DB Gene 4.7.0 Migration Guide

September 26th, 2025

DB Gene 4.7.0 Migration Guide

Preliminary notes **5**
Version 4.7.0 5
Version 4.6.2 5
Version 4.6.1 5
Application Changes **6**
Views & Dashboards **6**
Removed Deprecated Toolbar Controller 6
Data Changes **7**
Model **7**
Removed Deprecated JDL tags 7
Scenario Service **7**
Updated Scenario Service Notifications 8
Execution Service **8**
Removed Deprecated Features and APIs 8
Dev Changes **9**
REST API **9**
Updated Spring Rest API Clients 9
3rd-Party Components **10**
Updated Dependencies 10
Updated Angular Dependencies 11
Security **12**
Improved CVE Security 12
Improved DBOS Security 12
Improved Keycloak Password Policy 12
Improved Permissions-Policy Header 12
Improved Security Context in Kubernetes Deployments 12
Build **15**
Updated Java Dependency Name 15
New Collector Class Name Handling 15
Deployment **16**
Updated Security Context 16

Updated Docker Compose Template 16
Gene Online 16
Removed Tableau Widget 16
Improved Application Configuration Import/Export 16
Improved Data Model Rebuild 16
JupyterLab 17
Improved Idle Metric and Logging Configuration 17
New Voilà Integration in JupyterLab Image 17
Scripted Tasks Changes 18
Definition 18
Removed Method 18
Jobs 18
Removed Methods 18
UI Changes 19
General UI Changes 19
Removed Deprecated Method 19
Updated UI Style 19
Issue Widgets 20
Removed Deprecated Legacy Issue List Widget 20
Chart Widgets 20
New Scenario Series Display for the Radar Chart 20
Gantt Chart Widget 21
Deprecated Classes and Methods 21
Improved Multi-Series Custom Controller Support 21
New Multi-Series Selection 22
New Asynchronous and Progressive Model Building 22
New Rendering Options for Task Series 22
New Customizable Timescale Header 22
Image Widget 22
New Image Widget 22
Notebook Visualization Widget 23
New Notebook Visualization Widget 23
Pivot Table Widget 23
Removed Legacy Pivot Table Widget 23

Deprecated Features and APIs Scheduled for Removal in 4.8.0..... 24

- Each section is structured as follows: Removed, Deprecated, New, Improved, and Updated.
- *Dependency updates are listed in Section* ***Dev Changes*** *>* ***3rd-Party Components****.*
- *Future removals are listed in Section* ***Deprecated Features and APIs Scheduled for Removal in the Next Release****.*

For more details, please refer to the **DB Gene Documentation and Release Notes**.

Preliminary notes

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

Version 4.7.0

- **DOC-1349** If the configuration specific to the rules engine `RulesetRoutineConfiguration.java` is removed from a project, the backend service might fail to start.

Version 4.6.2

Version 4.6.1

- **DOC-1255** During scaffolding, the `CodeReplicasCheck` step was failing after `appUmbrellaUpdate`. **Fixed in 4.7.0.**
- **DOC-1340** Using a `processColumns()` endpoint in a `GeneTableController` can prevent editing cells, even in edit mode. Double-clicking does nothing.

Application Changes

This part lists all changes related to:

- the lifecycle, non-UI features, or APIs of views and dashboards, in Section Views & Dashboards.

Views & Dashboards

This section lists all changes related to the lifecycle, the non-UI features, or the APIs of views and dashboards.

Removed Deprecated Toolbar Controller

DBPF-8497

The interface `GeneWidgetCustomViewToolbarController`, which was deprecated, has now been removed.

Data Changes

This part lists all changes related to:

- concepts or the meta-model representation, in Section ***Model***.
- the Scenario Service API, in Section ***Scenario Service***.
- the Execution Service API, in Section ***Execution Service***.

Model

This section lists all changes related to concepts or the meta-model representation.

Removed Deprecated JDL tags

DBPF-8500

The following JDL tags, deprecated since 4.0.3-fp1, were removed:

```
// DOM [generated.jdl] : [true]
// DOM [java.className] : [Day]
// DOM [simple.primary.keys] : [id]
// DOM [default.attribute.value] : [FRA]
// DOM [SINGLE_ROW] : [true]
// DOM FIELD [plantId] - [foreign.key.reference] : [Plant id]
// DOM FIELD [plantId] - [java.referenceName] : [plant]
// DOM [java.isReferenceInverted] : [false]
// DOM [affects.primary.key] : [true]
```

They should no longer appear in your JDL definitions.

Scenario Service

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

Updated Scenario Service Notifications

DBPF-8584 (4.6.2)

Notifications from the Scenario Service are now sent asynchronously for improved performance. The Scenario Service synchronously notified all the activities to the UI.

Execution Service

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

Removed Deprecated Features and APIs

DBPF-8498

Some deprecated features and APIs were removed from the Execution Service.

Dev Changes

This part lists all changes related to:

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

REST API

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

Updated Spring Rest API Clients

DBPF-8643

In the client for the Data Service, the Scenario Service, and the Execution Service, the URL of the service can now be configured via one Spring property (`services.<service name>.base-url`) instead of two (`services.<service name>.host` and `services.<service name>.port`). The base URL takes precedence if all three properties are declared.

The Documentation has been updated to explain how to configure the Keycloak credentials. For more details, refer to Chapter Understanding the Data Service API.

3rd-Party Components

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

Updated Dependencies

ℹ️ **Note that, with the new version of Spring Cloud, the Spring Cloud Gateway has changed its properties namespace** to disambiguate the properties between `webmvc` and `webflux` implementations, as described in the two following links:

- Release v4.3.0 · spring-cloud/spring-cloud-gateway
- Migrates server-webflux properties to new namespace by spencergibb · Pull Request #3793 · spring-cloud/spring-cloud-gateway

=> You have to migrate all the properties starting with `spring.cloud.gateway` to `spring.cloud.gateway.server.webflux`.

- **DBPF-8443** Spring Boot updated from 3.4.5 to 3.5.5
- **DBPF-8443** Spring Cloud updated from 2024.0.1 to 2025.0.0
- **DBPF-8443** Spring Context updated from 6.2.6 to 6.2.10
- **DBPF-8443** Netty updated from 4.1.119 to 4.1.124.Final
- **DBPF-8443** graphql-java-extended-scalars updated from 22.0 to 24.0
- **DBPF-8443** Mongock updated from 5.5.0 to 5.5.1
- **DBPF-8443** Bouncycastle updated from 1.78.1 to 1.80
- **DBPF-8443** Apache commons-compress updated from 1.27.1 to 1.28.0
- **DBPF-8443** Apache commons-io updated from 2.18.0 to 2.20.0
- **DBPF-8443** Jackson updated from 2.18.3 to 2.19.2
- **DBPF-8450** AG Grid updated from 33.3.0 to 34.1.0
- **DBPF-8444** Keycloak updated from 26.2.4 to 26.3.1
- **DBPF-8445** MongoDB updated from 8.0.9 to 8.0.12
- **DBPF-8446** PostgreSQL updated from 15.13 to 15.14
- **DBPF-8447** RabbitMQ updated from 4.1.0 to 4.1.3
- **DBPF-8396** yeoman generator updated from 4.0.1 to 7.5.1
- **DBPF-8396** yo generator updated from 4.3.0 to 5.1.0
- **DBPF-8396** gulp-rename generator updated from 1.2.2 to 2.1.0
- **DBPF-8396** ngx-quill updated from 27.0.2 to 28.0.1
- **DBPF-8641** OpenApi generator updated from 7.12.0 to 7.14.0 (4.6.2)

Updated Angular Dependencies

DBPF-8448

Angular has been upgraded to 20.1.7 and NodeJS to 22.18.0.

None

```
@angular/animations:                    from 19.2.11 to 20.1.7
@angular/cdk:                           from 19.2.11 to 20.2.0
@angular/common:                        from 19.2.11 to 20.1.7
@angular/compiler:                      from 19.2.11 to 20.1.7
@angular/core:                          from 19.2.11 to 20.1.7
@angular/forms:                         from 19.2.11 to 20.1.7
@angular/localize:                      from 19.2.11 to 20.1.7
@angular/platform-browser:              from 19.2.11 to 20.1.7
@angular/platform-browser-dynamic:      from 19.2.11 to 20.1.7
@angular/router:                        from 19.2.11 to 20.1.7
@angular-devkit/build-angular:          from 19.2.11 to 20.1.6
@angular/cli:                           from 19.2.11 to 20.1.6
@angular/compiler-cli:                  from 19.2.11 to 20.1.7
@angular/language-service:              from 19.2.11 to 20.1.7
@danielmoncada/angular-datetime-picker: from 19.1.1  to 20.0.0
ngx-quill:                              from 27.0.2  to 28.0.1
typescript:                             from 5.5.4   to 5.8.3
```

ℹ️ **After the migration, it is recommended to execute the following command inside the web folder:**

None

```
ng generate @angular/core:control-flow
```

This command will replace structural directives (**`NgIf`**, **`NgSwitch`**, **`NgFor`**) with the built-in control flow (**`@if`**, **`@switch`**, **`@for`**) across all templates in the project. For more information, please refer to **Announcing Angular v20**.

Security

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

Improved CVE Security

DBPF-8493, DBPF-8496

The Gene Generator `base` image and `yeoman-generator` have been updated to address CVEs reported by Trivy.

Improved DBOS Security

DBPF-8469

DBOS now invalidates the JWT token on password change or if it was revoked by an admin, by integrating a session cleaning mechanism.

Improved Keycloak Password Policy

DBPF-7366

Keycloak password policy has been enhanced for password reuse.

Improved Permissions-Policy Header

DBPF-7224

The `Permissions-Policy` header has been added to the documentation and the DBOS web console to fix issues reported by Zap.

Improved Security Context in Kubernetes Deployments

DBPF-7364

The security context has been hardened for running containers with a user ID > 10k (the user ID `10001` is used for every container we deploy with the app umbrella Helm chart).

The security context has also been hardened to run containers with a read-only file system.
By default, the `/tmp`, `/var/log`, `/var/tmp`, and `/app` folders have write permissions. The Java micro services are run with the current directory set to `/app` to ease the creation of files with a code like new `File()` that does not specify where the file should be created. In the case of the migration of an application that has to keep the data in persistent volumes, the project team has to ask the IT team to change the ownership of the files in these volumes to the user ID `10001` and eventually to the group ID `10001`.

The following script can achieve the ownership change of the files in the databases:

```
None
#!/bin/sh
# Please set your kubectl context to the appropriate cluster with the following command:
#export KUBECONFIG=<PUT YOUR KUBECONFIG FILE HERE>

echo "Ensure the namespace is the one for wich you want to migrate the data ownership:"
kubectl config view --minify --output 'jsonpath={..namespace}'
echo ""
read -p "Continue [y|N]? " -n 1 -r
echo
if [ "$REPLY" = "y" ] || [ "$REPLY" = "Y" ]
then
   echo 'Migrating data ownership...'
   kubectl scale --replicas 0 deployment mongo postgres
   kubectl apply -f - <<EOF
apiVersion: batch/v1
kind: Job
metadata:
   name: migrate-mongo-and-postgres-data-ownership
spec:
   ttlSecondsAfterFinished: 3600
   template:
       spec:
           restartPolicy: Never
           volumes:
               - name: mongo-data-volume
                 persistentVolumeClaim:
                   claimName: mongo-data
               - name: postgres-data-volume
                 persistentVolumeClaim:
                   claimName: postgres-data
           containers:
               - name: busybox-pod
                 image: "k8s.gcr.io/busybox"
                 command: ["/bin/sh", "-c", "echo 'Starting change ownership ...' &&
chown -R 10001:10001 /mnt/mongo-data && chown -R 10001:10001 /mnt/postgres-data && echo
'Ownership change completed!!!'"]
                 volumeMounts:
                 - name: mongo-data-volume
                   mountPath: /mnt/mongo-data
                 - name: postgres-data-volume
                   mountPath: /mnt/postgres-data
EOF
   kubectl wait --for=condition=complete job/migrate-mongo-and-postgres-data-ownership
   kubectl scale --replicas 1 deployment mongo postgres
else
   echo 'Stopping script, bye!'
fi
```

Otherwise, if you have issues with changing the ownership of the data volumes or if there are issues with custom code that are not able to write files on the file system, you can take some inspiration from the following values for implementing a workaround in your project.

None

```
###-------------------------------------------------------------------------------------------
-----------------------------
### This values file tend to limit frictions during projects migrations
### It should not be used on new project, and only if you have issues with the security
context after
### a platform migration.
###-------------------------------------------------------------------------------------------
-----------------------------
### If you have issues with the hardened security-context that you did succeed to
overcome,
### You can globally disable it with the following property:
#global:
#  feature:
#    securityContext:
#      enabled: false
### Otherwise for easing the migration, you can run the container of the infra's  layer
with the previous user id.
dbos-infra:
# Run Keycloak without the read-only file-system
#  keycloak:
#    securityContext:
#      container:
#        readOnlyRootFilesystem: false
 mongo:
   securityContext:
#     Run with the previous userid (avoid data file ownership issues)
     pod:
       fsGroup: 999
       runAsGroup: 999
       runAsUser: 999
#     Run without the read-only file-system
     container:
       readOnlyRootFilesystem: false
 postgres:
   securityContext:
#     Run with the previous userid (avoid data file ownership issues)
     pod:
       fsGroup: 26
       runAsGroup: 26
       runAsUser: 26
#     Run without the read-only file-system
     container:
       readOnlyRootFilesystem: false
 rabbitmq:
   securityContext:
```

```
#     Run with the previous userid (avoid data file ownership issues)
     pod:
       fsGroup: 1000
       runAsGroup: 1000
       runAsUser: 1000
#     Run without the read-only file-system
     container:
       readOnlyRootFilesystem: false
### If you have some issues with file creation in you custom code,
### You can disable the read-only file-system for all the containers of the platform's
layer
<%= projectName %>-app:
 defaultSecurityContext:
   container:
     readOnlyRootFilesystem: false
```

Build

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

Updated Java Dependency Name

DBPF-8600

The library `com.decisionbrain.gene:security-service-client` has been renamed to `com.decisionbrain.gene:security-service-client-gene-identity`. In addition, the service that handles the gene identity must be enabled, if necessary, with the `@EnableClientGeneIdentity` on one of your Spring configuration `class.dat`.

New Collector Class Name Handling

DBPF-8156

The collector class name from the JDL can now be used automatically.

Deployment

This section lists all changes related to DOC integration to Docker and Helm.

Updated Security Context

DBPF-7364

Ensure security context specifies a userid > 10k and add `seccomp` profile.

Updated Docker Compose Template

DBPF-8336 (4.6.1)

The Docker Compose template now exposes missing variables for the gateway service.

Gene Online

This section lists all changes related to Gene Online.

Removed Tableau Widget

DBPF-8786

The Tableau widget is no longer available in Gene Online.

Improved Application Configuration Import/Export

DBPF-7642, DBPF-7611

In Gene Online, the application configuration import and export now include the data model.

Improved Data Model Rebuild

DBPF-8729

Gene Online now preserves scenario data when the data model changes.

JupyterLab

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

Improved Idle Metric and Logging Configuration

DOC-1112, **DBPF-8633**

JupyterLab idle metric now reports a more accurate value.

Additionally, JupyterLab's default virtual environment now implements a new logging configuration for the default Python logging module.

- The default console log level is set to DEBUG

A 1MB rotating log file, `logs.txt`, is stored in the Notebook folder. It uses the `DEBUG` level and includes the `Jupyter Kernel ID` and current `Thread ID` in all entries.

When a Notebook runs through the Voilà server, the default log level is `INFO` to avoid cluttering the visualization.

New Voilà Integration in JupyterLab Image

DBPF-8630

The Voilà server is a powerful tool that enables users to convert Jupyter notebooks into standalone web applications. One of its key features is the ability to render the visualizations and outputs from notebooks while hiding the underlying code. This means that when a notebook is run through Voilà, users can interact with the visualizations—such as plots, charts, and other graphical outputs—without seeing the code that generated them.

This capability is particularly beneficial for sharing results with non-technical stakeholders or for creating polished presentations of data analysis. By focusing solely on the visual elements, Voilà allows users to present their findings in a clean and user-friendly manner, enhancing the overall experience for viewers who may not be familiar with coding or the intricacies of the notebook's content.

Scripted Tasks Changes

This part lists all changes related to:

- the task definition and script language, in Section ***Definition***.
- job triggering, execution, and monitoring, in Section ***Jobs***.

Definition

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

Removed Method

DBPF-8498

The method `ExecuteRoutineStatement.onBackendService` has been removed.

Jobs

This section lists all changes related to job triggering, execution, and monitoring.

Removed Methods

DBPF-8498

The following methods have been removed:

- `JobInstanceService.eraseJob`
- `GeneTask.getPersistSystemLog`
- `GeneTask.setPersistSystemLog`

UI Changes

This part lists all changes related to:

- *all widgets, regardless of their type, or non-widget elements of the UI, such as the sidebar, menus, scenario picker, or views and dashboards, except for Access Control elements, such as permissions and API keys, in Section **General UI Changes**.*
- *specific UI widgets, in Section **Issue**, **Chart**, **Gantt Chart**, or **Image** widgets — if a change impacts more than one widget, it is listed in the most relevant section and referred to in the others.*

General UI Changes

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

Removed Deprecated Method

DBPF-8499

Removed deprecated method `getEntities()` in `gene-data-api.service.ts`. Use `getAllEntities(params)` instead.

Updated UI Style

DBPF-6457

Clarity packages have been upgraded, and the UI has been reworked to fit the new Clarity style.

Issue Widgets

This section lists all changes related to the Issue List and Issue Details widgets.

Removed Deprecated Legacy Issue List Widget

DBPF-6690

The deprecated legacy Issue List widget has been removed. It should be replaced with the new Issue List widget on the impacted dashboards.

Chart Widgets

This section lists all changes related to the Chart widget.

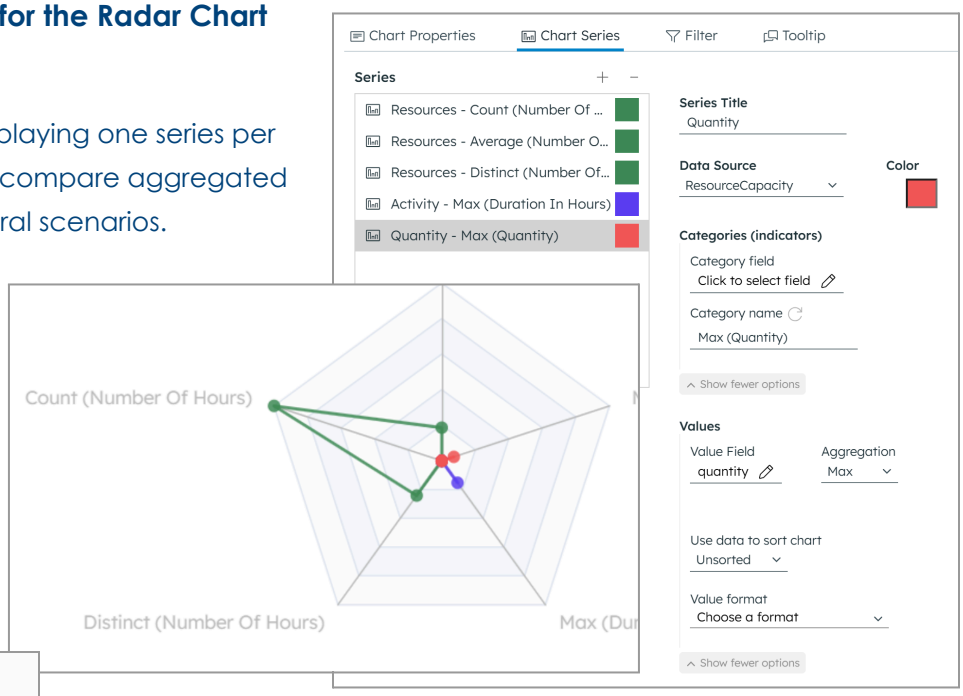
New Scenario Series Display for the Radar Chart

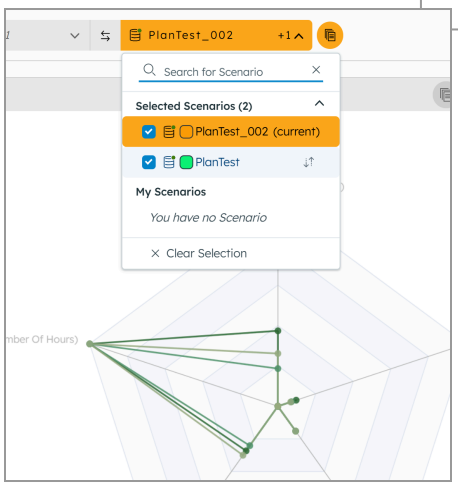
DOC-622

The Radar Chart now allows displaying one series per scenario without a category to compare aggregated entity values across one or several scenarios.

To do so, add a Chart Series for each entity to aggregate, set as the Data Source, and leave the Category Field to None.

The radar will display each aggregation as a single point.





Series with identical Series Title share the same color, and each label can be renamed via the Category Name option. By default, the label displays the Aggregation and Value Field.

It is then possible to use Scenario Comparison, where scenarios are represented in the chart with different colors.

Gantt Chart Widget

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

Deprecated Classes and Methods

DBPF-6696, DOC-1245, DOC-980

In the Gantt Chart widget:

- The classes `BaseGanttDataSource`, `DefaultGanttDataSource`, and `DynamicGanttDataSource` are now deprecated and no longer used by the framework. They have been replaced by `MultiSeriesGanttDataSource`, but kept for compatibility with existing customizations. The code related to loading the *TASK* series has been moved to `GanttTaskLoader` and `GanttDynamicTaskLoader`.
- In `BaseGanttDataSource`, the `buildEvents` method is deprecated. Instead, use `DbGanttBuilder.buildEvents`.
- In `DefaultGanttDataSource`, the `retrieveEvents` method is deprecated. Instead, use `DefaultGanttDataSource.retrieveEventsAndBuildModels`.
- In `DbGanttBuilder`, the `toggleGroup`, `getGroupStates`, and `setGroupStates` methods are deprecated. The responsibility has been moved to `MultiSeriesGanttDataSource`.
- In `DbGanttBuilder`, the `getResourceRequestedFields`, `getResourceIdFields`, and `getAdditionalGroupingFields` methods are deprecated. Instead, use the methods from `gantt-builder-utils`.
- In `GanttController`, since multiple TASK series are possible, the `providesMainRenderer` method will no longer be called. Instead, `getRenderer` will be called for each configured series.
- In `GanttController`, the `processEvents` and `processResources` methods are remnants of the legacy Gantt component. Instead, `postProcessChartModel` should be implemented.

Improved Multi-Series Custom Controller Support

DOC-1310

If custom loading is implemented, the Gantt Chart custom controller has better multi-series support and no longer requires a series configuration.

New Multi-Series Selection

DOC-1280, DOC-1311

In the Gantt Chart widget, mouse selection now supports multiple TASK series.

New Asynchronous and Progressive Model Building

DBPF-6696

The Gantt Chart widget now supports asynchronous and progressive model building.

New Rendering Options for Task Series

DBPF-8691

In the Gantt Chart widget, the TASK series has new rendering options.

New Customizable Timescale Header

DOC-1245, DOC-980

The Gantt Chart custom controller now allows customizing the timescale header.

Image Widget

This section lists all changes related to the Image widget.

New Image Widget

DBPF-8135 (4.6.1)

The new Image widget now allows displaying an image stored in a scenario. The entity must be stored as a BINARY or TEXT field containing base64-encoded data.

The option Display Type allows configuring the image to Cover the whole widget, Fit inside the widget, or Scale with the size of the widget.

If the query returns multiple images, users can click on the left and right arrows that appear on the image to browse them in carousel mode.

The Image Widget also displays a dynamic caption that can mix static text, icons, and entity values.

Notebook Visualization Widget

New Notebook Visualization Widget

DBPF-8633

JupyterLab allows for powerful custom visualizations for analysts and developers who work directly with code. The new Notebook Visualization widget can now display these visualizations in a code-free format that relies on the Voilà server, a tool that converts Jupyter notebooks into standalone web visualizations.

You can configure this widget to refresh when the scenario selection or data changes. The current scenario is provided to the Notebook, accessible in Python via the following helper method:

```
None
from dbgene.jupyter import GeneHelper as helper

scenario = helper.get_context_scenario(CapacityPlanning)
```

A sample Notebook visualization, gene_visualization.ipynb, demonstrates this API's usage.

Pivot Table Widget

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

Removed Legacy Pivot Table Widget

DBPF-7562

Version 4.7.0 no longer supports the deprecated legacy Pivot Table widget, and its import is not scaffolded by default.

To continue using the Pivot Table functionality, last documented in 4.3.0, please switch to the new Pivot Table widget, which is available since version 4.5.0.

Edit the impacted dashboards by removing the old widget and configuring the new one. While the new Pivot Table offers improved capabilities, some features may differ from the legacy widget. We recommend reviewing your dashboards and testing your use cases to ensure a smooth transition.

Deprecated Features and APIs Scheduled for Removal in 4.8.0

The following deprecated elements will no longer be supported from version 4.8.0.

DBPF-8494

This section lists all removals that are planned to come in 4.8.0:

- The Task API and master REST API (bucket/file API) from DBOS.
- The pagination feature in the Data Grid/Explorer.
- The property `fromSelection` from `GeneEntityFilter`.
- The Gantt Chart methods deprecated in 4.7.0.
- The integration of Tableau dashboards that display scenario data.
- `DbrfImportTask` in built-in tasks. Use `ScenarioImportTask` instead.
- `getPressedKeyCodes()` in `web/web-frontend-base/projects/gene/common-widget/src/lib/gene-table/model/gene-table-api.model.ts`. Use `{@see getLastUserInteractionEvent}` instead when you need to know if modifier keys are pressed.
- `readOnlyState`, `comparisonState`, and `editionState` in `web/web-frontend-base/projects/gene/common-widget/src/lib/gene-table/model/gene-table-state.ts`.
- `com.decisionbrain.gene.execution.model.script.StringExpression#idOfNewScenarioWithMetadata(com.decisionbrain.gene.execution.model.script.Expression, com.decisionbrain.gene.execution.model.script.Expression, com.decisionbrain.gene.execution.model.script.Expression, com.decisionbrain.gene.execution.model.script.Expression)`. Instead, use `{@link #idOfNewScenarioWithMetadata(Expression, Expression, Expression)}`.
- `idOfNewScenarioWithFolder(Expression scenarioName, Expression workspaceId, Expression folderId)`. Instead, use `{@link #idOfNewScenario(Expression, Expression)}`.
- `idOfNewScenarioWithMetadata(Expression file, Expression scenarioName, Expression workspaceId, Expression folderId, Expression lockIt)`. Instead, use `{@link #idOfNewScenario(Expression, Expression, Expression, Expression)}`.