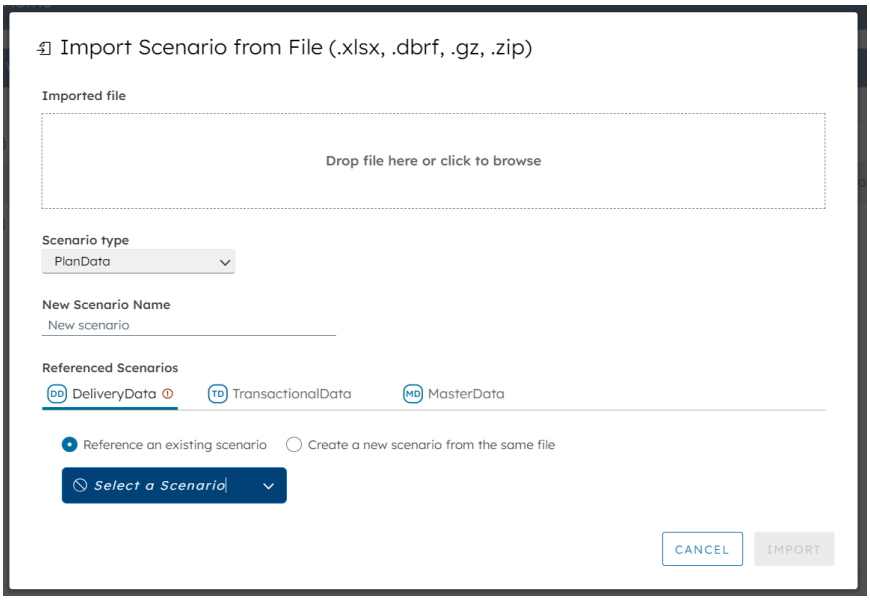
Improved JDL Syntax for the CDM

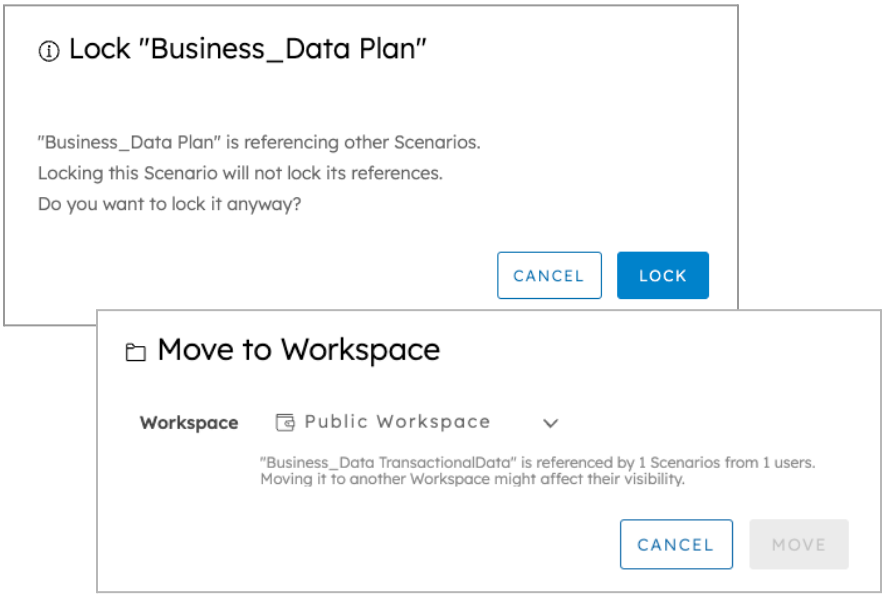
DOC JDL syntax still accepts files from version 4.1 and creates the same model and database structure. However, it has been extended to support the description of a CDM which can be spread across several JDL files, which are still looked for in the `gene-model/spec` directory of the project, but which can now also be located in subfolders of the `spec` directory.

Improved Import for the CDM

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 Lock Mechanism for the CDM

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.

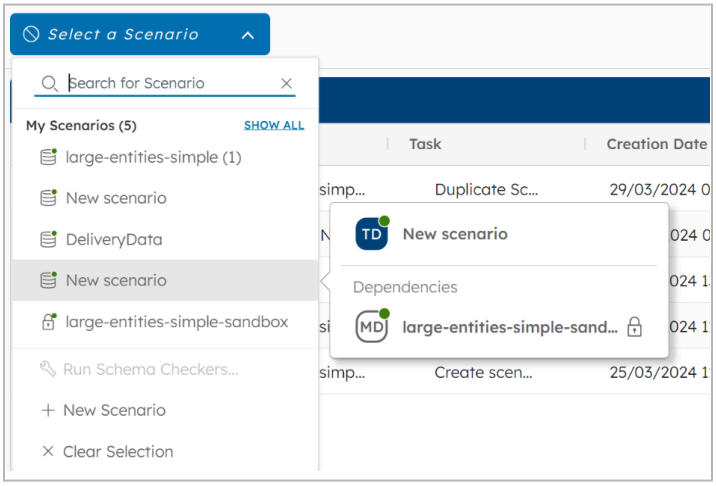
Improved Startup for the CDM

Existing permission rules on referenced scenarios are now checked and migrated at startup.

Improved Display for the CDM

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

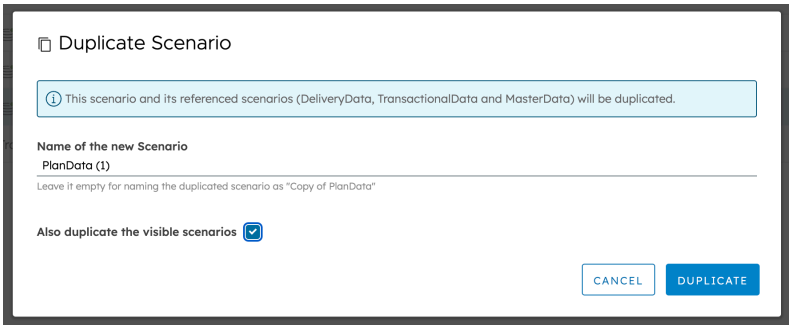
In the Scenario List widget, users can also display a column for the scenario type. It can also be found, along with its references, using the option "Scenario details" in the Actions column, or when hovering over the scenario.



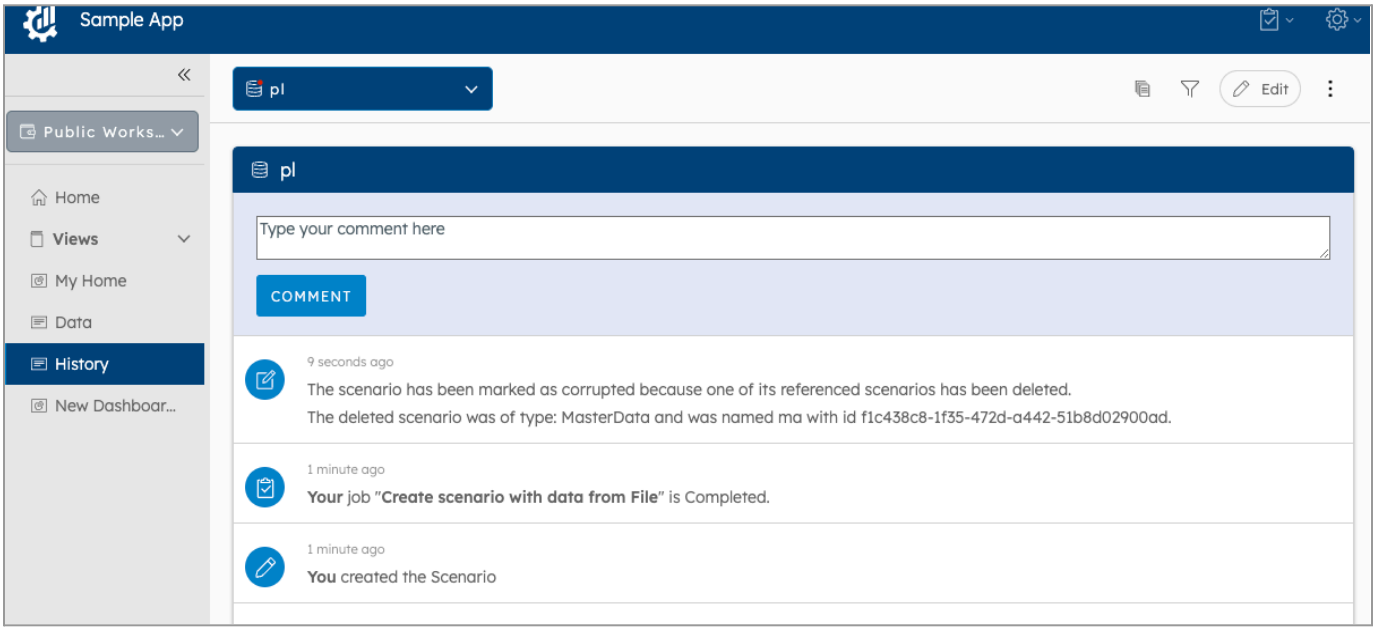
The tooltip displays the same information when hovering over a scenario in the Job List widget.

Improved Scenario Management for the CDM

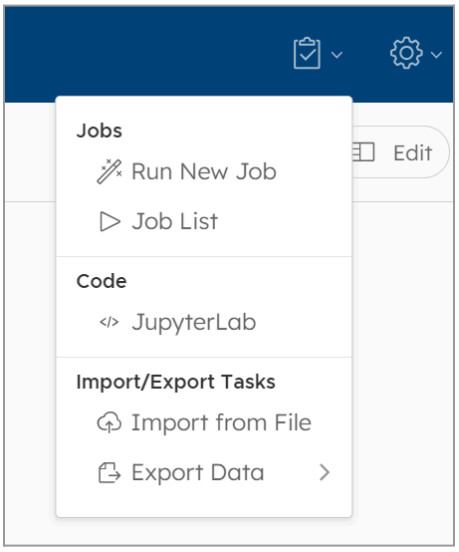
Users can select a scenario link as a reference when adding a CDM scenario. Also, they can also duplicate a scenario and its referenced scenarios.



As it is possible to delete multiple scenarios, regardless of their hierarchy within the composite data model. All the scenarios referencing a deleted scenario are now flagged as "corrupted".



New JupyterLab Integration



The new integration of JupyterLab into DOC as a beta feature eases online development with Python, Java, and CPLEX.

Users are provided with an option in the Topbar Tasks menu and the following sample notebooks in a base Docker image:

- CPLEX for Python
- CPLEX for Java
- CPO for Python
- CPO for Java

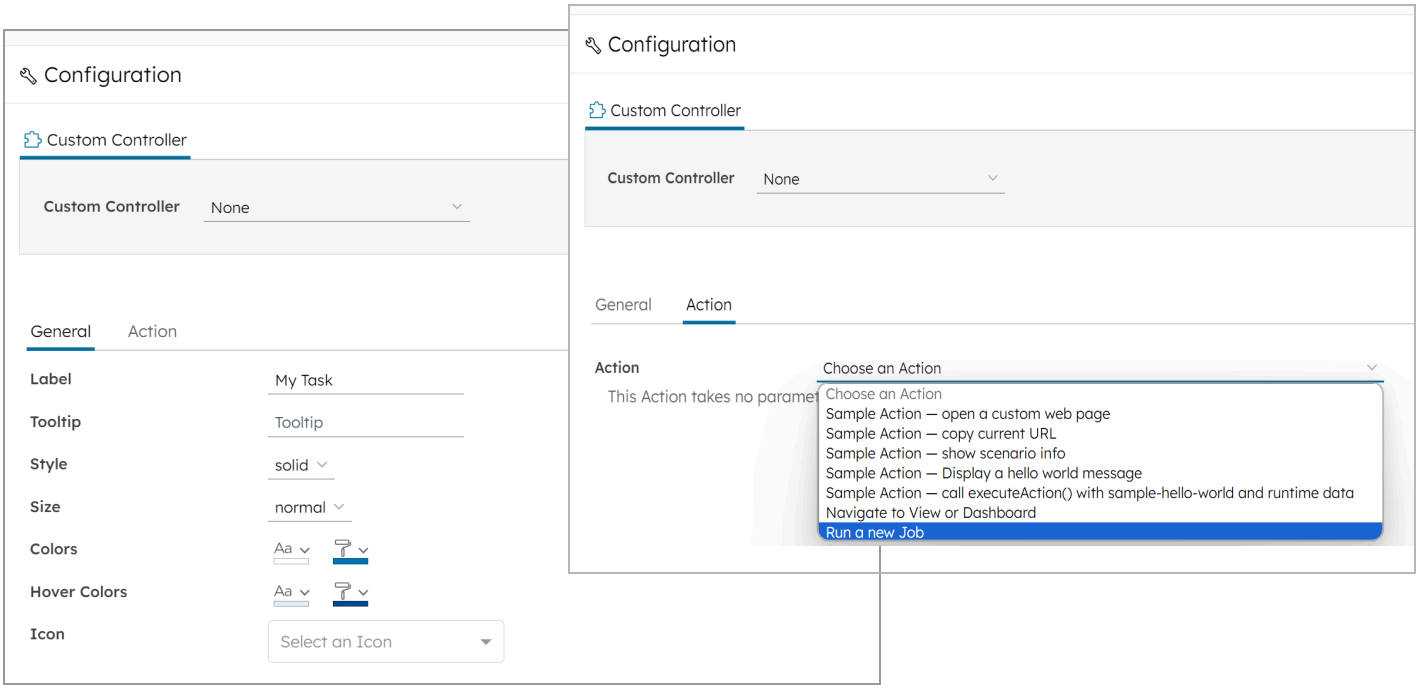
The application now also provides users with a Jupyter Notebook sample to import scenario data into a Pandas data frame.

The Helm chart has been improved to allow deploying the JupyterLab service.

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.

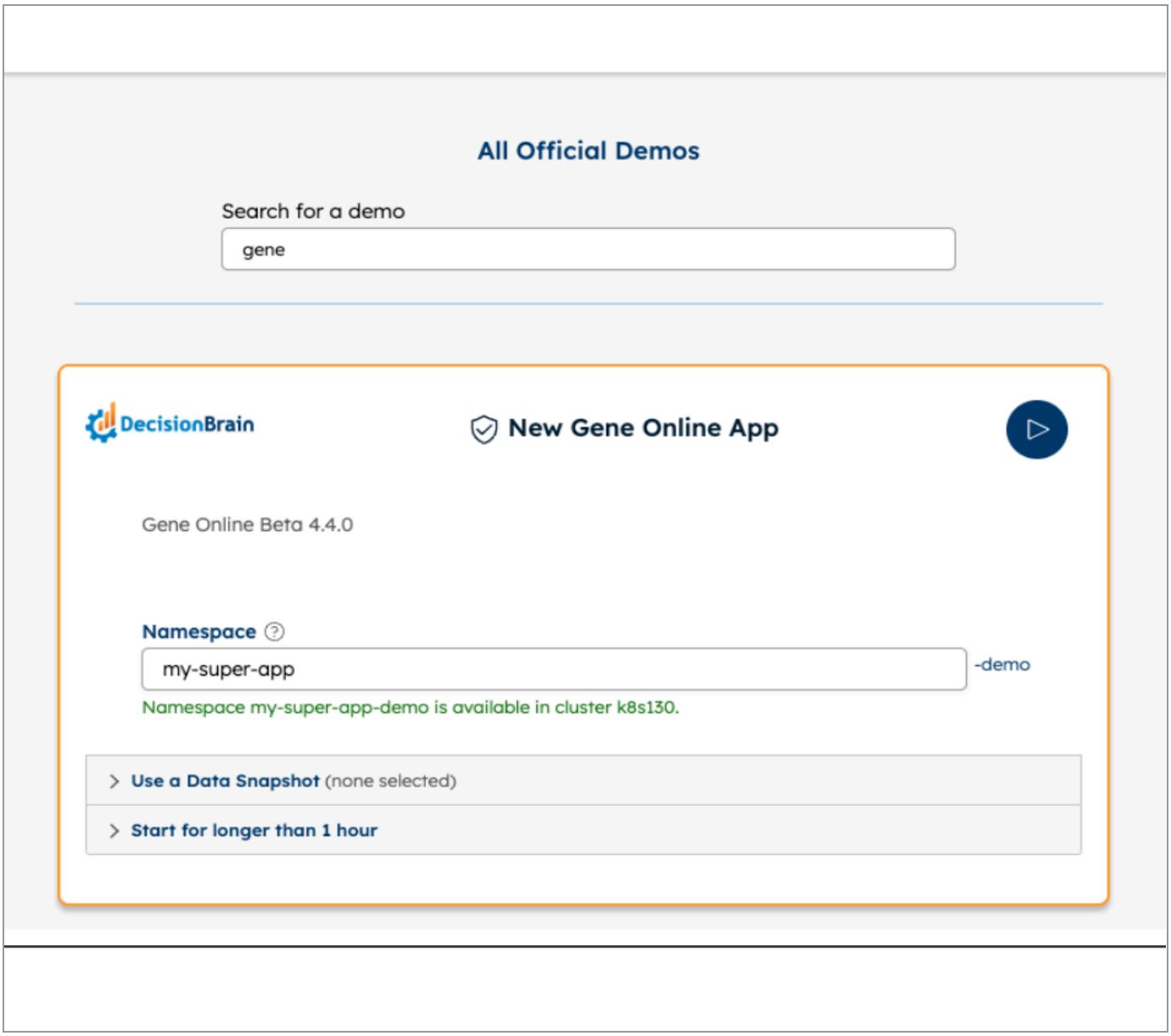
Note that, even if they are not available as new widgets to create, existing Navigation Button and New Job Button widgets still work as expected.



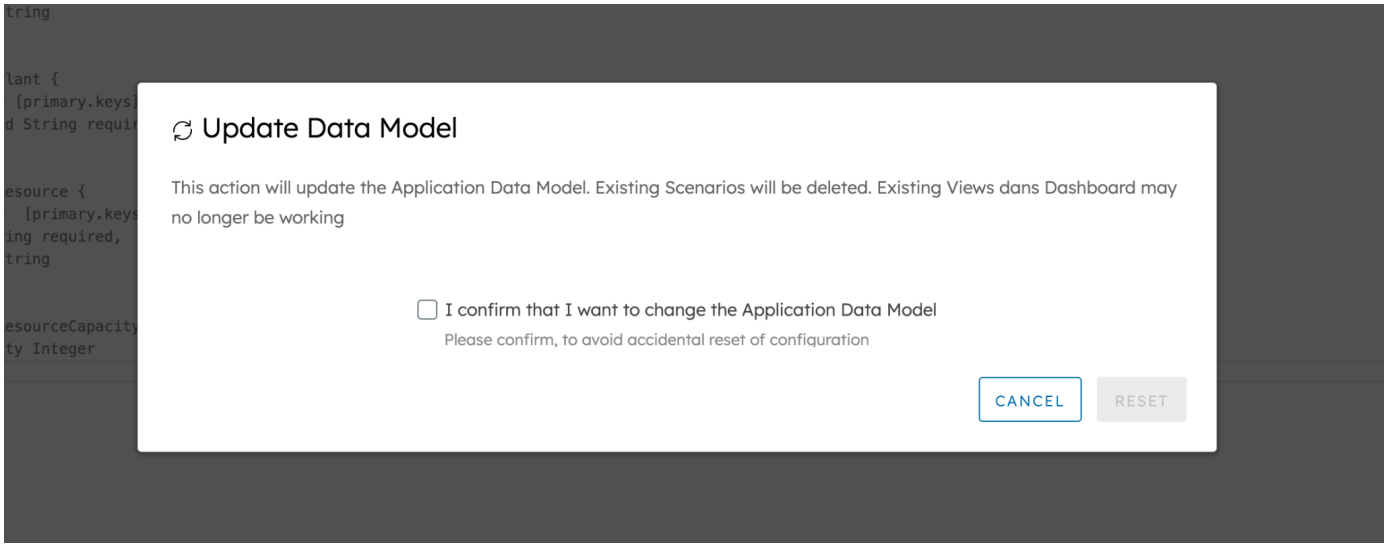
New Gene Online Beta Feature

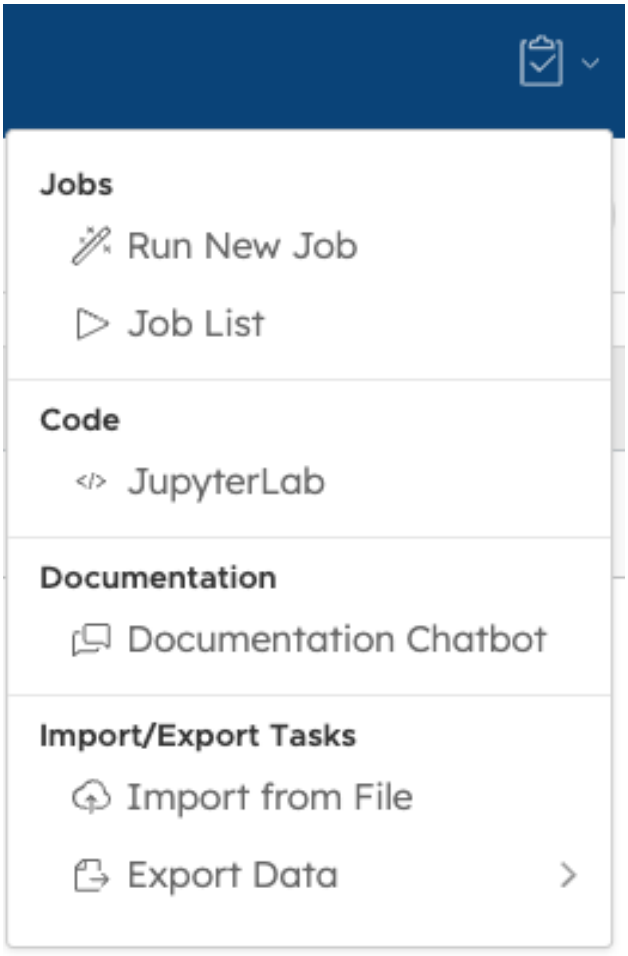
Gene Online Beta 4.4.0 is now available. It allows developers to deliver DOC-based applications faster, through a development cycle relying on the low-code UI and out-of-the-box configurations compared to a local installation.

There is no need for a development environment as an empty customizable application can be launched directly from the web.



Once the web client starts, users can quickly set the application data model through the JDL Editor, configure the application as usual, and use the JupyterLab integration to edit and run optimization/ML code from an online editor.





New Documentation Chatbot Beta Feature

The Documentation Chatbot now allows asking questions on the application features based on the Jira tickets, source code, and selected specification documents of the project.

Usage instructions are in the file `EXPERIMENTAL_FEATURES.md` generated for applications with the experimental features enabled.

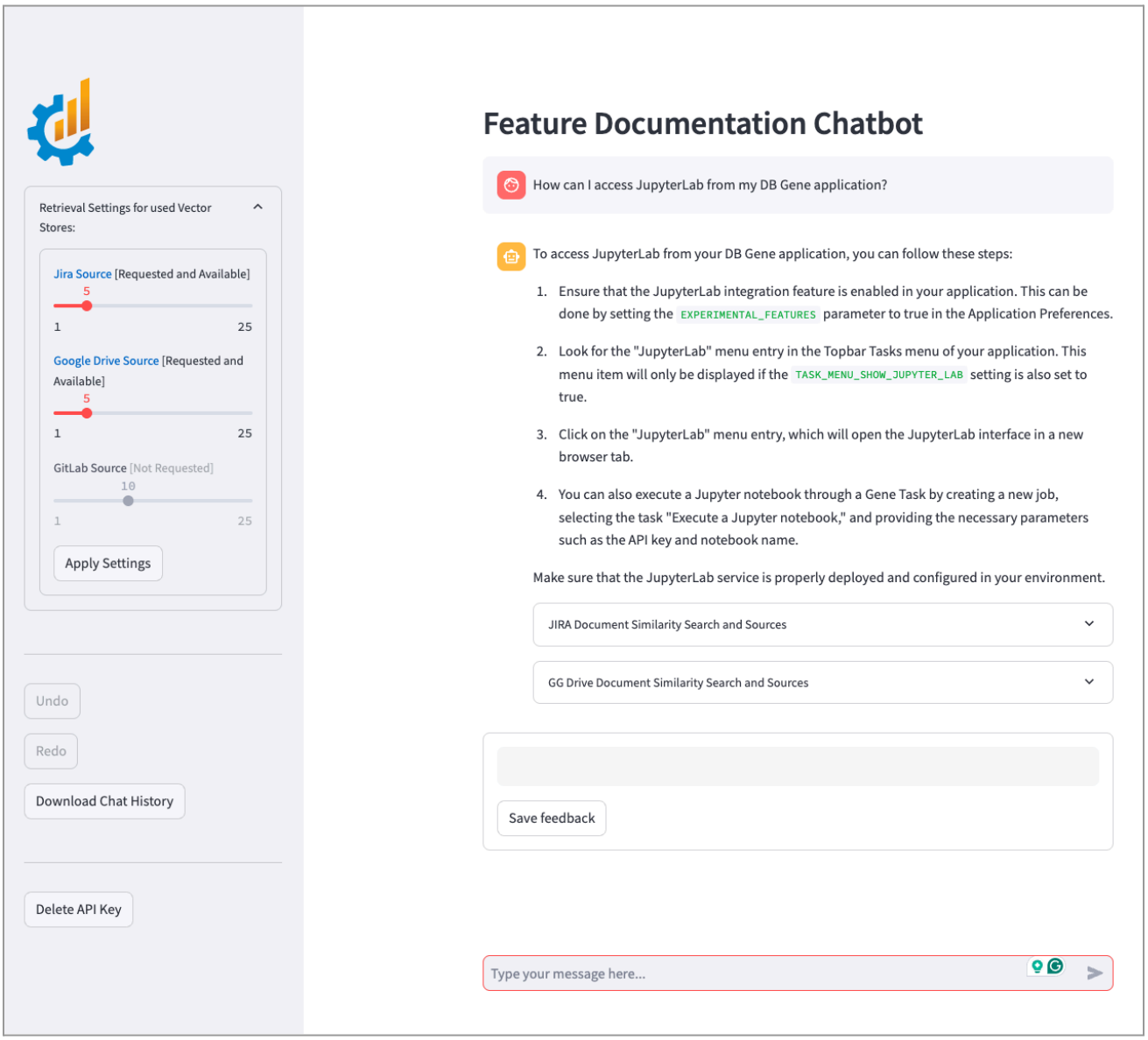
To activate this chatbot, follow the instructions in `documentation-chatbot/build.gradle`. Then, the build chain will scan the data sources (Jira, Gitlab, and/or Google Drive) to create vector stores and inject them into the Docker image for the chatbot.

From the application web client, users can open the chatbot via a command in the Task menu.

To display this menu, in the Application Preferences, the parameters *EXPERIMENTAL_FEATURES* and *TASK_MENU_SHOW_DOCUMENTATION_CHATBOT* must both be set to true.

Also, an OpenAI API key is required. It can be set as a Spring property in the `application.yml` file of the Backend Service extension, or provided at deployment time in the `app/.env` file for Docker Compose deployment, or in the values file of the deployment configuration when deploying with Kubernetes.

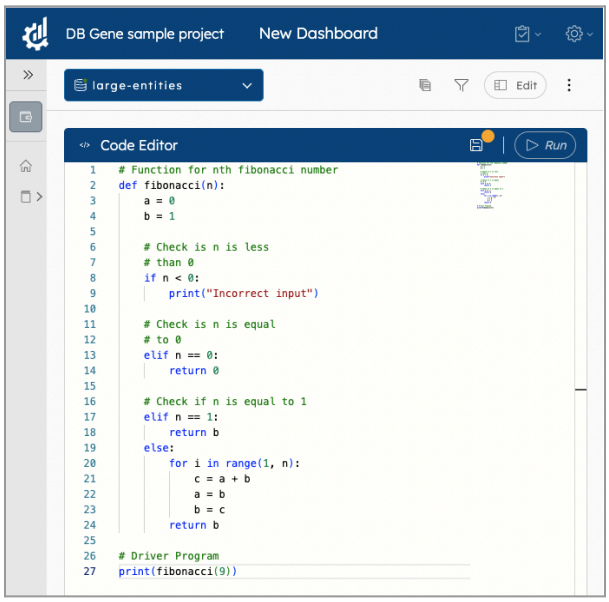
If no key is provided, it can be indicated using the Documentation Chatbot from the web client.



New Code Editor Widget

The new Code Editor widget, based on the Monaco widget from the VS Code project, allows users to:

- Edit and save free text with syntax highlighting capabilities in many languages.
- Use the text as input for a button in the widget toolbar to trigger an action such as running a task.
- Discuss with ChatGPT.



The content of a Code Editor widget is read from and saved as a parameter in a row of the `GeneParameter` entity of the current scenario. The `name` of the parameter can be edited in the widget configurator.

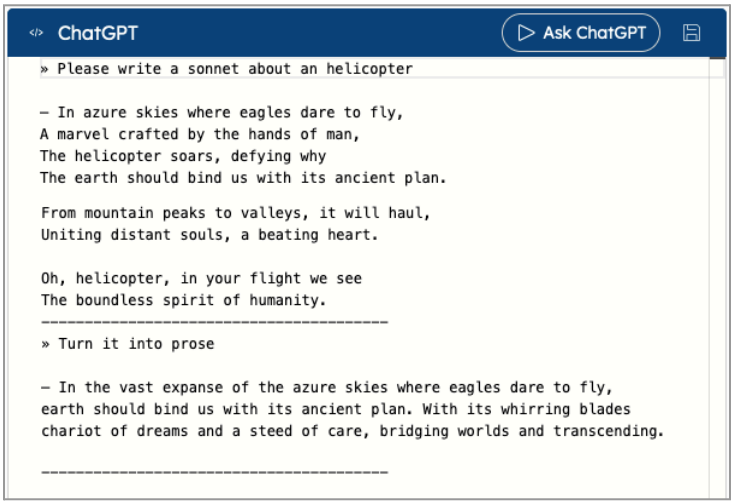
New Code Editor Task Input

Users can configure a button in the widget toolbar to trigger an action, such as running a task or any user-defined routine created beforehand.

Default tasks, allowing to 'Execute a Ruleset on a Scenario' for example, use the ID of the current scenario which can be edited to indicate a predefined scenario.

New Code Editor Chat GPT Task

The Code Editor widget can be configured with a button that launches a new default task `ProcessChatGptConversationTask` which turns the widget into a ChatGPT discussion terminal.

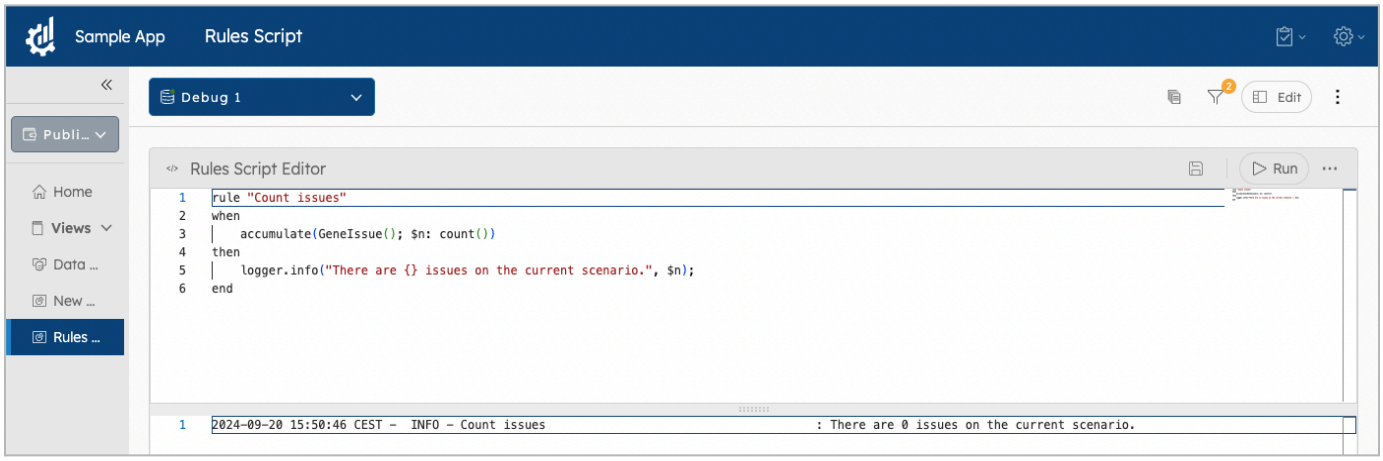


Markers » and – are used to differentiate the question and the answer, with question-answer pairs separated by lines of dashes.

Questions can be added below the last dashed line and sent to ChatGPT by clicking the button.

New Rule Script Editor Widget

The new Rule Script Editor widget allows executing operations, coded as Drools rules[1], on scenario data.



A Drools ruleset can be configured as follows:

Java

```drools
rule "Name of the rule, between quotes"
when
    $optionalVariable: ClassName(field == value, otherField < someValue)
    ...
then
    // some Java-like code using the $variables defined in the 'when' part
end
```

Example 1: Detecting and reporting issues

The rule engine tries to match each condition in the 'when' part with an entity instance in the scenario. For each successful match, the 'then' part is executed. An example of a (admittedly silly) rule that adds an issue if two employees have the same first name is:

```drools
rule "First names of employees should be distinct"
when
    $empl1: Employee($first1: firstName)
    $empl2: Employee(this != $empl1, firstName == $first1)
then
            helper.addIssue($empl2,   "Employee   has   same   first   name   as   employee
#%s".formatted($empl1.getId()));
end
```

[1] For more details on the syntax of the Drools Rule Language (DRL), which is used with the Java-based Drools Rules Engine, please refer to the Official Drools Documentation.

Any change that a rule performs is saved in the scenario.

All compilation and execution messages are emitted as an output of the routine named `executionLogs` and added to the logs. If the ruleset was retrieved from a row of the `GeneParameter` entity, the messages are stored in another row with the same `name`, plus the `-logs` suffix.

One way to extend the widget is to invoke user-defined routines in tasks. Note that a routine compiles the ruleset on each invocation and keeps the content of the scenario into the working memory of the rule engine. Depending on the use case, these elements may impact performance.

Another way to extend the widget behavior is to extend the `helper` class as two global variables are predefined. These variables can be used in the rules whenever useful.

- The `helper` global variable is bound to an instance of the `RulesCollectorHelper` class, which is scaffolded in the Backend Service extensions module. It natively contains a few methods to create issues (instances of `GeneIssue`) and add them to the collector. It is possible to add more utility methods that could be useful to the project at hand. These methods can then be called in the `when` or `then` parts of the rule. Note that methods with a side effect should not be called in the `when` part.
- The `dataset` global variable is bound to the collector that reflects the scenario.

Improved Exchange Formats

Version 4.4.0 introduces the following changes in terms of exchange format for scenario data:

- XCSV is the new exchange format supporting the Composite Data Model (CDM); it is now the standard exchange format in jobs for scenarios.
- Data integration is now available for the XCSV format.
- XCSV import and export REST APIs are now available.
- XCSV is now the default scenario data format in tasks.
- XCSV can now be loaded into Python collectors.
- XCSV can now be loaded into Pandas data frames.
- Pandas data frames can now be saved into XCSV.

Improved Permissions

When a user creates a scenario link or a CDM scenario referencing a scenario, a permission rule giving access to the referenced scenario is created. This allows users to revoke access to a specific scenario referenced through scenario links or other CDM scenarios.

Users can only display the shared scenario when they have access to the scenario workspace or when they have access to the created scenario link. Scenarios created before 4.3.0 and referenced by one or more scenario links are automatically migrated and associated with a permission rule giving access to the scenario. The new permission rule will override any existing one.

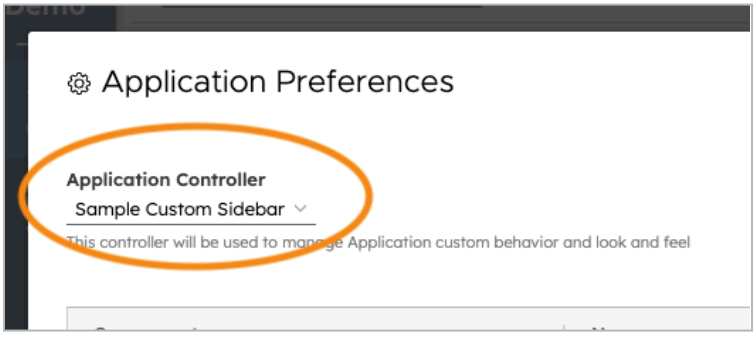
Improved Application Controller

The Application Controller API has been improved in several ways.

Improved Application Controller Configuration

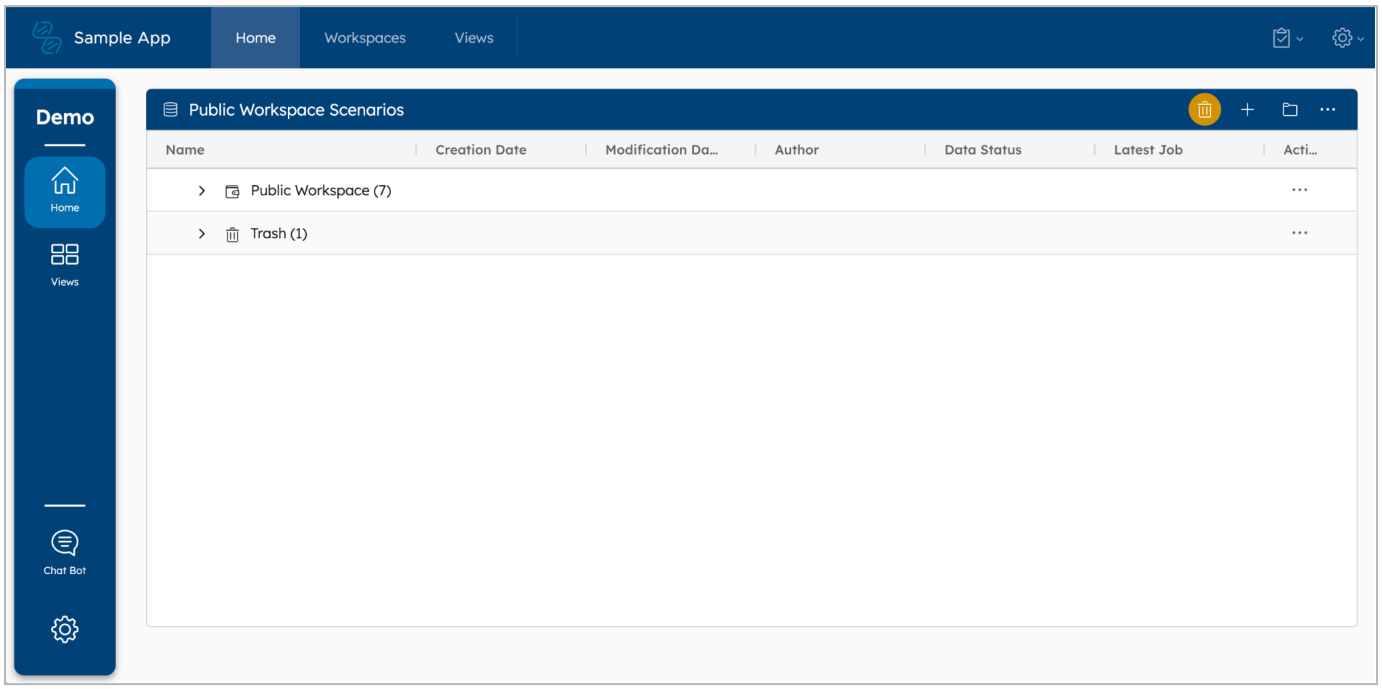
Formerly an application controller required some code to be activated. Now, the application controller mechanism follows the same workflow as the widget controllers:

- Developers register one or more application controllers by code using the provided samples.
- From the Application Preferences, users can use one of the available application controllers.



New Topbar and Sidenav Parameters in the Application Controller

The `GeneApplicationConfiguration` interface, which can be customized through the `GeneApplicationController` API, Two new properties allow customizing the Topbar and the Sidenav components and can be set with any Angular component that will replace the default provided ones.

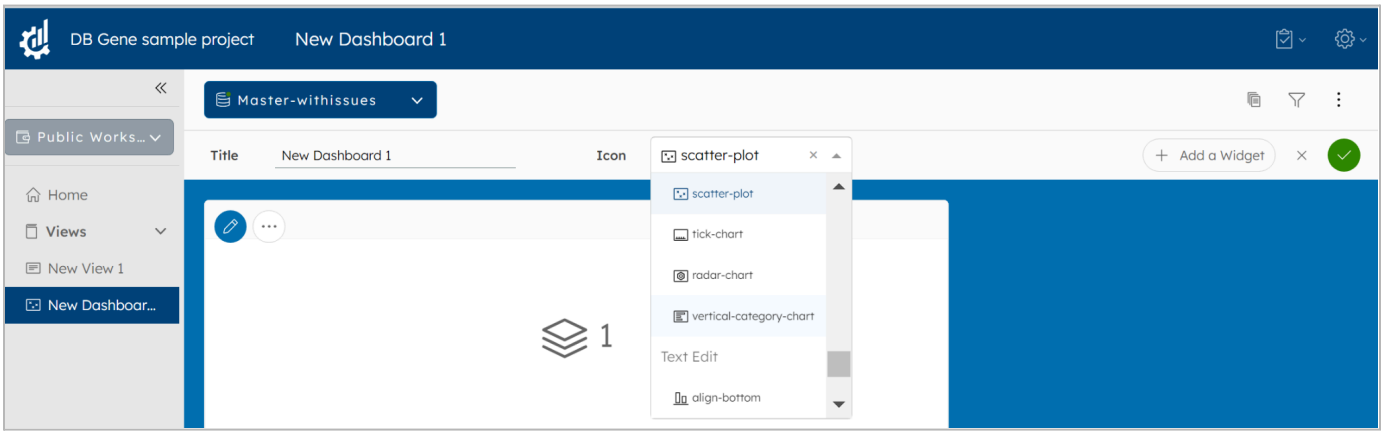


Improved Dashboards

The way in which data displayed in the web client is filtered has been improved in several ways.

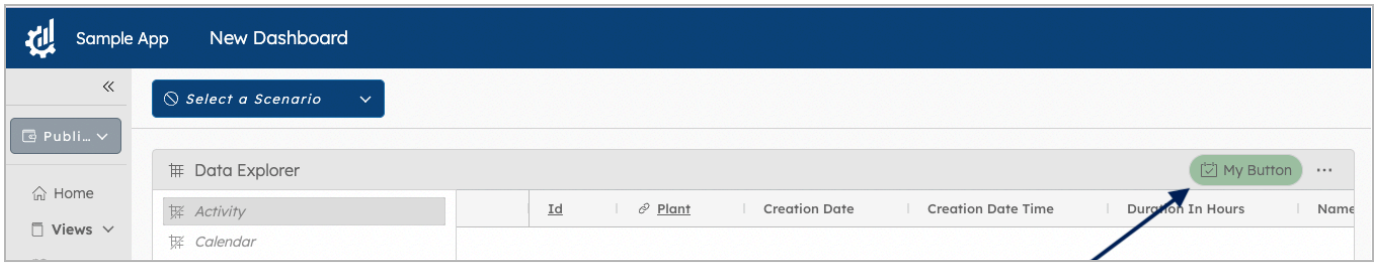
Improved Sidenav Icons

When editing the layout of a custom view or dashboard, users can now configure an *Icon* in addition to the *Title* that is displayed in the Sidenav.



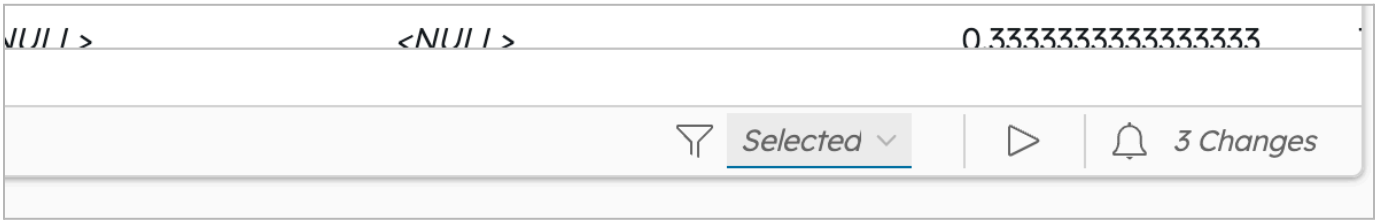
New Configurable Buttons for the Widget Toolbar

Every widget configurator now displays a new *Toolbar* tab. This tab allows users to add action buttons, with labels, custom colors, and icons, to the toolbar of any widget.



New Status Bar API

The Gene Status Bar API, which allows adding a status bar at the bottom of some widgets, is now available. The `SampleStatusBarController` sample is available by default.

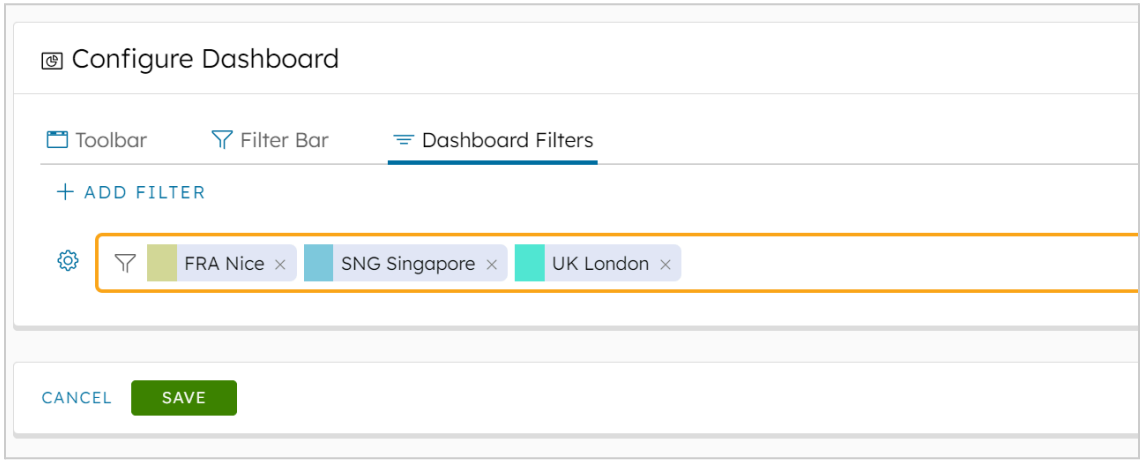


Improved Filtering

In addition to the Data Grid/Explorer advanced filters, filtering across widgets and dashboard widgets has been improved in several ways.

New Dashboard Filters

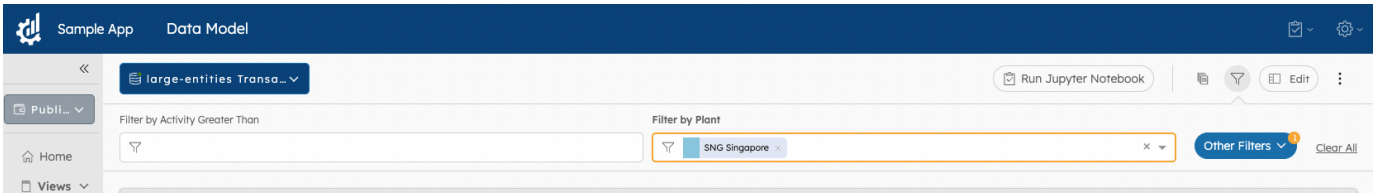
The option *Configure Toolbar*, available on custom dashboards, has evolved into *Configure Dashboard*. It introduces a new *Dashboard Filters* tab that allows setting permanent filters specific to the dashboard.



New Filter Bar

A new Filter Bar can now be displayed on all custom dashboards. It provides users with additional dynamic filtering on top of any permanent filters already set on the dashboard or on the widgets.

When using the Filter Bar, filters from other dashboards now show up in the *Other Filters* dropdown instead of in the *Context Selection* dropdown, which has now become a toggle button.



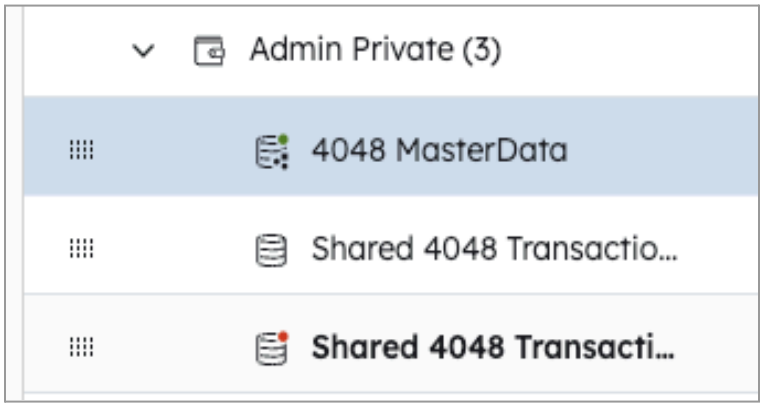
New Filtering Scope

In the Application Preferences, the new *FILTER_SCOPE* parameter is set by default to *GLOBAL*. This means that the Filter widgets and new Filter Bar apply to all views and dashboards across the application.
It can be set to *VIEW* to limit the filtering to the dashboard or view of the element.

New Boolean Filtering

The Filter widget and the Filter Bar now allow filtering Booleans.

Improved Scenario List Widget

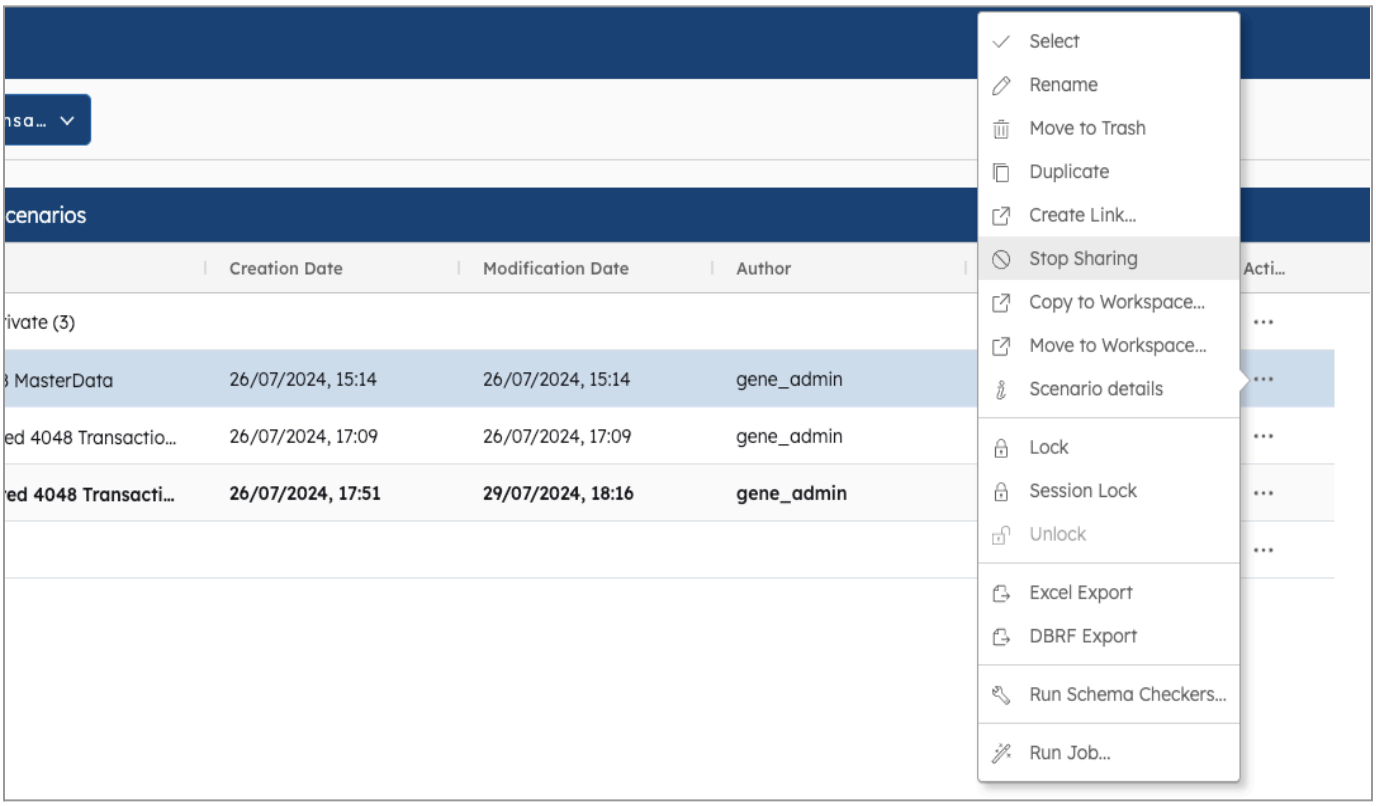


New Icon in the Scenario List Widget

In the Scenario List widget, an icon now indicates when a scenario is shared with users, i.e. via a scenario link or when referenced by another scenario in a composite data model application. Also, in the Scenario List widget, a new option now allows to *Stop Sharing* a scenario.

New Custom Actions in the Scenario List Widget

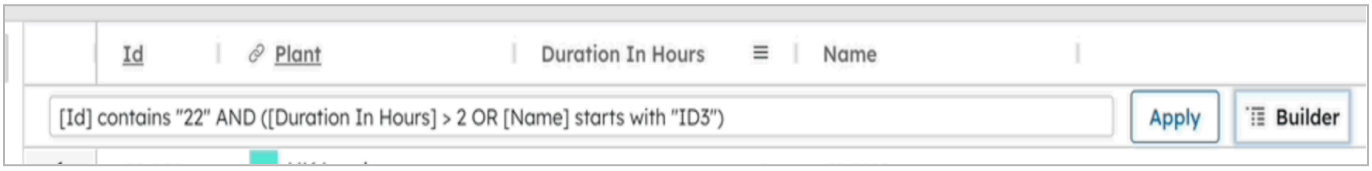
Custom actions defined via the Action API, introduced in version 4.1.0, are now available in the Scenario List widget from the menu Actions.



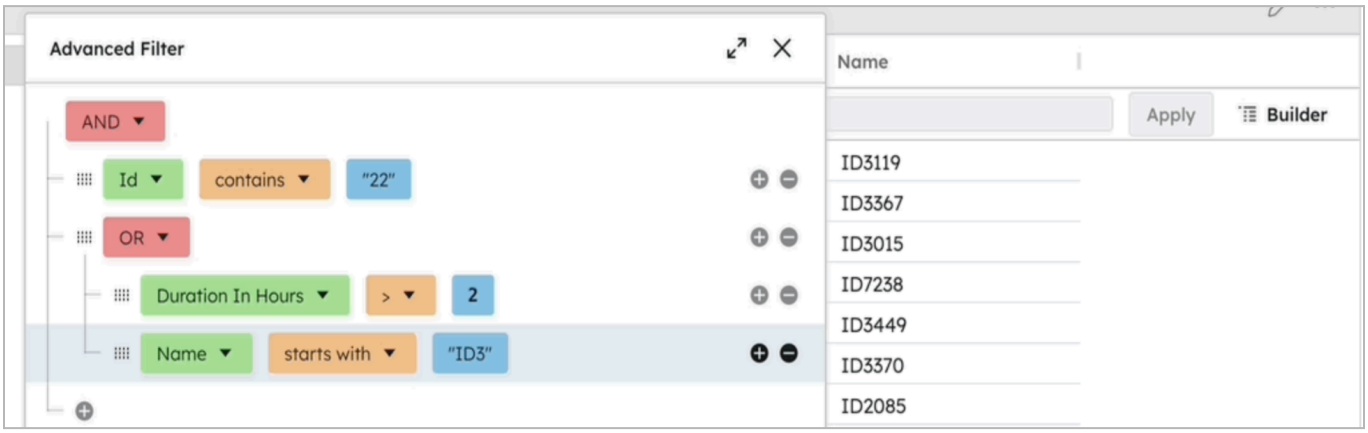
Improved Data Grid/Explorer Widget

The most notable change in version 4.4.0 Data Grid/Explorer widgets is the introduction of new experimental advanced filters.

It displays an input field beneath the column headers that allows for the combination of several filters.

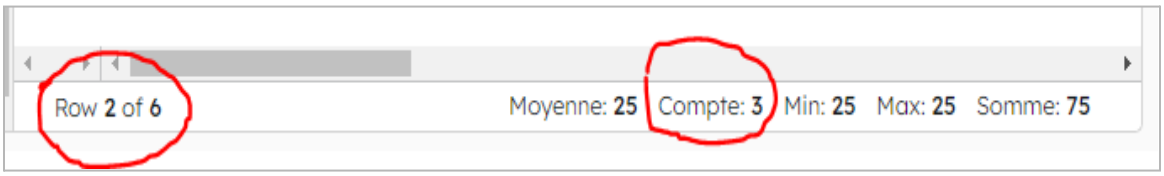


In addition, a *Builder* option helps users visualize the filter hierarchy.



In addition:

- It is now possible to use the option *Delete all rows* when not in Edit mode and for entities from referenced scenarios.
- The option *Hide aggregation panel* is now enabled by default.
- `DateTime` columns can now be filtered by `Date` only.
- Undefined Boolean values are now shown as *<NULL>*.
- It is now possible to localize the index message at the bottom of the widget.



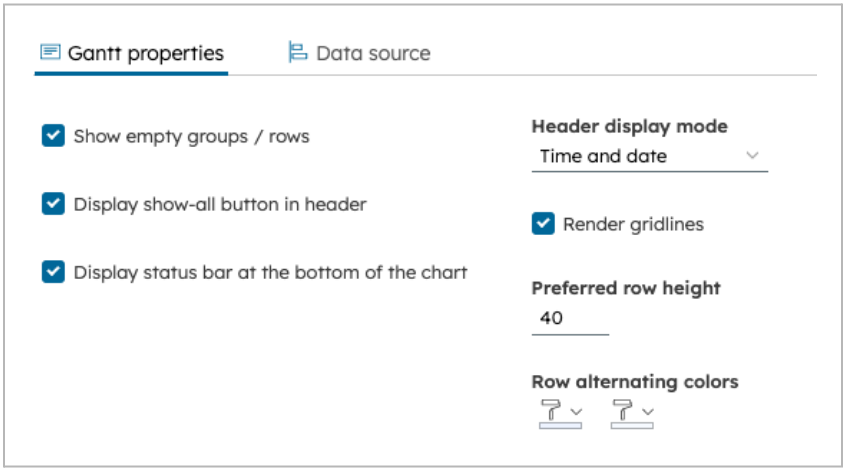
Improved Gantt Chart Widget

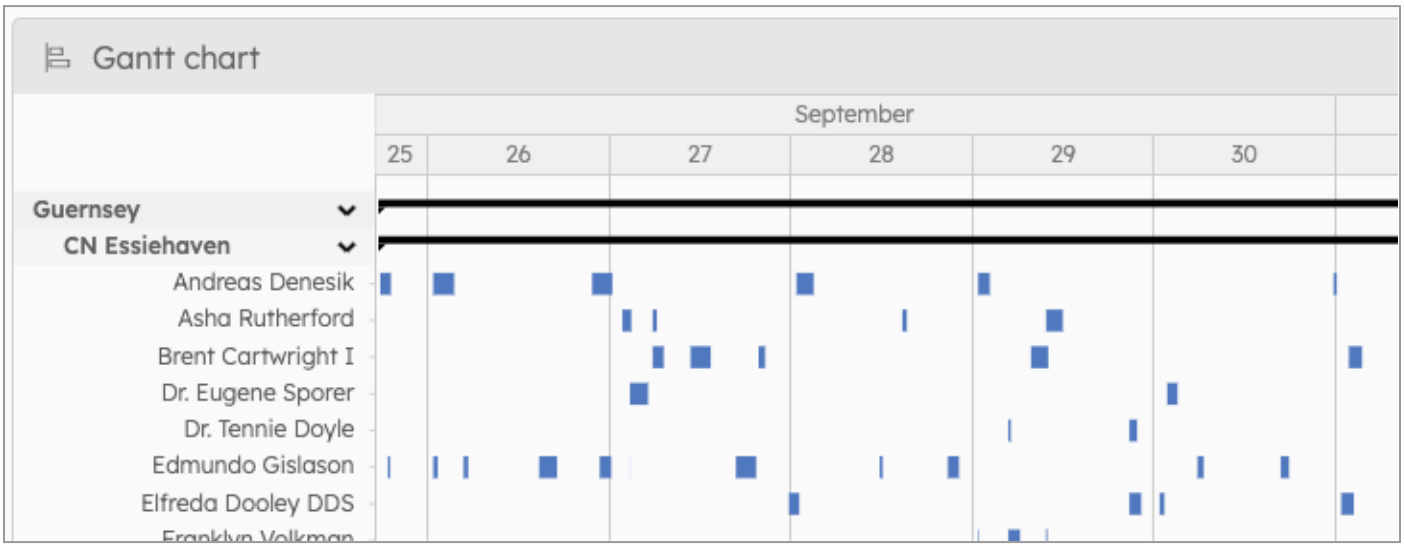
Several improvements have been introduced to the Gantt Chart widget:

- New options for row height and alternating colors are now available.
- The zoom level and scroll position are now saved after leaving the view or dashboard.
- The Gantt Chart widget now provides users with the same *Time axis* option as the Chart widget, allowing them to set manual or automatic scaling, i.e. *hour and day*, *day and month*, and so on.
- In addition, just like for the Data Grid/Explorer widgets, it is now possible to activate the selection of events and resources displayed in the widget, on the *None* / *Select* / *Highlight* basis.

Finally, the Gantt Chart widget now also allows:

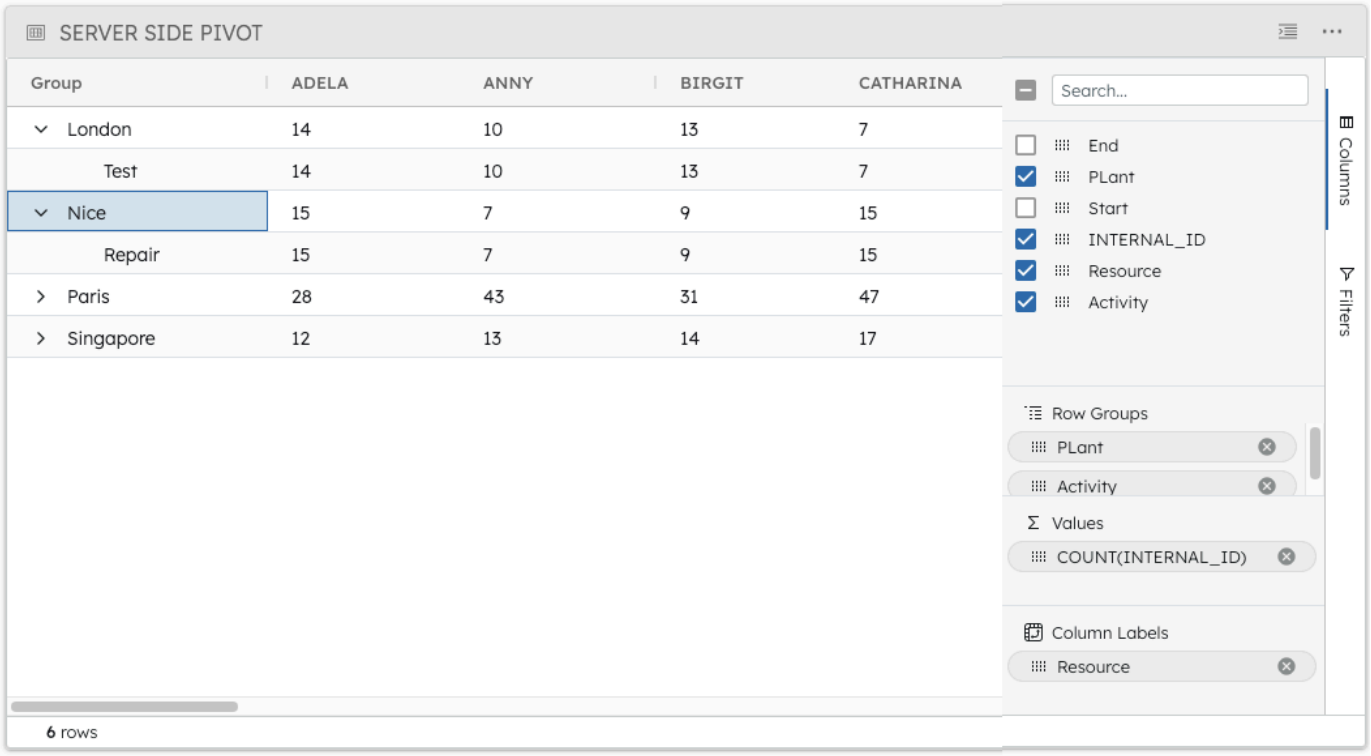
- setting a renderer for the Event label using a custom controller.
- Identifying resources and groups by `internalId`.
- Setting a custom context menu.
- Loading events dynamically.
- Grouping resources.





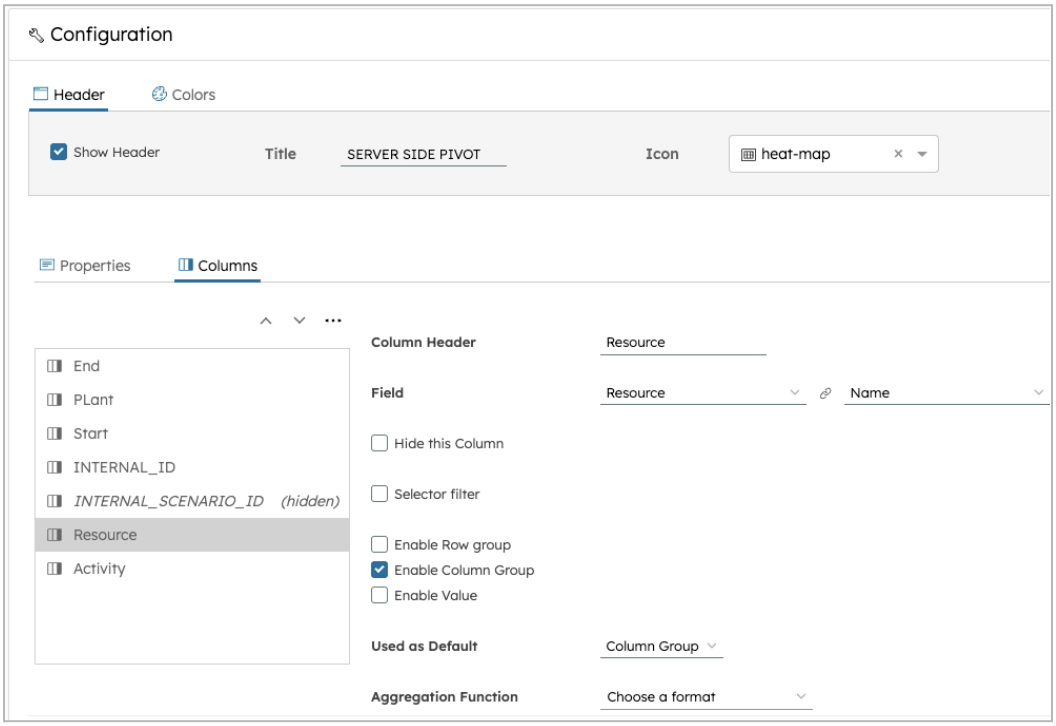
Improved Pivot Table Widget

DOC 4.4.0 introduces a new version of the Pivot Table widget.



This new version allows using a server-side data source.

It leverages the AG Grid enterprise Pivot feature and can handle large scenarios.



Improved Map Widget

In the Map widget, the following custom controller method now allows overriding the marker series tooltip.

Unset

```
getMarkerPopupHtmlContent = (markerInfo: MarkerInfo): string => {
    return `
        This <i>${markerInfo.tag}</i> is located at
        <ul>
            <li><b>Latitude</b>: ${markerInfo.position.latitude}</li>
            <li><b>Longitude</b>: ${markerInfo.position.longitude}</li>
        </ul>
        }
```

Technical Features

DOC 4.4.0 introduces several technical improvements regarding JSON object size limit, security, and memory usage.

DOC 4.3.0 introduces several technical improvements regarding login security, Python worker debugging, CPLEX integration, and GraphQL introspection.

Improved Jackson JSON Serialization Capabilities

The `jackson-core` library in version `>=2.15` introduced a configurable size limit for the serialized JSON object with a default value of `20MB`. This change in the `jackson-core` library used by DOC produced issues in some applications while transferring large scenarios between the Execution Service and Backend Service.

An application property now allows configuring the `jackson-core` size limit. This setting can be overridden and increased for both services when dealing with large scenarios.

Therefore, the value for both the Execution Service and Backend Service is now set to `250MB` by default. The value can be edited but should be the same in both the following files:

- `extensions/execution-service-extension/src/main/resources/application.yml`
- `extensions/backend-service-extension/src/main/resources/application.yml`

```
Unset
  # Configure the maximum size of a JSON payload. Used to send and receive
  input/outputs to/from a routine.
  gene:
    object-mapper:
      stream-max-string-length: 250000000 # 250MB
```

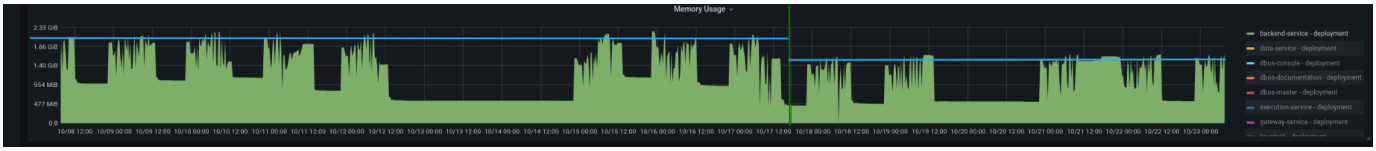
For more details, please refer to the Jackson Core Github page.

Improved Security for Trivy CVEs

Trivy CVEs are now fixed using RabbitMQ 4.0.2.

Improved Java Memory Usage Limit

Java native memory usage is now limited using the `MALLOC_ARENA_MAX` parameter, with negligible to no loss of performance.



Improved Login Security

The password must now be alphanumeric, contain a special character and respect a minimal length of 8 characters. The username is excluded from possible passwords. In addition, Keycloak is now configured by default with a stronger password policy limited to 5 failed login attempts.

Improved Debugging for Python Workers

It is now possible to set up breakpoints in a Python worker during development.

Improved CPLEX Integration

The integration of CPLEX Studio Home is now easier as, instead of manually combining the files of the `dockerPrepare` task and Linux CPLEX binaries to build, developers only need to have the CPLEX binaries archive and configure its path in a Gradle property before using `./gradlew docker`.

Improved GraphQL Introspection

GraphQL introspection feature is now disabled by default for the Data Service and Scenario Service as it is recommended to use it only for development. To enable introspection, use the following Spring property:

```
Unset
  spring:
    graphql:
      schema:
        introspection:
          enabled: true
```

Changelog

DOC 4.4.0 introduces several improvements and bugfixes listed in detail below.

Improvements

Improvements 4.4.0

DOC 4.4.0 introduces the following improvements compared to the unreleased version 4.3.0:

Data *Model*	*DBPF-6544*	*Internal Types are now exported in Excel for Composite Data Model applications*
Data *Scenario Service*	*DBPF-7096*	*In the Scenario Service GraphQL API, the "scenarioReferenceGraph" property has now been removed from the "Path" object*
Dev *3rd-party Components*	*DOC-842*	*Jackson "streamMaxStringLength" property is now configurable and set with 250MB default value*
	DBPF-7349	*platform-common-lib now relies on Spring Boot 3.3.5*
	DBPF-7153	*DOC now relies on Keycloak 25.0.5*
	DOC-328	*DOC now relies on AG Grid 32.3.1*
	DBPF-7150	*DOC now relies on Angular 18.2.10*
	DBPF-6850	*DOC is now compatible with CPLEX 22.1.2*
	DBPF-7098	*DOC now relies on Keycloak 26.0.5*
	DBPF-7162	*DOC is now compatible with Python 3.12.0*
	DBPF-7099	*DOC now relies on RabbitMQ 4.0.2*
	DBPF-7097	*DOC now relies on SpringBoot 3.3.5, Spring Cloud 2023.0.3, and Spring Framework 6.1.14*

Dev Deployment	DOC-967	Java native memory usage is now limited using the MALLOC_ARENA_MAX parameter
Dev Documentation Chatbot	DBPF-7165	The Documentation Chatbot is now available
Dev Gene Online	DBPF-7101	Gene Online Beta 4.4.0 is now available
Dev JupyterLab	DBPF-7137	The application now provides users with a Jupyter Notebook example to import scenario data into Pandas Data Frames
Dev Security	DBPF-7183	Trivy CVEs are now fixed using RabbitMQ 4.0.2
UI Extensibility	DBPF-7254	The Gene Status Bar API, which allows adding a status bar at the bottom of some widgets, is now available
UI Code Editor	DBPF-7189	A scripted task now allows using the Code Editor widget as a ChatGPT discussion terminal
UI Data Grid/Explorer	DOC-938	In the Data Grid/Explorer widgets, "DateTime" columns can now be filtered only by "Date"
	DBPF-7266	In the Data Grid/Explorer widgets, undefined Boolean values are now shown as <NULL>
UI Filter	DOC-894	The Filter widget and the Filter Bar now allow filtering Booleans

UI *Gantt*	*DOC-492*	*The Gantt Chart widget now allows grouping resources*
	DBPF-7156	*The Gantt Chart widget now identifies resources and groups by "internalId"*
	DBPF-7234	*The Gantt Chart widget now supports context menu customization*
	DOC-606	*The Gantt Chart widget now supports dynamic loading of events*
UI *Pivot Table*	*DBPF-7295*	*Pivot Table (Experimental) is now renamed to "Pivot Table" and "Pivot Table - Legacy" is no longer available in the widget dropdown list*

Improvements 4.3.0

Unreleased DOC 4.3.0 introduced the following improvements compared to the unreleased version 4.2.0:

Application *General*	*DBPF-6795*	*Action buttons with labels and icons can now be added to widget headers*
Application *Views & Dashboards*	*DBPF-6660*	*Filtering can now apply to one or all views and dashboards across the application*
	DBPF-6614	*Permanent filters are now available for custom views and dashboards*
	DBPF-6971	*Users can now configure Sidenav icons for each view and dashboard*
Application *Workspaces & Scenarios*	*DBPF-6448*	*Existing referenced scenarios are now checked and migrated at startup*
	DBPF-6564	*Scenarios referencing a deleted scenario are now marked as "Corrupted"*
	DBPF-6463	*Scenario duplication has been improved for CDM applications*
	DBPF-6450	*Users can now select a scenario link as a reference when adding a CDM scenario*
Data *Built-in Import/Export*	*DBPF-6705*	*XCSV import and export REST APIs are now available*
Data *Data Integration Framework*	*DBPF-6698*	*Data integration is now available for the XCSV format*
	DBPF-6784	*Pandas data frames can now be saved into XCSV*
	DBPF-6790	*XCSV can now be loaded into Pandas data frames*

	DBPF-6789	*XCSV can now be loaded into Python collector*
Data *GraphQL*	*DBPF-5666*	*GraphQL introspection feature can now be disabled*
Dev *3rd-party Components*	*DBPF-5940*	*Clarity packages now rely on more recent versions*
	DBPF-6413	*DBOS now relies on NGINX 1.26.0*
	DBPF-6386	*DOC now relies on AG Grid 31.2.0*
	DBPF-6385	*DOC now relies on Angular 17.3.3 and AG Grid 31.2.0*
	DBPF-6384	*DOC now relies on Angular 18.2.2*
	DBPF-6485	*DOC now relies on Apache Commons Compress 1.26.1, Apache Commons Text 1.12.0, and Apache POI 5.2.5*
	DBPF-6570	*DOC now relies on Gradle 7.6.4*
	DBPF-6381	*DOC now relies on Keycloak 24.0.2*
	DBPF-6412	*DOC now relies on NGINX 1.26.0*
	DBPF-6382	*DOC now relies on RabbitMQ 3.13.0*
	DBPF-6377	*DBOS now relies on Spring Boot 3.2.4 and Spring Cloud 2023.0.1*
	DBPF-6383	*DOC now relies on Spring Boot 3.2.4 and Spring Cloud 2023.0.1*
Dev *Documentation*	*DOC-894*	*Documentation on all widget configurators is now available*
	DOC-920	*Documentation on how to prevent scenarios from being moved to the Lost and Found workspace is now available*
	DOC-834	*Documentation on Prometheus endpoints is now improved*
Dev	*DBPF-6871*	*A base Docker image for JupyterLab is now available for DOC*

Build	*DBPF-6872*	*Gradle modules are now improved*
	DOC-761	*The dependencies in "package.json" are now sorted alphabetically*
	DBPF-4735	*DOC now relies on the latest "code-replicate" plugin*
	DBPF-6791	*The scaffolding now provides the user with an option to enable or disable the generation of beta features*
	DBPF-6767	*The scaffolding of a JupyterLab module is now only available as a beta feature*
Dev *CPLEX*	*DBPF-6708*	*The CPLEX integration is now improved*
Dev *Deployment*	*DBPF-6763*	*The Helm chart now allows deploying the JupyterLab service*
	DOC-931	*The scaffolded Helm chart now allows users to pass arguments to the "mongodb" command line*
	DOC-910	*The Helm chart parameter "InitialRAMPercentage" is now removed from "/deployment/helm/src/values.yaml"*
Dev *Python*	*DBPF-6478*	*It is now possible to set up breakpoints in a Python worker during development*
	DBPF-6760	*JupyterLab integration is now available as a beta feature*
Scripted Tasks *Definition*	*DBPF-6797*	*Generic tasks and routines to call a rule engine on a scenario are now available*

	DBPF-6627	*Scenario creation is now available in all scripted tasks*
	DBPF-6658	*Scenario creation tasks now allow setting scenario characteristics*
	DBPF-6634	*Scripted tasks now allow automatic scenario creation*
	DBPF-6242	*The sample task "Create an empty scenario" is now improved for CDM applications*
	DBPF-6700	*XCSV is now the default scenario data format in tasks*
UI *Code Editor*	*DBPF-6794*	*The Code Editor widget is now available as a beta feature*
	DBPF-6796	*The "Run" button of the Code Editor widget now passes the script name as its second argument*
UI *Data Grid/Explorer*	*DBPF-6709*	*In the Data Grid/Explorer widgets, advanced filters are now available as an experimental feature*
	DOC-917	*It is now possible to localize the message "row X of Y" at the bottom left of the Data Grid widget*
	DBPF-6981	*In the Data Grid/Explorer widgets, the option "Delete all rows" is now available for entities from referenced scenarios*
	DOC-686	*In the Data Grid/Explorer widgets, the option "Hide aggregation" panel is now enabled by default*
UI *Extensibility*	*DBPF-6716*	*The Application Controller API is now improved*
UI	*DBPF-6640*	*A new Filter Bar is available for custom dashboards from the new option Configure Dashboard*

Filter Bar	*DBPF-6641*	*When using a Filter Bar, filters from other dashboards now show up in the "Other Filters" dropdown*
	DBPF-6642	*When using a Filter Bar, the Context Selection dropdown now switches to a toggle button*
UI *Gantt*	*DOC-742*	*In the Gantt Chart widget, new options for header colors and row height are now available*
	DBPF-5423	*The Gantt Chart widget API now relies on REST instead of GraphQL to improve loading time*
	DOC-672	*The Gantt Chart widget now displays hours on the x-axis when needed*
	DBPF-6720	*The Gantt Chart widget now allows selecting events and resources within its display area*
	DBPF-6719	*The Gantt Chart widget now allows setting a renderer for the Event label using a custom controller*
	DBPF-6934	*The Gantt Chart widget now saves the zoom level and scroll position*
UI *Map*	*DOC-903*	*In the Map widget, custom controllers now allow overriding the marker tooltip*
UI *Pivot Table*	*DOC-837*	*The Pivot Table widget now allows using a server-side data source*
UI *Scenario/ Workspace List*	*DBPF-6447*	*In the Scenario List widget, an icon now indicates when a scenario is shared with all users*
	DBPF-6451	*In the Scenario List widget, an option now allows to "Stop sharing" a scenario*

Improvements 4.2.0

Unreleased DOC 4.2.0 introduced the following improvements compared to version 4.1.0:

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*	*DOC now relies on AG Grid 31.0*
	DBPF-5669	*DOC now relies on PostgreSQL 15.5*
	DBPF-5670	*DOC now relies on Java JDK 17.0.9*
	DBPF-5939	*DOC now relies on Angular 17.2*
	DBPF-5941	*DOC 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

Bugfixes 4.4.0

DOC 4.4.0 introduces the following bugfixes compared to the unreleased version 4.3.0:

Data *Built-in Import/Export*	*DOC-996*	*In some cases, the application was not importing DBRF/ZIP scenario files*
	DBPF-7324	*When a scenario was being imported in the background, using the option "Create a new scenario from the same file" was resetting the dialog*
	DBPF-7411	*When adding a scenario, the dialog was allowing to type in the "Scenario type" input field*
Data *Database*	*DOC-947*	*After a deadlock, the application was not properly retrying to fetch some data*
	DOC-944	*Cancelling a schema checker job was producing a numerous logs*
	DOC-962	*During a transactional query, the application was not notifying changes if all creation, update and deletion parts were not present*
	DBPF-7274	*In some cases, the schema checker was triggering an NPE*
	DBPF-7390	*When saving a CSV scenario file into a Composite Data Model application, the file was incorrectly processed as a standard CSV file, resulting in the loss of relationships*

Data *Scenario Service*	*DOC-961*	*Scenario duplication was failing after resetting workspaces and folders*
	DBPF-7389	*The endpoint to retrieve a reference graph from the Scenario Service was not working properly*
	DBPF-7404	*The Execution Service was receiving 403 errors from the Scenario Service when creating scenario events due to missing job ID*
DBOS *Master*	*DBPF-3102*	*If RabbitMQ restarted during a DBOS job execution, the Execution Service was continuously stacking due to missing cleanup of anonymous queues*
	DBPF-6409	*Workers were failing to finish on Windows due to incorrect handling of file paths in URI resolution*
Dev *3rd-party Components*	*DBPF-7365*	*Spring BOM was missing in "test-fixtures" dependencies, causing issues with version resolution*
	DBPF-7211	*The "npm-login.sh" script was not working if an old version of Node JS was installed*
	DBPF-7071	*The Keycloak login was not displaying properly due to a CSS error*
Dev *REST API*	*DBPF-7094*	*Some files were missing from "public-api" exports*

Dev *Security*	*DOC-932*	*Prometheus endpoint 'scenario_sessions_actives' was displaying inaccurate session counts when messages were lost*
Scripted Tasks *Jobs*	*DBPF-7028*	*The routine allowing to execute rules was not working properly*
UI *Application Preferences*	*DBPF-7221*	*The application was not taking some Application Preferences settings into account*
UI *Code Editor*	*DBPF-7109*	*When using the "Autosave" option, the Code Editor widget was not working properly*
UI *Data Grid/Explorer*	*DBPF-1767*	*In some cases in a Data Grid/Explorer widget based on a custom entity, the configurator was incorrectly displaying the "Is editable" checkbox*
	DBPF-7058	*Editing a Data Grid was triggering two refreshes of the dashboard*
	DOC-973	*The application was using incorrect color hints for nested fields*
	DBPF-7072	*Using a space character as a filter for text columns was causing the Data Grid/Explorer widget to crash*
	DOC-971	*When using a Data Grid in "Selection" mode, enabling scenario comparison was triggering a JS error*

UI *Filter*	*DBPF-7041*	*"Selection" mode was not properly applying to other dashboards and views despite the filter scope being set to "VIEW"*
	DOC-803	*Having multiple filters on the same Entity within a single Filter widget or Filter Bar was only using the value of the last one in the array*
	DBPF-7042	*In the Filter Bar, the drop-down list "Other filters" was not displaying "Selection" values properly*
	DBPF-6942	*In the Filter Bar, filter drop-down lists were not displaying values properly in the configurator preview*
	DBPF-7040	*When deleting the last filter of the Filter Bar, the configurator was still displaying it in the preview*
UI *Gantt*	*DBPF-6911*	*In some cases in a Gantt Chart widget with overlapping activities, the longer-lasting ones were hiding the shorter ones*
	DBPF-7080	*In some cases when using an existing Gantt Chart widget configuration, the application was displaying an empty chart*
	DBPF-7335	*In the Gantt Chart widget, the resource Selection option was available for a primitive grouping field*
	DOC-991	*In the Gantt Chart widget, the time axis was not aligning properly with the event timestamp*
	DBPF-7100	*The Gantt Chart widget was not properly loading filters when the scope was set to "VIEW"*

UI *Job*	*DBPF-7086*	*In the Job Details widget, the loading spinner was not displaying properly*
UI *Pivot Table*	*DBPF-7012*	*In the Pivot Table widget, the value "Aggregation" option was enabled but was not working properly*
	DBPF-7075	*In the Pivot Table widget, the value "Aggregation" option was not available for "internal_scenario_id"*
	DBPF-7013	*The Pivot Table widget configurator was allowing to save an incomplete configuration, causing the widget to not display properly*
	DOC-941	*The Pivot Table widget was not allowing AG Grid localization*

Bugfixes 4.3.0

DOC 4.3.0 introduces the following bugfixes compared to the unreleased version 4.2.1:

Application *General*	*DBPF-7007*	*Missing dashboards were triggering a high number of errors*
Application *Access Control*	*DOC-817*	*The web client was displaying application elements over which the user did not have sufficient permissions*
Application *Views & Dashboards*	*DBPF-6001*	*In some cases, adding a new widget was replacing an existing one in the process*
	DBPF-6746	*In some cases, it was not possible to delete a workspace*
	DOC-474	*In the Filter widget, setting the Filter Field option to "None" was not enabling the "Field Operator" option accordingly*
Application *Workspaces & Scenarios*	*DBPF-6251*	*Saving entities on a scenario with inaccessible referenced scenarios was triggering an error*
	DOC-843	*In some cases, when using "collector.loadSnapshot", scenarios were losing relations to referenced scenarios*
Data *Built-in Import/Export*	*DBPF-6756*	*Relationships between entities were being lost when importing scenarios*
	DBPF-6609	*The web client was not refreshing after scenario data was modified using the "/data/scenario-import" API*
Data *Data Integration Framework*	*DBPF-6595*	*It was not possible to remove all entities from the Python collector*
	DOC-777	*Python collector loading was failing when the model used the field name "type"*
	DOC-854	*The CRF reader was not recognizing the right file format*

Data *JDL*	*DOC-852*	*Adding decimal min or max values in the JDL was causing the data service to return an error*
	DBPF-6301	*In some cases, schema checkers were not properly working*
Data *Model*	*DBPF-6302*	*Composite data model applications were containing the default scenario type*
Data *Scenario Service*	*DBPF-6475*	*The Scenario Service was logging an error when creating a scenario link*
	DBPF-6410	*The Scenario Service was not loading scenario graph objects through GraphQL*
DBOS *Master*	*DBPF-6809*	*A pod/job that failed in Kubernetes was running forever in DBOS*
	DOC-912	*In some cases, the worker output was triggering a "HeapSpace" error in DBOS Master*
Dev *3rd-party Components*	*DOC-865*	*The Keycloak realm was mentioning Keycloak 14 instead of Keycloak 24*
Dev *Deployment*	*DBPF-6758*	*Helm charts were not defining InitialRAMPercentage*
	DOC-864	*Helm charts were not defining Resource limits for PostgreSQL*
Dev *REST API*	*DOC-853*	*The "excel-export" API was incorrectly marked as deprecated*

Scripted Tasks *Definition*	*DOC-860*	*The "AskInputStatement" of "JobInputType.scenarioId()" was raising an exception if optional and not provided*
Scripted Tasks *Jobs*	*DBPF-6810*	*The job execution was failing with NPE depending on how expressions were written*
Scripted Tasks *Routines (Python)*	*DBPF-6594*	*A partially filled-in Python data frame was causing the function "save_data_frame_dict" to fail*
	DOC-776	*When using Python Collectors, "find by business key" methods were not working properly with compound keys*
UI *Extensibility*	*DBPF-6688*	*It was impossible to properly evaluate the "disabled" state of a Custom Action*
	DOC-884	*When using a custom controller, menus with labels were not rendering correctly*
UI *Scenario/Workspace List*	*DBPF-6864*	*In the Scenario List widget configurator, the "Task" option in the "Action" tab was treating the default value "<None>" as "Current scenario"*
	DOC-846	*In the Scenario List widget, it was not possible to rename the "Name" column*
	DBPF-5646	*In the Scenario List widget, the scroll area was not resizing properly*
	DBPF-6368	*The Scenario list widget was displaying the column Type even if configured otherwise*
	DBPF-6320	*The scenario icon was switching several times between "cloud" and "lock" during job execution*

UI Data Grid/Explorer	DBPF-6391	The Data Explorer widget was fetching data multiple times
	DBPF-6340	The Data Explorer was not displaying issues of referenced scenarios
	DBPF-6010	In the Data Grid/Explorer widgets, the option "Show issues only" was not working properly when the filter was set to "Other"
	DOC-937	In the Data Grid/Explorer widgets, it was impossible to resize linked columns
	DOC-786	In the Data Grid/Explorer widgets, the "Blank/Not Blank" column filter was not properly working for relation fields
	DBPF-6693	In the Data Grid/Explorer widgets, editing duration fields was returning values different from the ones typed in
UI Filter	DOC-785	In the Filter widget, the date format was inconsistent
UI Gantt	DBPF-7008	In some cases, in the Gantt Chart widget, the option "Color By" was not working properly
	DBPF-7004	In the Gantt Chart widget, the selection was failing when using a custom renderer without "getDataInRange"
	DOC-741	The Gantt Chart widget was experiencing performance issues when using custom controllers "renderEvent()"
	DOC-675	When scrolling in the Gantt Chart widget, the whole view was also scrolling
UI Button	DOC-919	The default value of boolean inputs was not working properly

Bugfixes 4.2.1

DOC 4.2.1 introduces the following bugfixes compared to the unreleased version 4.2.0:

Data *Built-in Import/Export*	*DBPF-6490*	*Having duplicate entities in imported scenarios was throwing an error during relation resolution*
	DBPF-6454	*Importing a scenario with data pointing to missing entities of a referenced scenario was not creating issues*
Data *Data Integration Framework*	*DBPF-6554*	*When importing a corrupted scenario, the Data Integration API was throwing an NPE*
	DOC-830	*For entities using another entity as a primary key, updating its business key was breaking the method 'getFrom'*
UI *Tables*	*DBPF-6401*	*In some cases, importing a scenario was causing duplicated and/or phantom issues to be displayed*

CDM = Composite Data Model

Bugfixes 4.2.0

DOC 4.2.1 introduces the following bugfixes compared to version 4.1.0:

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*	*DOC 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, DOC 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	DOC was not displaying the message "There is too much data to display" when comparison was failing