Release Notes
DB Gene 4.3.0

September 20th, 2024

Note:

DB Gene 4.3.0 introduces several infrastructure updates and deprecations.
They are described in the DB Gene 4.3.0 Migration Guide, available on the DecisionBrain website.

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

Important Notes

*Please take into account the following important information when using the new version of **DB Gene 4.3.0**, released on September 27th, 2024.*

Updates

DB Gene 4.3.0 introduces the following infrastructure updates.

Spring Update

DB Gene and DBOS 4.3.0 now use Spring Boot 3.2.4 and Spring Cloud 2023.0.1. Formerly, it was version 3.0.12 and version 2022.0.3, respectively.

Gradle Update

DB Gene 4.3.0 now uses Gradle 7.6.4. Formerly, it was version 7.4.2.

Keycloak Update

DB Gene 4.3.0 now uses Keycloak 24.0.2. Formerly, it was version 23.0.4.

AG Grid Update

DB Gene 4.3.0 now uses AG Grid 31.2.0. Formerly, it was version 31.0.0.

Angular Update

DB Gene 4.3.0 now uses Angular 18.2.2. Formerly, it was version 17.2.3.

Apache Update

DB Gene 4.3.0 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

DB Gene 4.3.0 now uses RabbitMQ 3.13.0. Formerly, it was version 3.11.2.

NGINX Update

DB Gene and DBOS 4.3.0 now use NGINX 1.26. Formerly, it was version 1.22.

Deprecations

- A newer version has replaced the Pivot Table widget. The former has been renamed *Pivot Table (Legacy)* and will be removed in a future release.

- The Scenario Service GraphQL API was returning the `Path { scenarioReferenceGraph }` property. This property is incompatible with GraphQL structures and is now deprecated. It will be removed in DB Gene 4.4.0. To access this property, use the corresponding REST API instead.

- The following methods of the `data-service-base` library have been deprecated:

  - In `BatchCollectorService`, method `saveItems(String scenarioId, List<T> entityDTOs)` has been deprecated in favor of `saveEntities(String scenarioId, List<T> entityDTOs)`.

  - In `ScenarioUpdateService`, method `saveItems(String, List<DataServerEntityDTO> items, SchemaCheckersRunOptions)` has been deprecated in favor of `saveEntities(String, List<DataServerEntityDTO> entities, SchemaCheckersRunOptions)`.

- The following endpoints have been deprecated:

  - `/data/simple-excel-export` is now replaced by `/data/excel-export`.

  - `/data/simple-excel-import` is now replaced by `/data/excel-import`.

  The deprecated parameter `sortColumns` has also been removed from the Excel Export API.

- The old syntax of `code-replicate-plugin` DSL (Domain Specific Language) is now deprecated in favor of the new one. The content of the block `codeUpdates { }` should be migrated to `codeReplicas { }`.

- In method `ScenarioDataExpression`, the CSV and DBRF file formats have been marked as deprecated.

- Method `setCustomApplicationController(..)` from the `GeneApplicationService` has been deprecated and replaced by `registerApplicationController(..)`.

End-User Features

DB Gene 4.3.0 introduces new beta features. These features are still under development and will be completed with enhancements in the coming releases. It also implements a new exchange format for scenario data, a more robust Composite Data Model, enhanced tasks and permissions, new intuitive filtering functionalities, an improved application controller, and augmented widgets.

New Beta Features

DBPF-6791, **DBPF-6792**

For the first time, DB Gene introduces beta features. It provides users with **JupyterLab integration** and a **Code Editor Widget** that can also be used as a **Rule Script Editor and Runtime**.

To enable or disable the generation of beta features, the scaffolding now provides users with the `--experimentalFeatures[=yes|no]` flag.

Once generated, they can be enabled from the *Application Preferences* by setting the *EXPERIMENTAL_FEATURES* parameter to *true*.

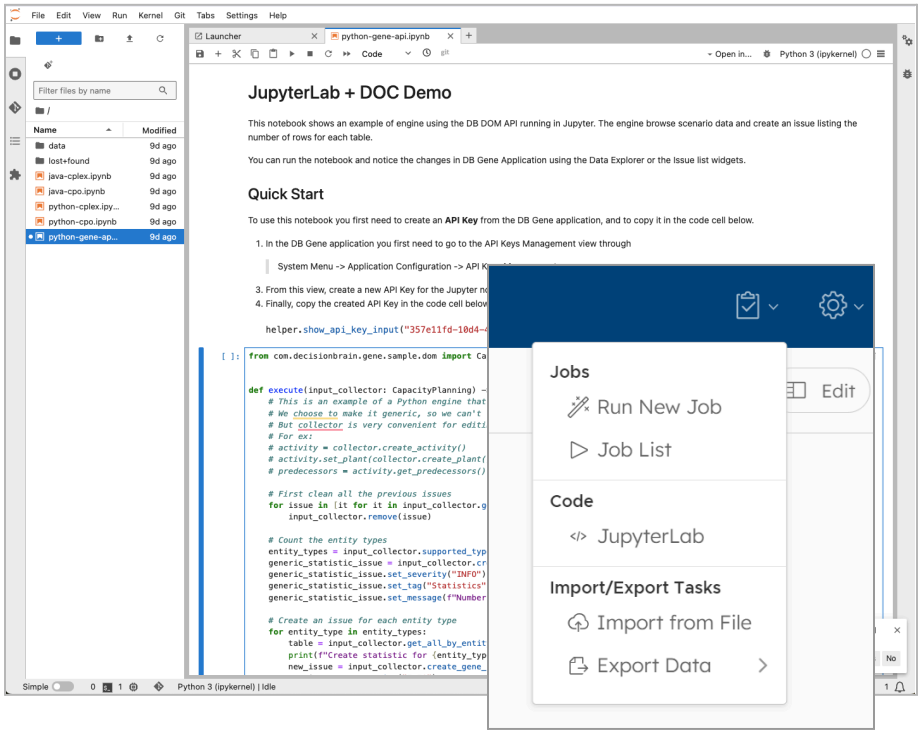
New JupyterLab Integration

DBPF-6767, **DBPF-6760**, **DBPF-6871**, **DBPF-6763**

The new integration of JupyterLab into the Platform 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



In addition, the Helm chart has been improved to allow deploying the JupyterLab service.

New Code Editor Widget

DBPF-6794, **DBPF-6796**

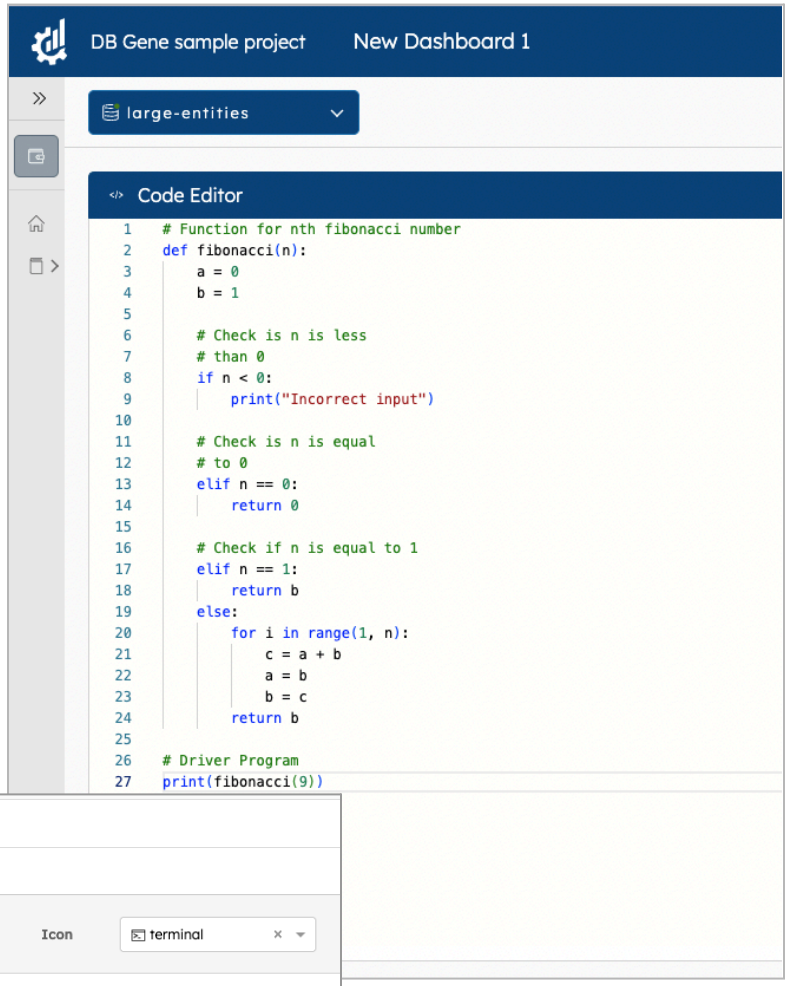
The new Code Editor widget is now available as a beta feature.

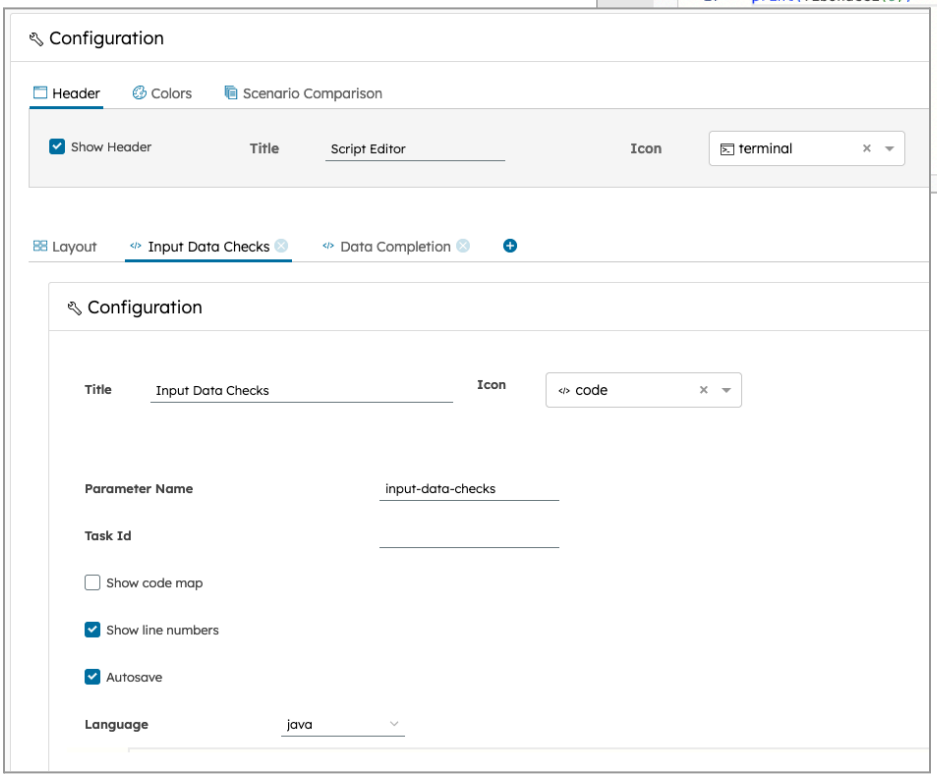
It allows a user to edit the content of a `GeneParameter` in the selected scenario.

The name of the parameter is defined in the widget configuration.

The row is selected on its `name` column, and its `value` column is used.

The widget can also be associated with a scripted task.





If so, the widget displays a *Run* button which, when clicked, invokes the task with two input parameters:

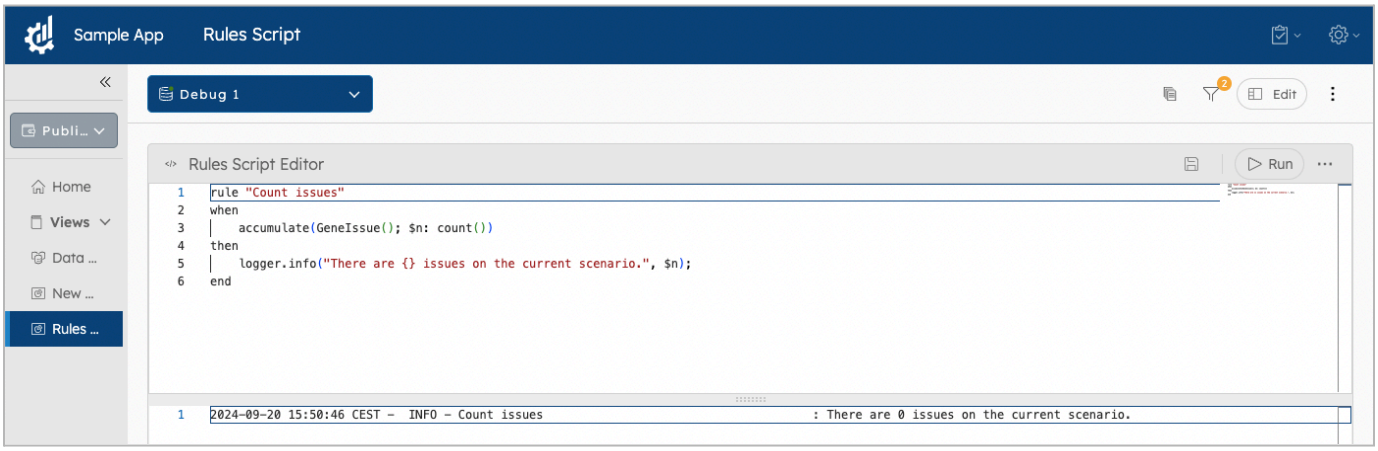
- one named *scenario*, which receives the ID of the current scenario, and
- one named *scriptName*, which receives the name of the `GeneParameter`.

New Rule Script Editor

DBPF-6797

A generic task and a generic routine now allow calling a rule engine on a scenario.

The added task is identified by the id `ExperimentalExecuteRulesetOnScenarioTask` and is compatible with the new Rule Script Editor widget.

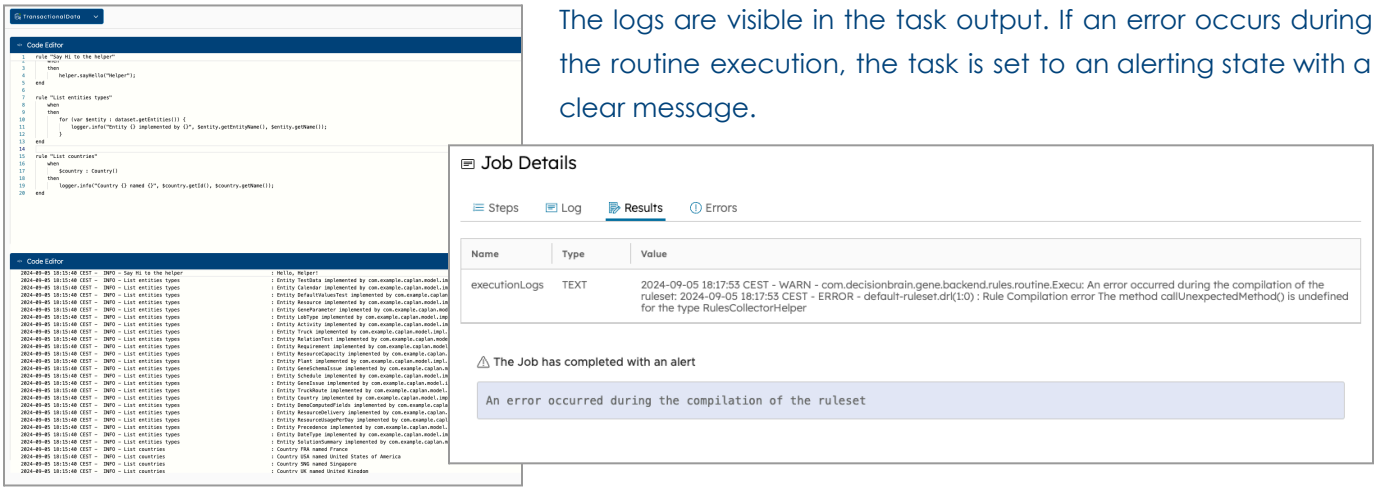


If the user configures the widget to launch this task when clicking the *Run* button, the task is run with the parameter name containing the script written in the Code Editor and the current scenario.

When launched, the task:

- calls the routine,
- retrieves the ruleset script from the parameter name as the input parameter,
- compiles the ruleset, and
- runs it on the current scenario.

The logs of the execution or failed compilation are stored in another parameter. For example, if the script is stored in a parameter named `checker-editor-script`, the logs of the last run will be stored in a parameter named `checker-editor-script-logs`.



The logs are visible in the task output. If an error occurs during the routine execution, the task is set to an alerting state with a clear message.

Deprecated and New Exchange Formats

DBPF-6701, **DBPF-6698**, **DBPF-6705**, **DBPF-6700**, **DBPF-6789**, **DBPF-6790**, **DBPF-6784**

Version 4.3.0 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 dataframes.
- Pandas dataframes can now be saved into XCSV.

Note: *These changes apply to simple and CDM applications.*

Improved Composite Data Model

DB Gene 4.3.0 makes the Composite Data Model (CDM) feature more robust by fixing various bugs, introducing the XCSV exchange format mentioned above, and through the following improvements.

Improved Scenario Checks at Startup

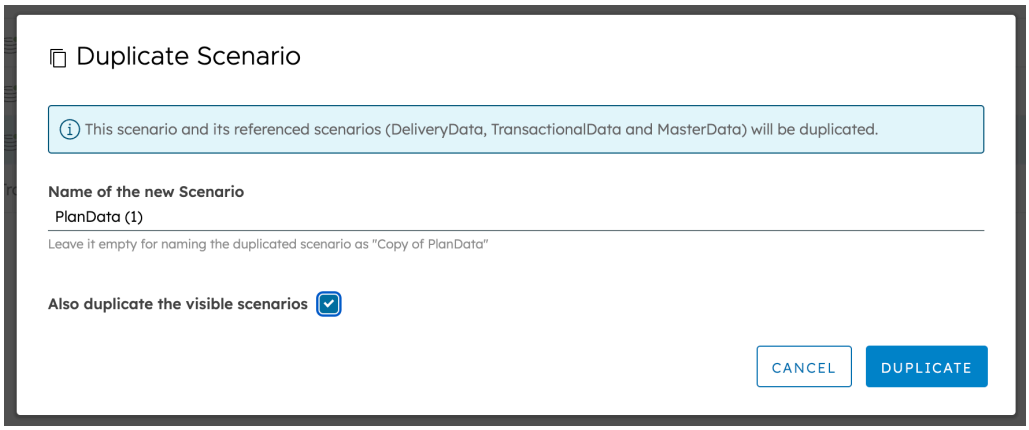
DBPF-6448

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

Improved Scenario Addition and Duplication

DBPF-6450, **DBPF-6463**

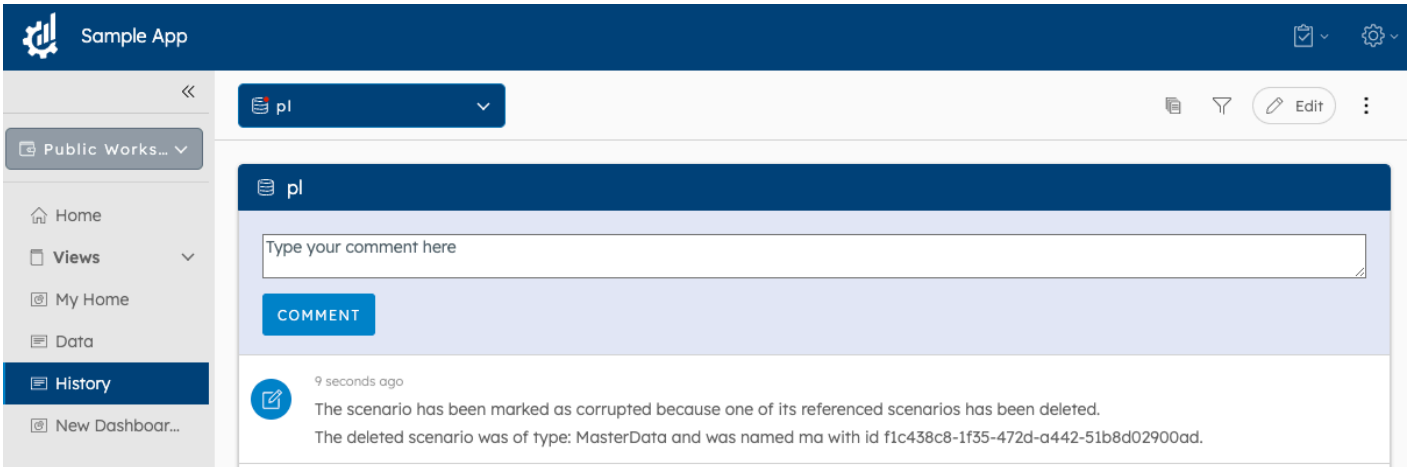
Users can now select a scenario link as a reference when adding a CDM scenario.
They can also duplicate a scenario and its referenced scenarios.



Improved Management for Corrupted Scenarios

DBPF-6564

It is now 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".



Improved Tasks

DBPF-6627, **DBPF-6634**, **DBPF-6658**, **DBPF-6242**

Scenario creation is now available in all scripted tasks and can be automated. In addition, scenario creation tasks now also allow setting scenario characteristics. Finally, the sample task *Create an empty scenario* has been improved for CDM applications.

Note: *As mentioned above, XCSV is now the default scenario data format in tasks.*

Improved Permissions

DBPF-6449, **DBPF-6444**

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.

Note:

- *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

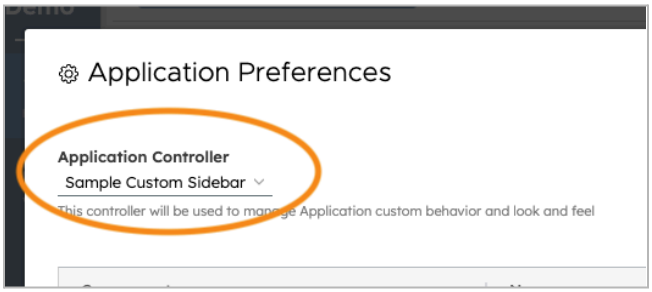
DBPF-6716

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 controller:

- 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.

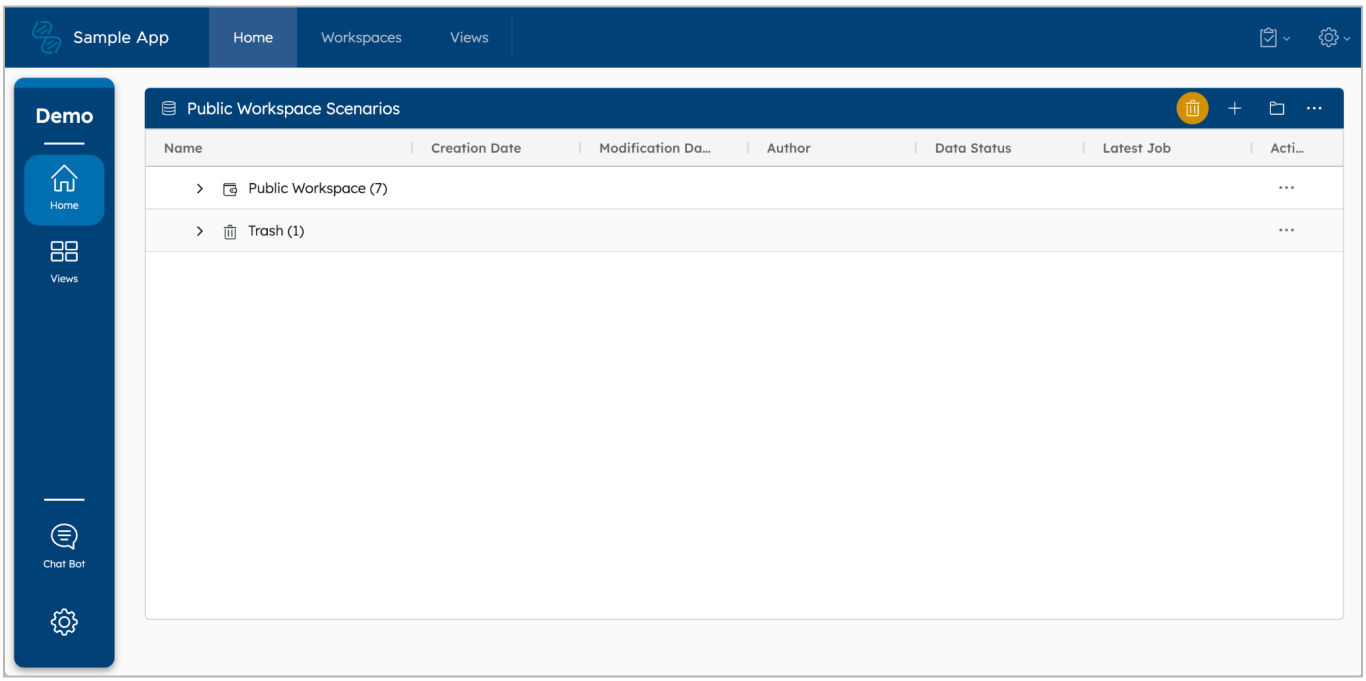


New Topbar and Sidenav Parameters in the Application Controller

The `GeneApplicationConfiguration` interface, which can be customized through the `GeneApplicationController` API, now has two new properties.

The new properties allow customizing the Topbar and the Sidenav components.

These two properties can be set with any Angular component that will replace the default provided ones.



```
export interface GeneApplicationConfiguration {

    header?: {
    // When provided, the component will replace the default
    // Gene component for the header bar
        component?: Type<any>;
       // ....
    },

    sidenav?: {
    // When provided, the component will replace the default
    // Gene component for the sidebar
        component?: Type<any>;
    }
```

The following example illustrates how to implement it.

```
export class SampleCustomHeaderAndSidebarApplicationController implements
GeneApplicationController {

    // Customize Default component for sidebar and header bar
     customizeDefaultConfiguration(context: GeneContext, view: ViewDashboard):
Observable<Partial<GeneApplicationConfiguration>> {
        return of({
                    sidenav: {
                        component: CustomSidebarComponent,
                        minWidth:120,
                        maxWidth:120,
                    },
                    header: {
                        component: CustomHeaderComponent,
                    }
        });
    }
}
```

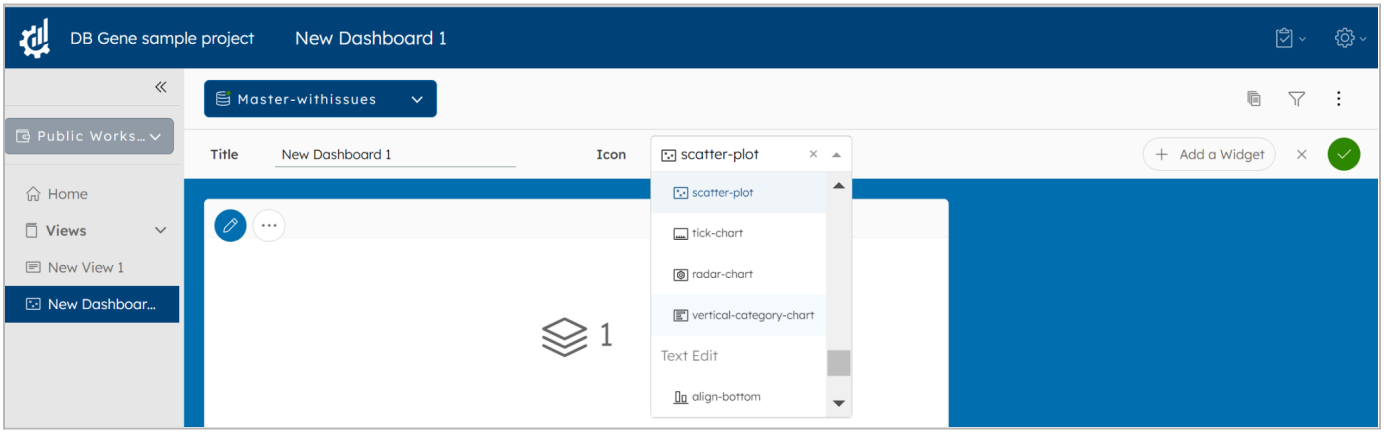
Improved Dashboards

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

Improved Sidenav Icons

DBPF-6971

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.

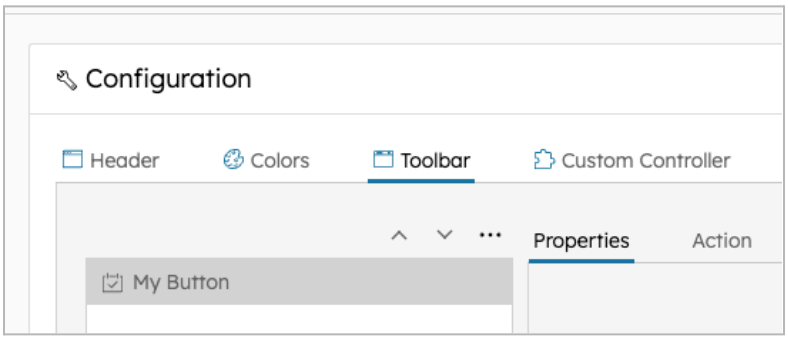


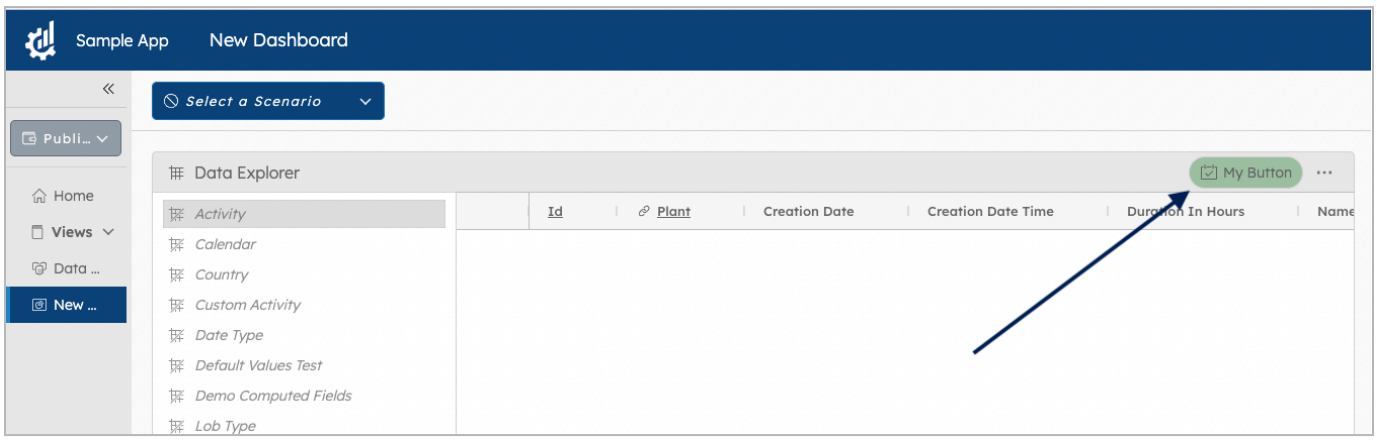
New Configurable Buttons for the Widget Toolbar

DBPF-6795

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.

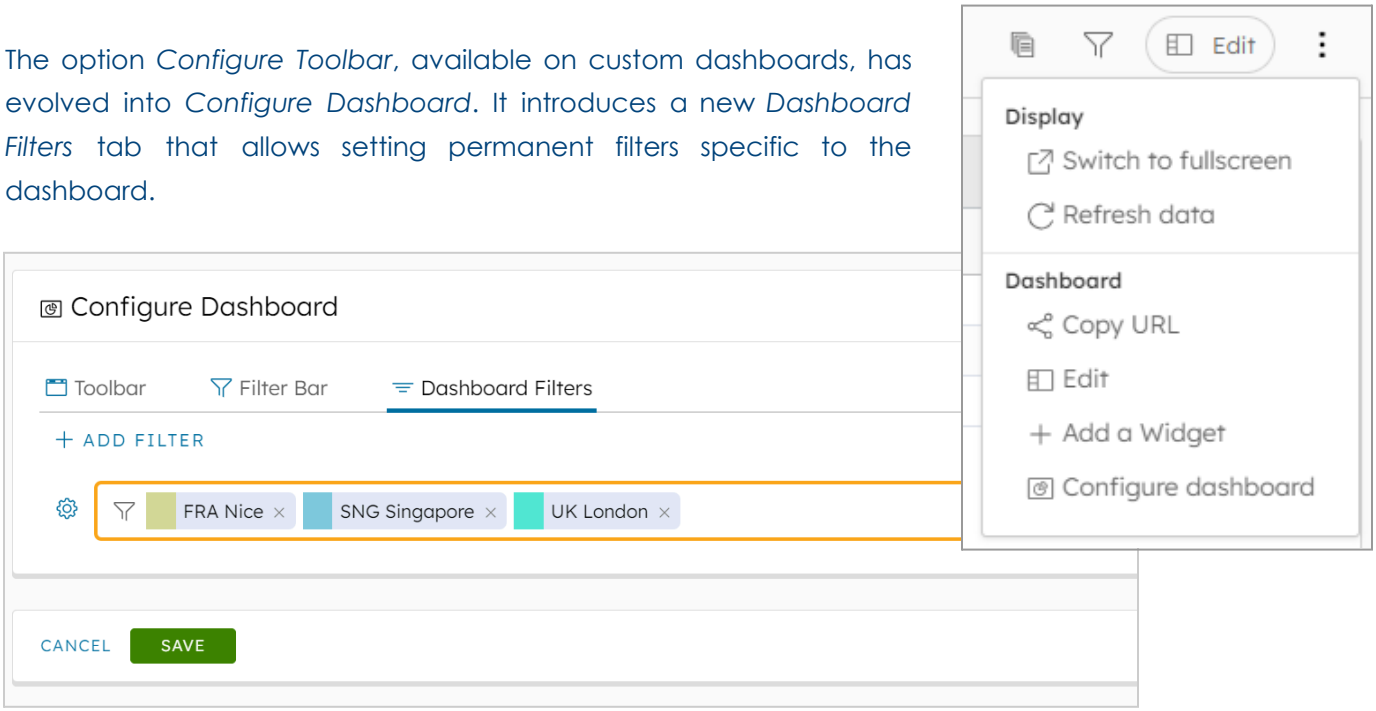




New Dashboard Filters

DBPF-6614

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.

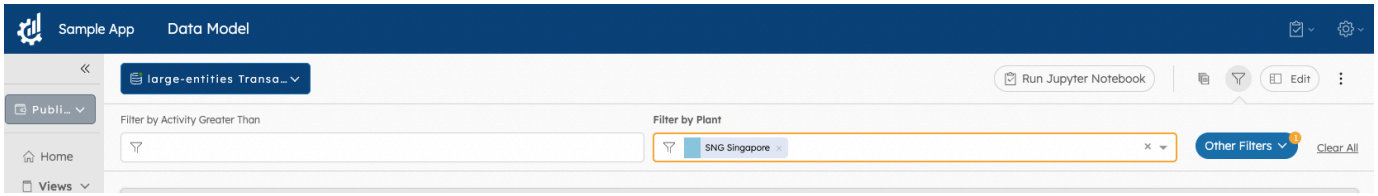


New Filter Bar

DBPF-6640, DBPF-6641, DBPF-6642

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

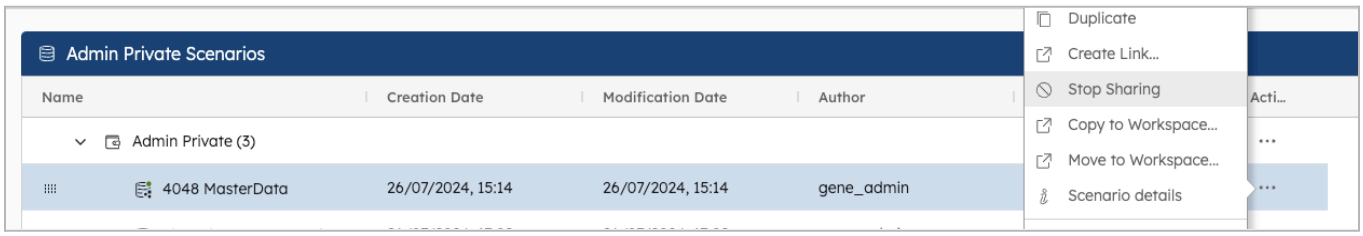
DBPF-6660

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.

Improved Scenario List Widget

DBPF-6447, DBPF-6451

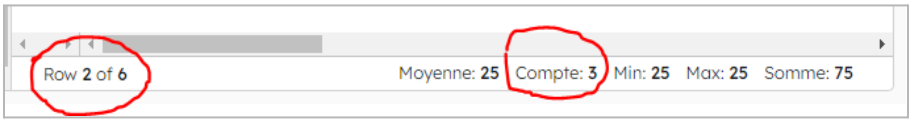
In the Scenario List widget, an icon now indicates when a scenario is shared with users, i.e. via a scenario link or when referenced by another scenario in a composite data model application. Also, in the Scenario List widget, a new option now allows to *Stop Sharing* a scenario.



Improved Data Grid/Explorer Widget

DBPF-6981, DOC-917, DOC-686, DBPF-6709

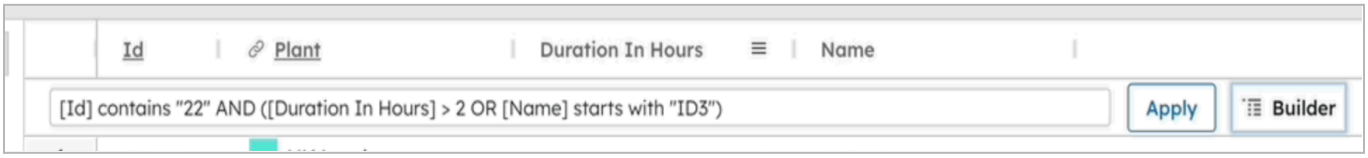
It is now possible to localize the index message at the bottom of the Data Grid widget.



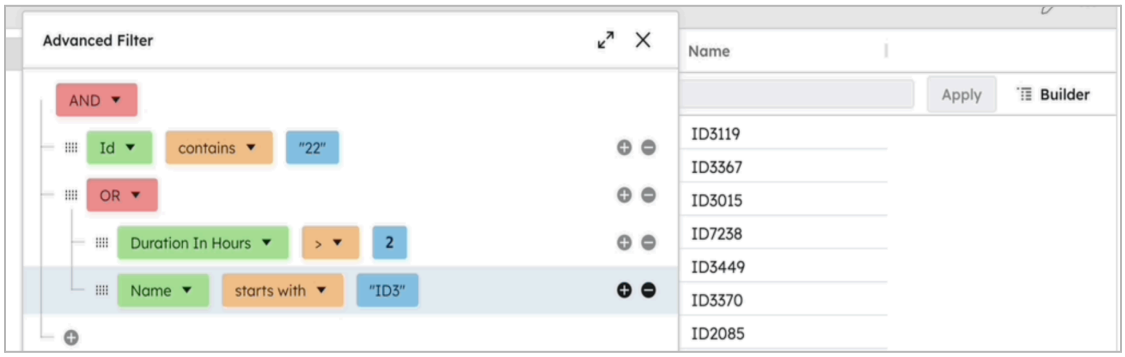
In the Data Grid/Explorer widgets, it is now possible to use the option *Delete all rows* when not in Edit mode and for entities from referenced scenarios.

Also, the option *Hide aggregation panel* is now enabled by default.

The most notable change in version 4.3.0 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.

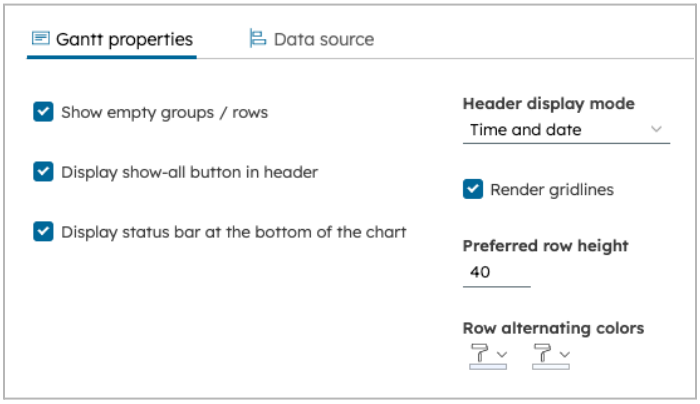


In addition, a *Builder* option helps users visualize the filter hierarchy.



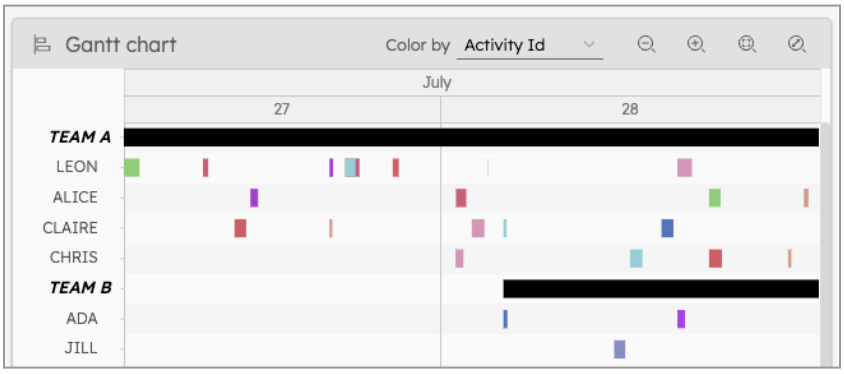
Improved Gantt Chart Widget

DOC-742, DOC-672, DBPF-6719, DBPF-6720, DBPF-6934



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.



- 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.
- It is now possible to set a renderer for the *Event* label using a custom controller.

Improved Map Widget

DOC-903

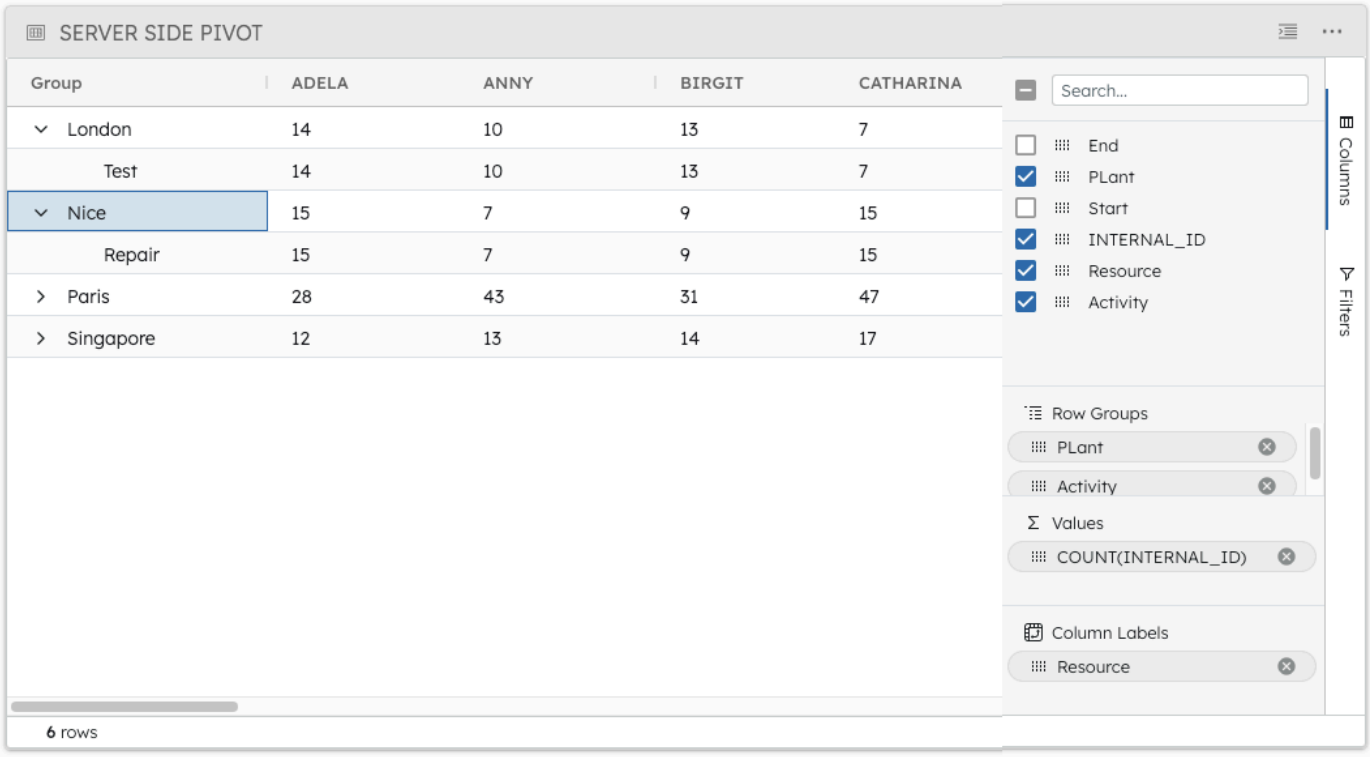
In the Map widget, the following custom controller method now allows overriding the marker series tooltip.


```
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>
            }
```

Improved Pivot Table Widget

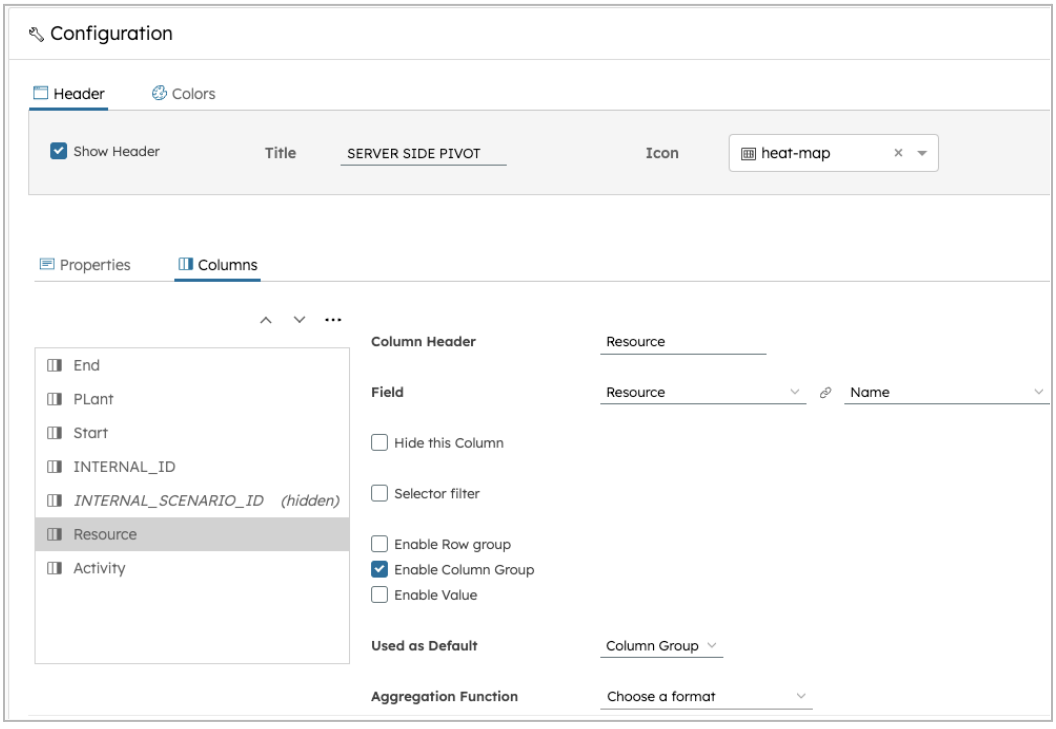
DOC-837

DB Gene 4.3.0 introduces a new version of the Pivot Table widget. The former version has been renamed *Pivot Table (Legacy)* and might be removed in a future release.



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

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



Technical Features

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

Improved Login Security

DBPF-5668

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

DBPF-6478

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

Improved CPLEX Integration

DBPF-6708

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

DBPF-5666

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

DB Gene 4.3.0 introduces several improvements and bugfixes listed in detail below.

Improvements

DB Gene 4.3.0 introduces the following improvements:

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 dataframes can now be saved into XCSV*
	DBPF-6790	*XCSV can now be loaded into Pandas dataframes*
	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	*The Platform now relies on AG Grid 31.2.0*
	DBPF-6385	*The Platform now relies on Angular 17.3.3 and AG Grid 31.2.0*
	DBPF-6384	*The Platform now relies on Angular 18.2.2*
	DBPF-6485	*The Platform now relies on Apache Commons Compress 1.26.1, Apache Commons Text 1.12.0, and Apache POI 5.2.5*
	DBPF-6570	*The Platform now relies on Gradle 7.6.4*
	DBPF-6381	*The Platform now relies on Keycloak 24.0.2*
	DBPF-6412	*The Platform now relies on NGINX 1.26.0*
	DBPF-6382	*The Platform 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	*The Platform 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 Build	DBPF-6871	A base Docker image for JupyterLab is now available for the Platform
	DBPF-6872	Gradle modules are now improved
	DOC-761	The dependencies in "package.json" are now sorted alphabetically
	DBPF-4735	The Platform 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 Filter Bar	DBPF-6640	A new Filter Bar is available for custom dashboards from the new option Configure Dashboard
	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

Bugfixes

DB Gene 4.3.0 introduces the following bugfixes:

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	DBPF-6391	The Data Explorer widget was fetching data multiple times

Data Grid/Explorer	*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*