IBM DOC 4.4.0.4.6.0 Migration Guide

June 23rd, 2025

IBM DOC 4.4.0.4.6.0 Migration Guide

- 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 **IBM DOC Documentation and Release Notes**.

Application Changes

This part lists all changes related to:

- *the application configuration file, in Section **Application Configuration**.*
- *the lifecycle, non-UI features, or APIs of views and dashboards, in Section **Views & Dashboards**.*

Application Configuration

This section lists all changes related to the application configuration file.

New Anti-Duplicate Migration Script

> ℹ️ **Note on `app-config.json`:**
>
> Please be aware that the `app-config.json` file **may contain duplicate `widgetId` values** across different widgets. This can lead to unintended behavior, including:
>
> - **Incorrect widget positioning** on dashboards, as the layout logic may not correctly distinguish between widgets with the same ID.
> - **Shared or conflicting widget state**, where multiple widgets unintentionally reference the same state due to identical `widgetId`s.
>
> To avoid these issues, ensure that all widgets within `app-config.json` have **unique `widgetId` values** before proceeding with the migration.

To ensure all widgets in your `app-config.json` have unique `widgetId` values, use the `app-config` migration script included just after this section.

This is especially important to prevent issues such as incorrect widget positioning or shared state between widgets.

Run the script as follows:

```
None
node fix-app-config-duplicate-widget-ids.mjs --input = path/to/app-config.json
```

You can include the `--dry-run` flag to preview changes without modifying the file, and `--verbose` for detailed logging.

Optionally, use `--output=path/to/output.json` to write the result to a different file.

The script scans all custom dashboards and views, detects duplicate or missing `widgetId` occurrences, and fixes them automatically.

```
JavaScript
import {readFileSync, writeFileSync} from 'node:fs';
import {basename} from "node:path";

////////////////////////////////
//            CLI             //
////////////////////////////////

// Command line arguments
const args = process.argv.slice(2);

// display help and exit
if(args.includes('-h') || args.includes('--help')) {
    showHelp();
    process.exit();
}

// parse command line options
const dryRun = args.includes('--dry-run') || args.includes('-d');
const verbose = args.includes('--verbose') || args.includes('-v');
const inFile = args.find(arg => arg.startsWith('--input=') ||
arg.startsWith('-i='))?.split('=')?.[1];
const outFile = args.find(arg => arg.startsWith('--output=') ||
arg.startsWith('-o='))?.split('=')?.[1] ?? inFile;

if(!inFile) {
    console.error('No input file specified. Use --input=<file> or -i=<file>');
    showHelp();
    process.exit(1);
}
```

```
////////////////////////////////
//            MAIN            //
////////////////////////////////

const appConfig = JSON.parse(readFileSync(inFile, { encoding: 'utf8' }));

let dashboardsCount = 0;
let viewsCount = 0;
let widgetCount = 0;
let widgetIds = new Map();
let duplicates = 0;
let missingWidgetConfiguration = 0;
let uuid = 0;

for(const _ of appConfig.children) {
     for(const pathNode of _.children) {
         // children[0].children[4].path.properties[0].key
         for(const { value } of pathNode.path.properties.filter(p => p.key ===
'custom_dashboard_configuration')) {
            dashboardsCount++;
            // children[0].children[4].path.properties[0].value.items[0].manifest
            for(const widget of value.items) {
                widget.widgetConfiguration = fixWidgetId(widget.widgetConfiguration);
            }
         }
         for(const { value } of pathNode.path.properties.filter(p => p.key ===
'custom_view_configuration')) {
             viewsCount++;
             // children[0].children[3].path.properties[0].value.manifest
             value.widgetConfiguration = fixWidgetId(value.widgetConfiguration);
         }
     }
}

verbose && console.log(`Found ${dashboardsCount} custom dashboards entries`);
verbose && console.log(`Found ${viewsCount} custom views entries`);
verbose && console.log(`Found ${widgetCount} unique widgetIds`);
verbose && console.log(`Found ${widgetIds.size} widgets entries`);
console.log(`Fixed ${missingWidgetConfiguration} widgets missing a configuration`);
console.log(`Fixed ${duplicates} duplicate widgetIds`);

!dryRun && writeFileSync(outFile, JSON.stringify(appConfig, null, 2), { encoding: 'utf8'
});

////////////////////////////////
```

```
//            UTILS            //
/////////////////////////////////

function fixWidgetId(widgetConfig) {
    if(!widgetConfig) {
        missingWidgetConfiguration++;
        return {
            widgetId: `widget-${uuid++}`,
        };
    }
    if(widgetIds.has(widgetConfig.widgetId)) {
        duplicates++;
        widgetConfig.widgetId += `-${uuid++}`;
    }
    else if(!widgetConfig.widgetId) {
        console.warn(`Found widget without widgetId in widget configuration`);
        widgetConfig.widgetId = uuid++;
    }
    else {
        widgetCount++;
    }
    widgetIds.set(widgetConfig.widgetId, widgetConfig);
    return widgetConfig;
}

/**
 * Show help message
 */
function showHelp() {
    console.info(`
Fix an app-config file by ensuring all widgets have a unique widgetId.

Usage: node ${basename(process.argv[1])} --input=<file> [options]
Options:
  --help, -h                Show this help
  --verbose, -v             Verbose mode
  --dry-run, -d             Do not edit any file
  --input=<file>, -i=<file>  Input file to read the app-config from
  --output=<file>, -o=<file> Output file to write the fixed app-config to (default: same
as input file)
`);
}
```

Views & Dashboards

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

Removed Custom Views

DBPF-8086

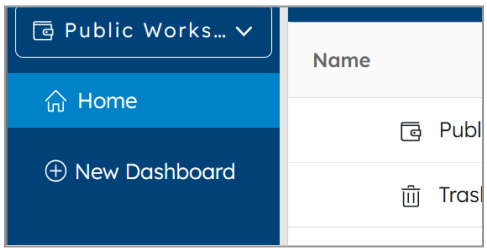
Custom views have been removed. It is no longer possible to create them. Existing Custom views will be converted to Custom dashboards with a 'Fit to screen' 1x1 grid.

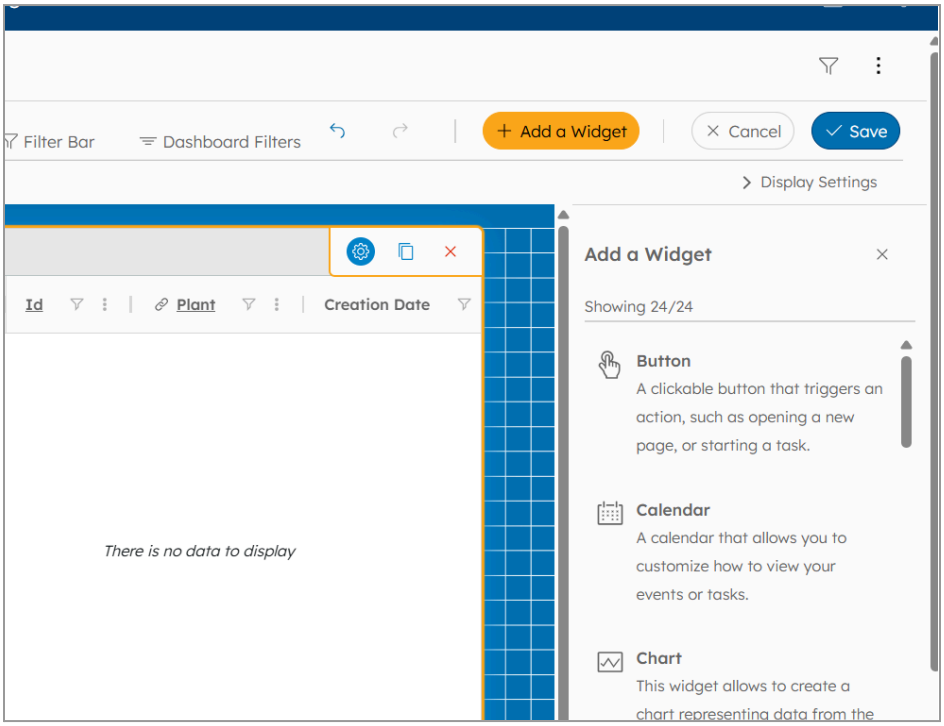
The interface `GeneWidgetCustomViewToolbarController` is now deprecated. `GeneWidgetCustomViewToolbarController.customizeToolbar` will still work; however, the preferred method for adding custom buttons to the toolbar is via the dashboard settings. `GeneWidgetCustomViewToolbarController.customizeToolBarConfiguration` will no longer work. The toolbar configuration should be done using the dashboard settings.

New Button "New Dashboard" in the Sidenav

DBPF-8096

On applications without a dashboard, a *"+ New Dashboard"* button is now available by default in the Sidenav.





New Widget Gallery for Dashboards

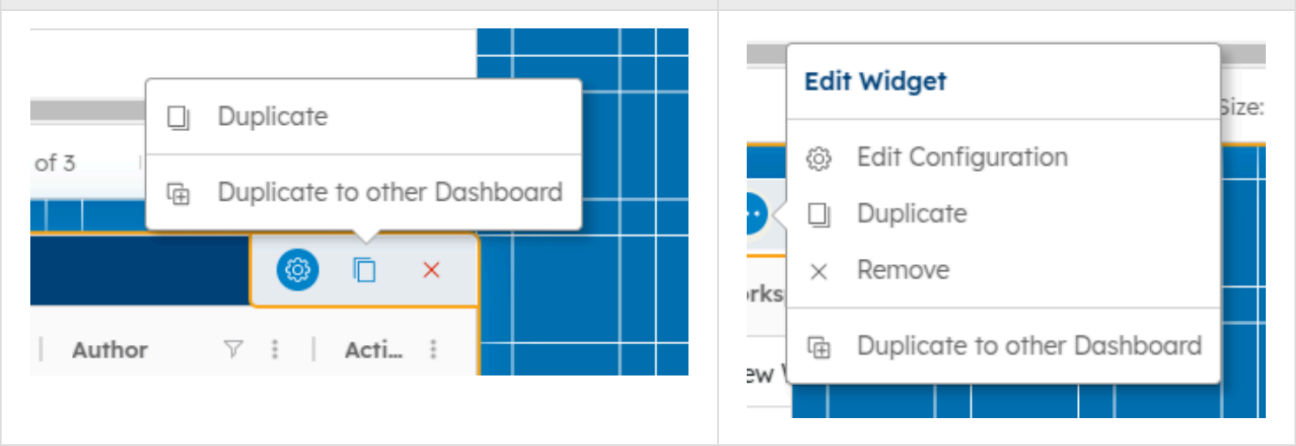
DBPF-8170

On a dashboard in Edition mode, clicking on *"Add a Widget"* now displays a widget gallery with descriptions instead of a dropdown list.

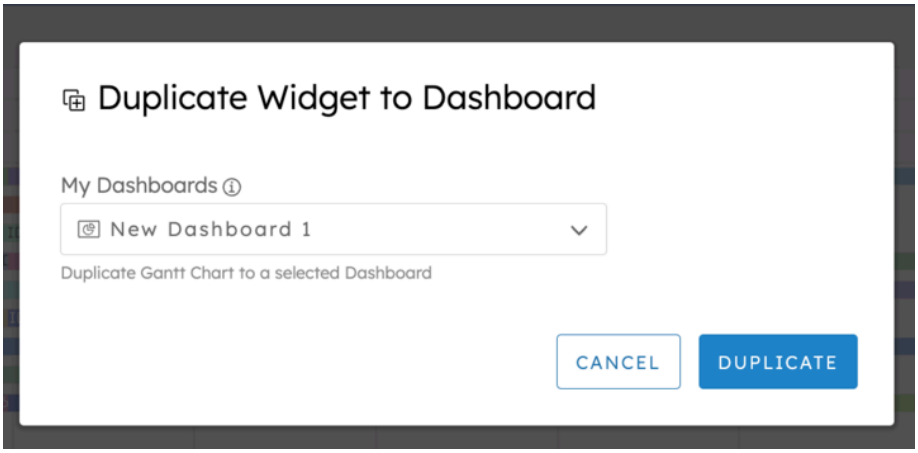
New Widget Duplication Mechanism

DBPF-8096

An option, "*Duplicate to other Dashboard*", is now available from the widget toolbar in Edition Mode. Its location depends on the widget size,



It opens a modal to select the dashboard where to replicate the widget and to determine some layout options.



Data Changes

This part lists all changes related to:

- *the Data Integration core API, in Section **Data Integration Framework**.*

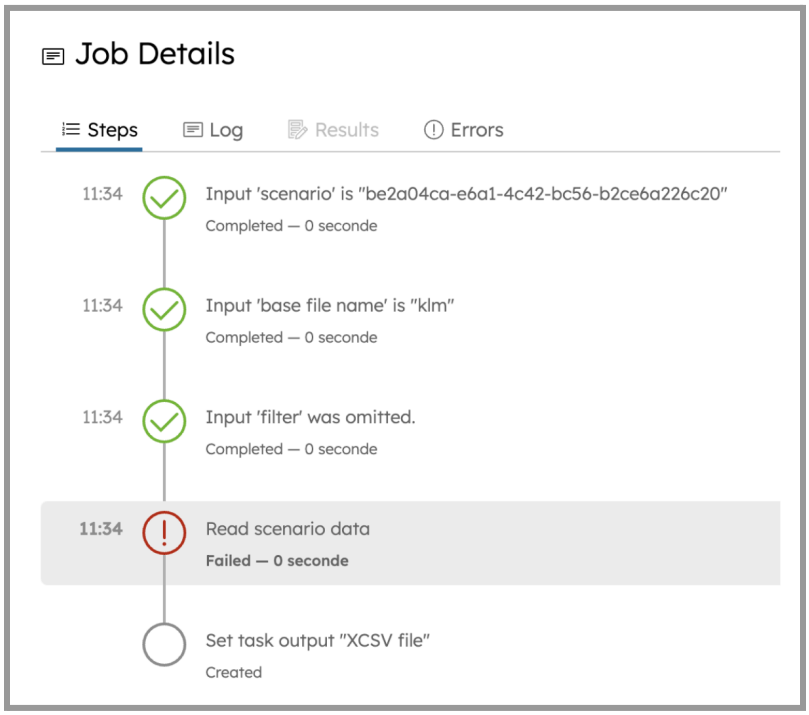
Data Integration Framework

This section lists all changes related to the Data Integration core API.

New Auto-Stop Mechanism for Data Integration

DOC-1148

The import and export tasks using XCSV now stop when an unrecoverable error (such as a database restart) occurs.



Dev Changes

This part lists all changes related to:

- *external dependencies and libraries, in Section **3rd-Party Components**.*
- *scaffolding and Gradle scripts, in Section **Build**.*
- *deployment online, in Section **Gene Online**.*
- *security issues, in Section **Security**.*
- *JupyterLab integration, in **Section JupyterLab**.*
- *Python integration, except for Workers (Python) and Routines, in Section **Python**.*

3rd-Party Components

This section lists all changes related to the IBM DOC external dependencies and libraries, including Spring, Angular, and Keycloak.

Updated Dependencies

DBPF-8195 *Spring Boot* updated from 3.4.3 to 3.4.5

DBPF-7921 *Spring Cloud* updated from 2024.0.0 to 2024.0.1

DBPF-7926 *Keycloak* updated from 26.1.3 to 26.2.4

DBPF-7921 *Jackson* updated from 2.18.2 to 2.18.3

DBPF-8195 *Test Containers* updated from 1.19.1 to 1.21.0

DBPF-8195 *Flapdoodle* updated from 4.18.0 to 4.20.0

DBPF-8197 *Apache POI* updated from 5.2.5 to 5.4.1

DBPF-8197 *Apache commons-text* updated from 1.12.0 to 1.13.1

DBPF-8198 *Mongo* updated from 8.0.5 to 8.0.9

DBPF-8199 *Postgres* updated from 15.10 to 15.13

DBPF-8200 *RabbitMQ* updated from 4.0.7 to 4.1.0

DBPF-8206 *nginx* updated from 1.27.3 to 1.28.0

Updated Angular Dependencies

DBPF-7930, DBPF-8065, DBPF-8201, DBPF-8202, DBPF-8203, DBPF-8204, DBPF-8207

The following Angular dependencies have been updated:

None

```
// Updated Angular Dependencies

ag-grid-angular    from 33.1.1 to 33.3.0
ag-grid-community  from 33.1.1 to 33.3.0
ag-grid-enterprise from 33.1.1 to 33.3.0

graphql from 16.3.0 to 16.10.0
json-to-graphql-query from 2.2.2 to 2.3.0
jquery-ui-dist from 1.13.1 to 1.13.3
ngx-quill from 27.0.1 to 27.0.2

@angular/common from 19.1.7 to 19.2.11
@angular/core from 19.1.7 to 19.2.11
@angular/forms from 19.1.7 to 19.2.11
@angular/platform-browser from 19.1.7 to 19.2.11
@angular/platform-browser-dynamic from 19.1.7 to 19.2.11
@angular/router from 19.1.7 to 19.2.11
@angular/animations 19.1.7 to 19.2.11
@angular/cdk 19.1.5 to 19.2.11

@danielmoncada/angular-datetime-picker from 19.1.0 to 19.1.1

@fullcalendar/angular from 6.1.16 to 6.1.17
@fullcalendar/core from 6.1.15 to 6.1.17
@fullcalendar/daygrid from 6.1.15 to 6.1.17
@fullcalendar/interaction from 6.1.15 to 6.1.17
@fullcalendar/list from 6.1.15 to 6.1.17
@fullcalendar/timegrid from 6.1.15 to 6.1.17

messageformat from 2.3.0 to 4.0.0-12

// New Angular-Related Dependencies

quill-delta-to-html in 0.12.1
@messageformat/icu-messageformat-1 in 0.10.2
```

The library `keycloak-angular` has been replaced by `angular-oauth2-oidc`, which implies light changes in the web UI and the following dependency update:

```
None
Removed keycloak-js
Removed keycloak-angular

New angular-oauth2-oidc in 19.0.0
New jwt-decode in 4.0.0
```

Also, the provider `keycloakServiceProvider` has to be removed from the web module:

- Delete the file `web/src/app/gene-lib/security/keycloak.provider.ts`,
- Remove its declaration, and
- Add the interceptor from the DI declaration in `AppModule`:

```
None
@NgModule({
   imports: [...],
   declarations: [AppComponent],
   providers: [
       provideHttpClient(withInterceptorsFromDi()),
       provideAppInitializer(() =>
appInitializerFactory(inject(TranslateService))()),
   ]
})

export class AppModule implements DoBootstrap {
   constructor(private readonly authenticationService: AuthenticationService) {
}
   ngDoBootstrap(appRef: ApplicationRef): void {
       this.authenticationService.init()
           .then(() => appRef.bootstrap(AppComponent, 'app-root'))
           .catch((error) => console.error('[ngDoBootstrap] init Keycloak
failed', error));
   }
}
```

Additionally, the new library no longer uses an iframe to refresh the token, so it should be removed from the Gateway CSP in `extensions/gateway-service-extension/config/application.yml`:


```
None
spring:
 cloud:
   gateway:
     filter:
       secure-headers:
         # Accepted patterns are :
         # ========================
         # - Web sockets: Since there is a bug in safari (do not rely on 'self' for WS)
         #   and it is hard to compute the WS schema from the public route secured
property in Spring boot yaml expression.
         #   We open both ws:// and ws:// websocket policies.
         # - fonts from the same origin, base64 data urls and google fonts
         # - images from the same origin, base64 data urls and Gmaps
         # - iframe and redirect to keycloak and anywhere in the same origin
         # - js from
         #      - the same origin
         #      - embed tag <script> (We should disable the embed tags since it expand
the XSS surface)
         #      - unsafe-eval (clarity requires lit-element that use eval() internally
:'( :'( :'( )
         #      - Gmaps
         #      - Tableau
         # - Styles from the same origin and embed tag <style> (the last one is
necessary for angular) and Gmaps

         content-security-policy: "default-src 'self'
ws://${routes.gateway.host}:${routes.gateway.port}
wss://${routes.gateway.host}:${routes.gateway.port} ${gateway.keycloak.server-url};
           object-src 'none';
           font-src 'self' data: fonts.gstatic.com;
           img-src 'self' data: maps.gstatic.com *.googleapis.com *.ggpht
api.maptiler.com;
           frame-src ${tableau.server-domain};
           script-src 'self' 'unsafe-eval' *.googleapis.com online.tableau.com
${tableau.server-domain};
           style-src 'self' 'unsafe-inline' *.googleapis.com;
           connect-src 'self' ws://${routes.gateway.host}:${routes.gateway.port}
wss://${routes.gateway.host}:${routes.gateway.port} ${gateway.keycloak.server-url}
api.mapbox.com;"
```

Updated DBOS Angular Dependencies

DBPF-7222 *file-saver* updated from 2.0.0-rc.4 to 2.0.5

Updated Python Dependencies

DBPF-7218, **DBPF-8194**

The new Python max version has been updated to 3.12.10.

Also, Gene now relies on the `hatchling` Python dependency (build time only)

Build

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

Improved Collector Class Name Input

DBPF-8156

When the collector class name is present in the JDL, it can now be used automatically, and the user is no longer prompted for input.

Improved JupyterLab Docker File

DBPF-7942

There are changes in the `processing/jupyterlab/Dockerfile` file. The JupyterLab Docker file is now simplified using internal artifacts.

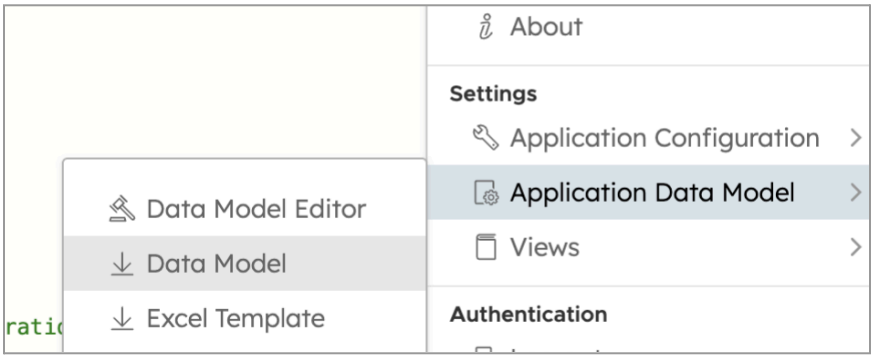
Gene Online

This section lists all changes related to Gene Online.

New Download Option for the Data Model in Gene Online

DBPF-7614

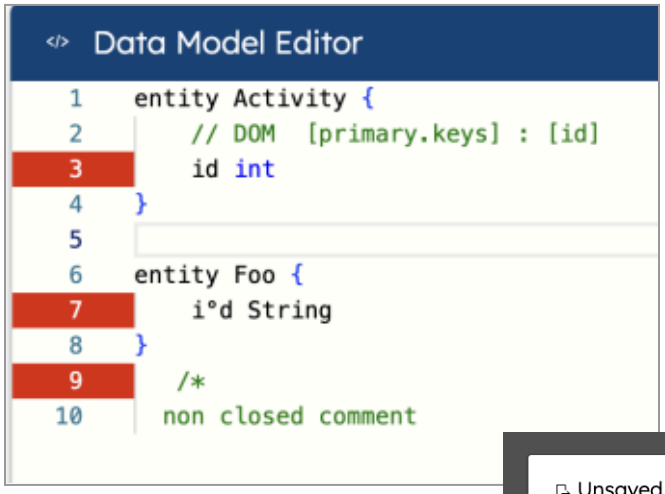
It is now possible to download the data model of a Gene Online application from the Settings menu in the Topbar.

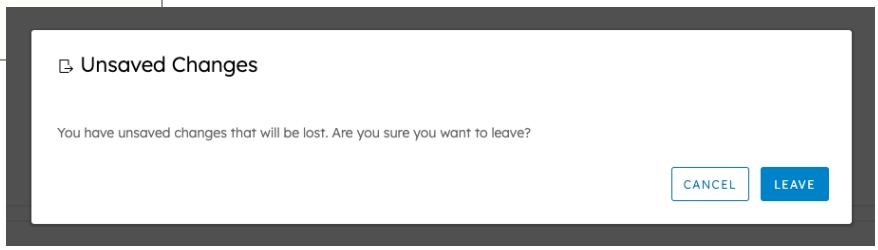


Improved Display for the Data Model Editor in Gene Online

DBPF-7825, DBPF-7606

In the JDL Data Model Editor, all syntax errors reported by the parser are now displayed at the same time. Also, the JDL Editor now notifies the user of unsaved changes before leaving the view.





Improved JDL Converter in Gene Online

DBPF-8007

In the Data Model Editor, when using the AI Assistant to generate a JDL model from imported files, the conversion mechanism leverages the error-reporting improvements made to the JDL parser. This allows the AI Assistant to converge more quickly to a valid JDL to propose.

Security

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

Removed CSP Directive for the Gateway

DBPF-7226

The `unsafe-eval` Content Security Policy (CSP) directive for the gateway has been removed.
If your project or one of its web dependencies depends on `unsafe-eval` to function properly, you can add back the directive in `spring.cloud.gateway.filter.secure-headers.content-security-policy` of `extensions/gateway-service-extension/config/application.yml`.

If this is the case, it is, however, highly recommended to find an alternative solution that does not increase the attack surface of the project.

New Limitations for GraphQL and REST Endpoints

DBPF-7354

The following APIs endpoints expose sensitive data and are only used for the permission table. They are now secured and limited to `permission_admin` users.

GraphQL:

```
None
getUsers
getRoles
```

REST:

```
None
/permissions/users
/permissions/roles
/permissions/roles/non-system
/permissions/users/non-system
```

JupyterLab

This section lists all changes related to the JupyterLab integration to IBM DOC.

New Base64 and Binary Image Generation for JupyterLab

DBPF-8136

Jupyter Helper now provides binary helper methods for file reading and Base64 encoding, so it is now possible to store image content in scenario data, either in Base64 `String` fields or in `Binary` fields.

Also, when displaying a collector, the size of the printed data is now limited to look the same as a DataFrame.

New Idle Prevention Mechanism for JupyterLab in Gene Online

DOC-1112

The system's idle time calculation (`gene_idle_time_seconds`) has been enhanced to accurately incorporate user activity from JupyterLab sessions.

It aims to prevent premature auto-pausing when JupyterLab is in use, by considering both actively open JupyterLab sessions and the most recent recorded kernel activity. Jupyterlab sessions are now considered to report activity on a Gene application.

Python

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

New Python API for Partial Scenario Data Handling

DOC-1121

When working with either dataframes or a collector, scenarios can now be filtered when loading or saving from the Python API. Therefore, it is now possible to add an entity filter parameter to load/save scenario functions in Python.

When working with dataframes, you can now:

- Load a partial `DataFrameDict`:

DataFrameDict contains **1 entity types** and a total of **7 rows**

Entity	Count
Plant	**7**

```
None
scenario_data = scenario.load_as_dataframes({'Plant'})
display(scenario_data)
```

ℹ️ **Note that the `display` function, which is available in Jupyter notebooks, only pertains to notebooks, not workers.**

However, it uses `_repr_html_`, which is available for workers. It creates an HTML block to represent the current entity, as in `<entity_>repr_html_()`.

- Add a new row:

```
None
df = scenario_data['Plant']
new_rows = DataFrame([{
   'plant_id': 'Foo',
   'country.id': 'FRA'
}])

scenario_data['Plant'] = concat([df, new_rows], ignore_index = True)
```

ℹ️ **Note that:**

For the moment, using `country.id` to set the country of the new plant only works in Jupyter notebooks, not workers.

As a reminder, this is how to set a relation in a Python worker. This has not changed:

```
None
new_rows = DataFrame([{
    'plant_id': 'Foo',
    'country.id': 'FRA',
    'country_id': 'any valid internal id of a Country that also have <FRA>
for ID'
}])
```

- Save only one table:

```
None
# Save back the data
scenario.save_from_dataframes(scenario_data, {'Plant'})
```

When working with a collector, you can now:

- Load a partial collector:

```
None
collector = scenario.load_as_collector({'Plant', 'Country'})
display(collector)
```

DbDomCollector contains **24 entity types** and a total of **16 rows**

Entity	Count
Activity	0
Calendar	0
Country	**4**
DateType	0
DefaultValuesTest	0
DemoComputedFields	0
GeneIssue	0
GeneParameter	0
GeneSchemaIssue	0
LobType	0
Plant	**12**

- Add a new entity:

```
None
country = collector.find_country_by_business_key('FRA')
plant = collector.create_plant()
plant.set_plant_id('BAZ')
plant.set_country(country)
```

- Save only one entity:

```
None
# Save back the data
scenario.save_from_collector(collector, {'Plant'})
```

Improved Python Gradle Files

DBPF-7218

The `setup.py` files have been replaced by `pyproject.toml` files. The `package-requirements.txt` files have been deleted, their content has been migrated to the `dependencies = [ ]` section of the `pyproject.toml` file.

The Gradle files that relate to the Python build have been changed to support the migration from `setup.py` to `pyproject.toml`.

The `python-requirements.gradle` files and the corresponding `requirements.txt` files have been removed. As a replacement for the multiple `requirements.txt` files, a unique `requirements.txt` file has been introduced in the root folder. The `requirements.txt` file for JupyterLab remains in place.

Updated Python Module

DBPF-7942

The Pandas deprecation warning has been fixed in the `dbgene-data-python` module.

Updated Python Docker Image

DBPF-8194

Docker image with Python has been updated to use `conda-forge`.

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**.*
- *the UI layout, styling, theming, and interaction principles, in Section **Look & Feel**.*
- *the Scenario Comparison Mode, in Section **Scenario Comparison**.*
- *specific UI widgets, in Section **Gantt Chart Widget**, or **Calendar Widget** — 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.

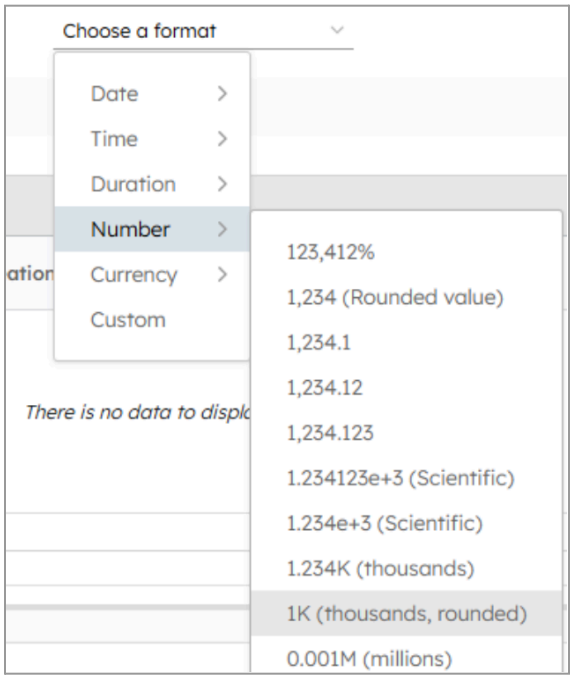
Improved ICU Formatting Library

DBPF-7226

The library used to format ICU messages has been updated.

One notable change affects the option `Number → 1K (thousands, rounded)` of the `gene-icu-format-picker` that can be found in several widgets configurators like the Data Grid, Chart, Gantt Chart, KPI, etc.

If this format was previously selected, it will need to be reselected after the upgrade. This is because the underlying format definition has slightly changed, and values are no longer formatted as expected.



For example, a value like `1234` is now rendered as `1.234K` instead of the intended rounded `1K`. Re-selecting the format will ensure proper rounding behavior is applied.

Improved HTTP API

DOC-1237

The calls to the HTTP API now support both relative URLs `/api/xxx` and `api/xxx`.

Updated Widget Names

DBPF-8171

The following widget names have been updated:

- *Issue Details Widget > Issue Details*
- *Issues List Widget > Issue List*
- *Composite Widget > Composite Widget Container*
- *Map Widget > Map*
- *Calendar Widget > Calendar*

Look & Feel

This section lists all changes related to the UI layout, styling, theming, and interaction principles.

Improved Custom Colors

DBPF-7539

The color of the text now automatically adjusts to CSS variables everywhere in the application when Custom Colors are set.

Scenario Comparison

This section lists all changes related to the Scenario Comparison Mode.

New Multi-Scenarios Comparison

DBPF-7826, DBPF-7827, DBPF-8301

In the Application Preferences, a new parameter `ENABLE_MULTIPLE_SCENARIO_COMPARISON` is set to `true` by default. This allows comparing up to 10 scenarios simultaneously on a dashboard, using the KPI or Chart widgets.

The Comparison Scenario Selector updates accordingly, provided that the dashboard contains at least one of these widgets that support multi-scenario comparison.

Improved Error Management in Scenario Comparison

DOC-773

Scenario comparison is now also available between scenarios with issues of type "errors", as long as they are not schema errors, i.e., they do not appear in `GeneSchemaIssue` but in `GeneIssue`.

Gantt Chart Widget

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

Removed Methods in the Gantt Chart Widget

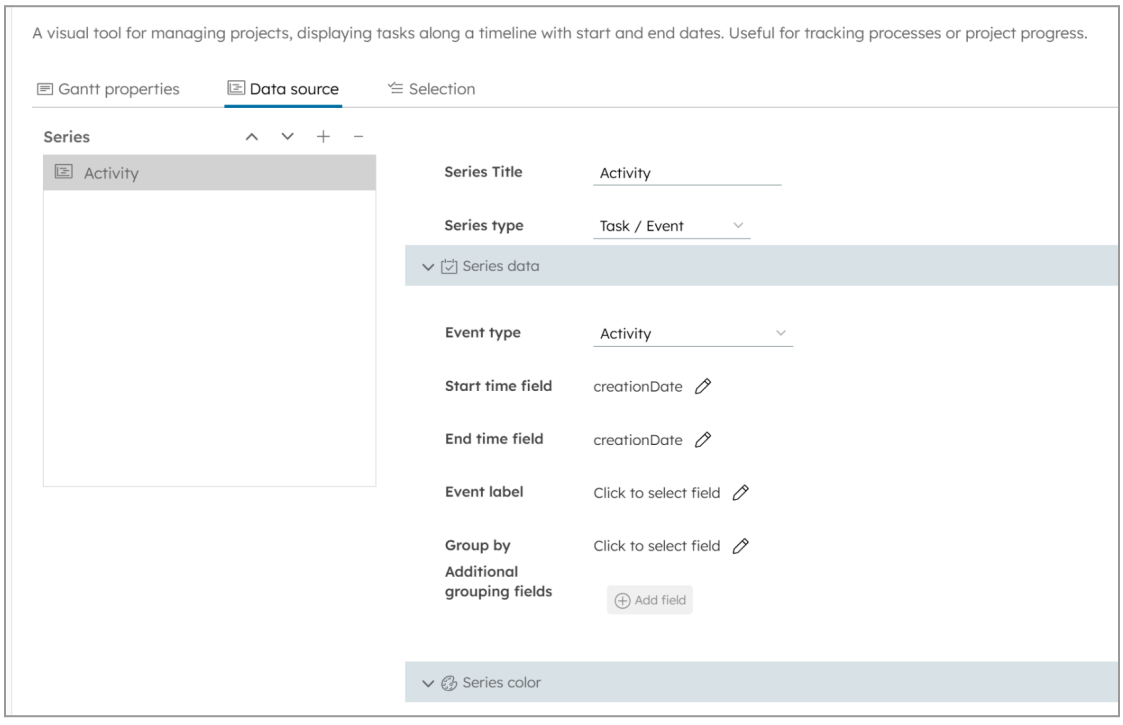
DBPF-7589

The deprecated methods `loadEvents` and `loadResources` have been removed.

New Multi-Series Options in the Gantt Chart Widget

DOC-670, DOC-676

The Gantt Chart widget now supports Multi-Series with a configurator similar to the Chart widget, with a series list on the left and a series configurator on the right.



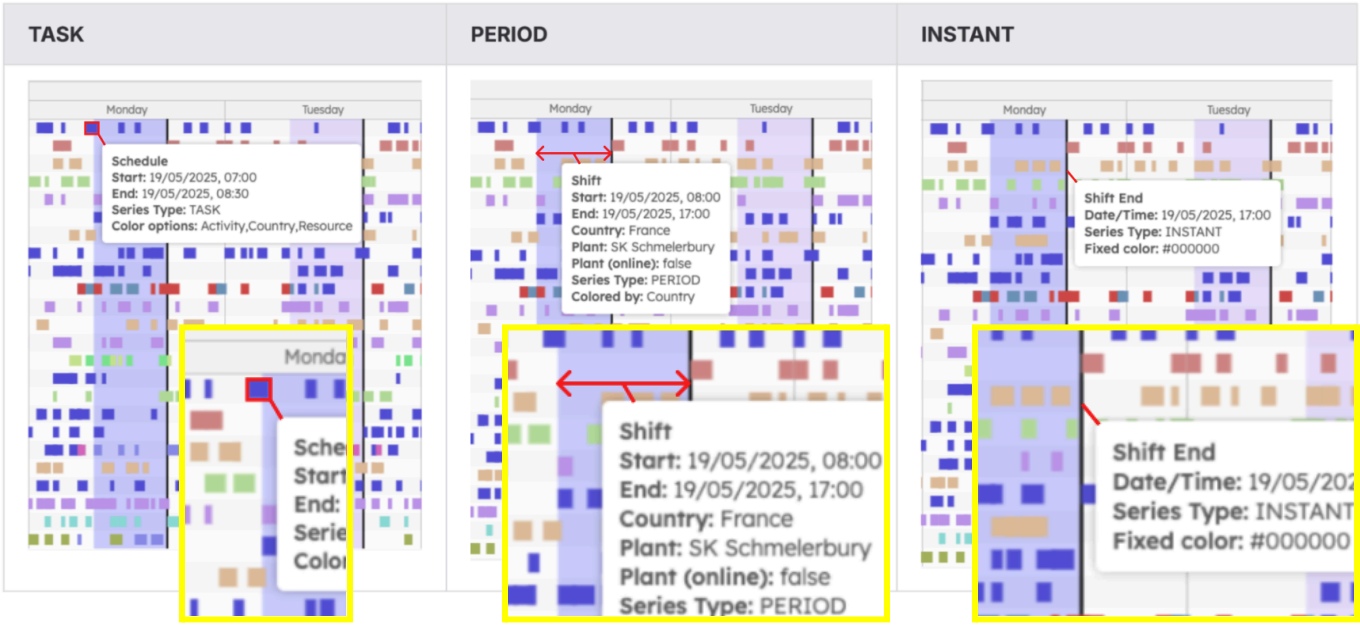
In addition to the color opacity that can now be defined, it is also possible to indicate either:

- a "Single color" for the whole series, to differentiate between the multiple series, or
- a "Color by property" (formerly named "Event color by") and select a field.

New Series Types in the Gantt Chart Widget

DBPF-8002

Different types of series are now available: "*Task / Event*" and "*Period / Shift*", which both span over a determined time length, and "*Instant / Deadline*", for punctual events.



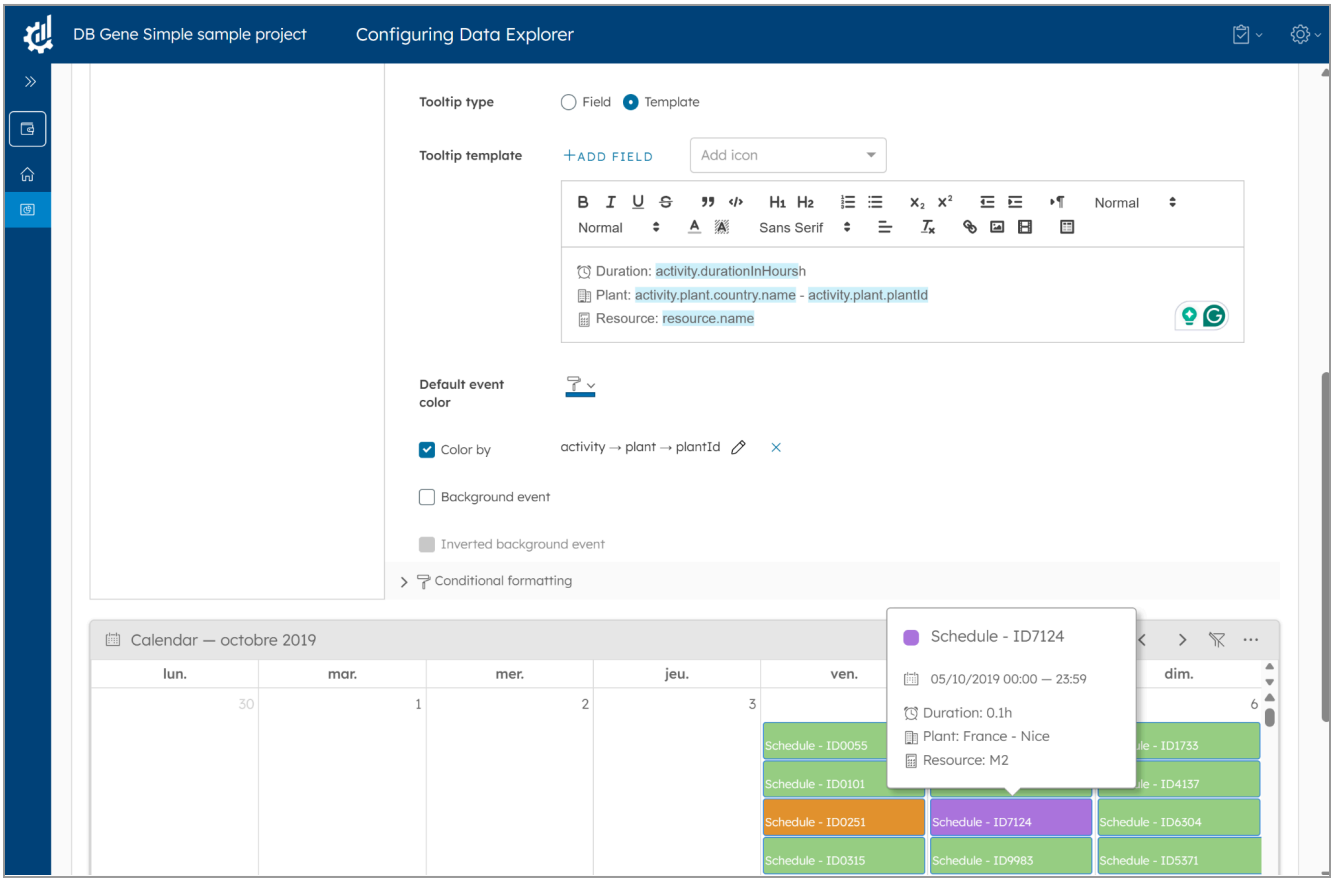
Calendar Widget

This section lists all changes related to the Calendar widget.

New Customizable Event Label and Tooltip in the Calendar Widget

DOC-1134, DOC-1135

The Calendar widget can now use the rich text editor to customize tooltips and labels.



New Color-By Option for the Calendar Widget

DOC-1132

The Calendar widget now supports the "Color by" configuration for events.

Deprecated Features and APIs Scheduled for Removal in 4.4.0.4.7.0

The following deprecated elements will no longer be supported from version 4.4.0.4.7.0.

DOM

- The following JDL syntaxes were **deprecated in 4.0.3-fp1**.

  ```
  - // DOM HIDDEN FK
  - // DOM FIELD [xxx] [foreign.key.reference] ,
  - // DOM FIELD [xxx] - [java.referenceName]
  ```

  → They are now ignored and can be removed from your JDL.

Execution Service

- `com.decisionbrain.gene.execution.model.script.ExecuteRoutineStatement#onBackendService()` was **deprecated in 4.0.1-fp2**.
  → Instead, use `#of(Expression)`.

- `com.decisionbrain.gene.execution.service.JobInstanceService#eraseJob()` was **deprecated in 4.0.3**.
  → Instead, use `JobInstanceService#deleteJobInstanceAndOutputs(String)`.

- `com.decisionbrain.gene.execution.model.ScriptedTaskDescription#getPersistSystemLog()` and `#setPersistSystemLog()` were **deprecated in 4.0.3-fp3**.
  → They should no longer be used.

Web Client

- The **Legacy Pivot Table** and **Legacy Issue List** widgets were **deprecated in version 4.0.3-fp3**.
  → Instead, use the non-legacy widgets.

- The interface `GeneWidgetCustomViewToolbarController` was **deprecated in version 4.4.0.4.6.0.**
  → It should no longer be used.

- `com.decisionbrain.gene.execution.model.script.ExecuteRoutineStatement#onBackendService()` was **deprecated in version 4.0.3-fp3**.
  → Instead, use `#of(Expression)`.

- In **`web/web-frontend-base/projects/gene/widget-data/src/lib/modules/application-state/gene-application.service.ts`**, the function **`getApplication()`** was **deprecated in version 4.0.3-fp3**.
  → Instead, use **`#getCurrentApplication()`**.

- In **`web/web-frontend-base/projects/gene/data/src/lib/modules/data-api/service/gene-data-api.service.ts`**, the functions **`getCurrentSchema()`** and **`getEntities()`** were **deprecated in version 4.0.3-fp1**.
  → Instead, use **`GeneMetaModelService`** and **`getAllEntities(params)`**, respectively.