

DBPLUS
Performance Monitor for PostgreSQL
description of changes in version 2020.4

Date: December 31, 2020

Table of Contents

New in version 2020.4	3
1.1. Changes to the alert mechanism	3
1.1.1. New mechanism for calculating trends for performance statistics.....	3
1.1.2. Excluding PostgreSQL instances from alert monitoring.....	3
1.1.3. Excluding inquiries from monitoring Alerts	4
1.2. Object Size explorer	4
1.3. New Settings page	6
2.1 General Improvements	8
2.1.1 Improving SQL FIND function	8
2.1.2 Data precision in the tooltip.....	8
2.1.3 PostgreSQL instance restart date	8
2.1.4 Security Settings.....	8
2.1.5 Support for objects on Show Plan Objects page.....	9
2.1.6 PostgreSQL version 13 support	10

Below is a list of changes to the DBPLUS Performance Monitor system for monitoring PostgreSQL instances.

New in version 2020.4

1.1.Changes to the alert mechanism

In the latest version of the application, we have introduced several changes to the automatic problem detection mechanism in the monitored PostgreSQL Instance. The main changes:

- Changing the trend counting mechanism for the statistics on which the alerts are based
- Option to temporarily exclude a given instance from alert monitoring
- Ability to exclude individual queries from alert monitoring

1.1.1. New mechanism for calculating trends for performance statistics

The basic change is related to the way of calculating the trend for performance statistics that are involved in calculating alerts. From this version, the trends calculated for the statistics should better reflect the behavior of queries on the monitored instance.

Additionally, the default Alert Problem definitions visible in the **Reasons & Problems definitions** tab have been added and modified.

1.1.2. Excluding PostgreSQL instances from alert monitoring

In the latest version, we have added the function of temporarily excluding a monitored instance from alert monitoring. Configuration is available in the **Configuration> Alert Settings** menu in the **Alerts Outage> Instance Outage** tab. Disabling alerts is a useful feature especially for test environments where is no data continuity, and motion generated during tests doesn't reflect the actual operation of the production environment.

After opening the site, the current configuration will be presented. By navigating the date range, it is possible to verify historical data. To exclude a PostgreSQL instance from monitoring, add a new configuration by click on the **[Add new outage]** button.

Instance	Enabled	Period	Duration in days	Outage days	Outage hours	Description
DESKTOP-HR1BE66/SQL2014	<input checked="" type="checkbox"/>	From 2020-12-01 to 2020-12-31	31	Every Sat, Sun	Whole day	Weekend monitoring Outages
DESKTOP-HR1BE66/SQL_2019	<input checked="" type="checkbox"/>	From 2020-12-28 to 2020-12-28	1	Mon, Tue, Wed, Thu, Fri, Sun	between 11:00 - 15:00	Scheduled work

Then, complete the configuration in the newly opened window, select the instance to be excluded, the date range and, if necessary, the days and times when the exclusion should take place.

OUTAGE DEFINITION

Instance: DESKTOP-HR1BE66/SQL_2019

Enabled:

Period setting

Use begin date: 2020/12/28

Use end date: 2021/01/03

Days pattern and hours range

Outage day(s): Mon Tue Wed Thu Fri Sat Sun

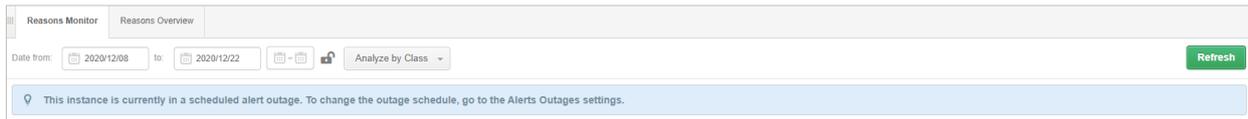
Use range by hours for specified day(s): [] - []

Outage reason and description

New Scheduled work

OK Cancel

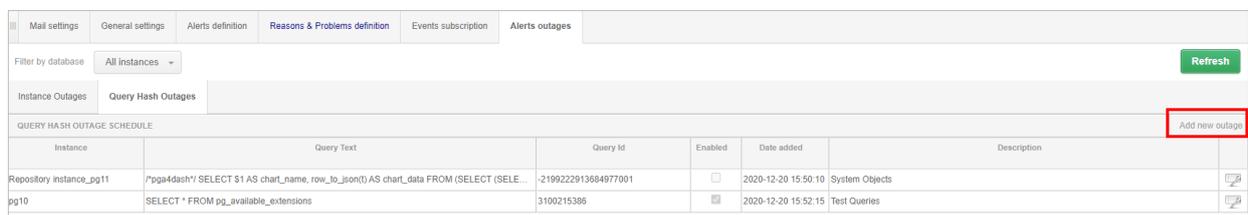
On the Anomaly Monitor screen, if the PostgreSQL instance is excluded from the alert monitoring, a message will be presented:



1.1.3. Excluding inquiries from monitoring Alerts

In addition to temporarily excluding instances from alert monitoring, in the latest version we have added the option to exclude individual queries from alert monitoring. This option is especially useful when the alert thresholds are often exceeded for a given query and the administrator cannot improve the performance of a given query (e.g. in the case of a query in system views). The functionality of excluding the query from monitoring should be in the **Configuration > Alert Settings** menu in the **Alerts Outage > Query Hash Outages** tab.

After opening the site, there is a list of queries for which exclusion is currently configured. To add a new query, click the **[Add new outage]** button. A given query is always excluded from monitoring for a selected PostgreSQL instance, and for a specific query hash.



Excluding from monitoring means that from that moment, when there are performance issues related to the excluded query, the alert will not be reported by the applications.

If the query for alerting is re-enabled, select the editing options from the list and disable the given configuration, or delete the entered configuration by selecting **[Delete]**. After changing the settings, information about problems for the query will be presented again.

Important! After re-enabling monitoring, alerts that were not displayed at the time when the query was excluded from monitoring will not be restored.

1.2. Object Size explorer

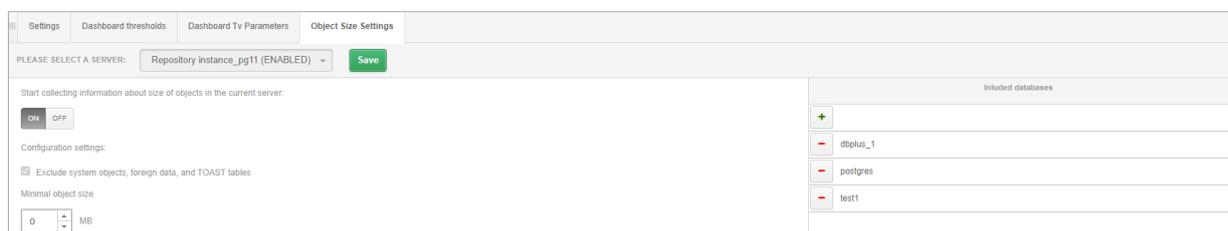
In the latest version, we have added the function of collecting information about the occupancy and the number of rows in objects located in the monitored PostgreSQL instance. From now on it will be possible to verify the occupied space by a given object over time.

Collecting information about the occupancy of objects is a function launched on demand by the User (**disabled by default**). This function, due to the large number of objects for which data must be collected, can significantly affect the size of the DBPLUS repository database. Data on the occupancy of objects are collected at two-hour intervals and saved to the repository database.

From the DBPLUS side, we recommend to enable object collection only for databases on monitored PostgreSQL instances.

Configuration

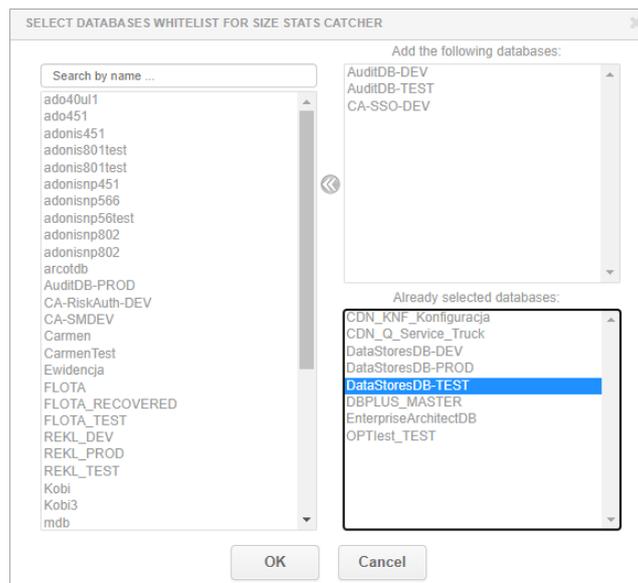
To start collect data on the occupancy of objects, go to the main **Configuration** menu and select the **Settings** submenu, then go to the **Objects Size Settings** tab.



The configuration is always completed for the selected PostgreSQL instance. The configuration consists in selecting from the monitored list the instance for which User wants to collect data on the occupancy of objects. In the next step, User should set the minimum size of the object that will be monitored. The default value of 10 MB means that information about objects (e.g. tables) which are larger than the given size will be collected. The last step of the configuration is to select the databases for which the User wants to monitor the size of objects.

Important! System objects, "foreign data" and "TOAST" table have been removed from the collection.

To add databases to the list, click the  (plus) button on the right side of the screen. Then select databases from the list of available databases (Note! System databases cannot be selected).



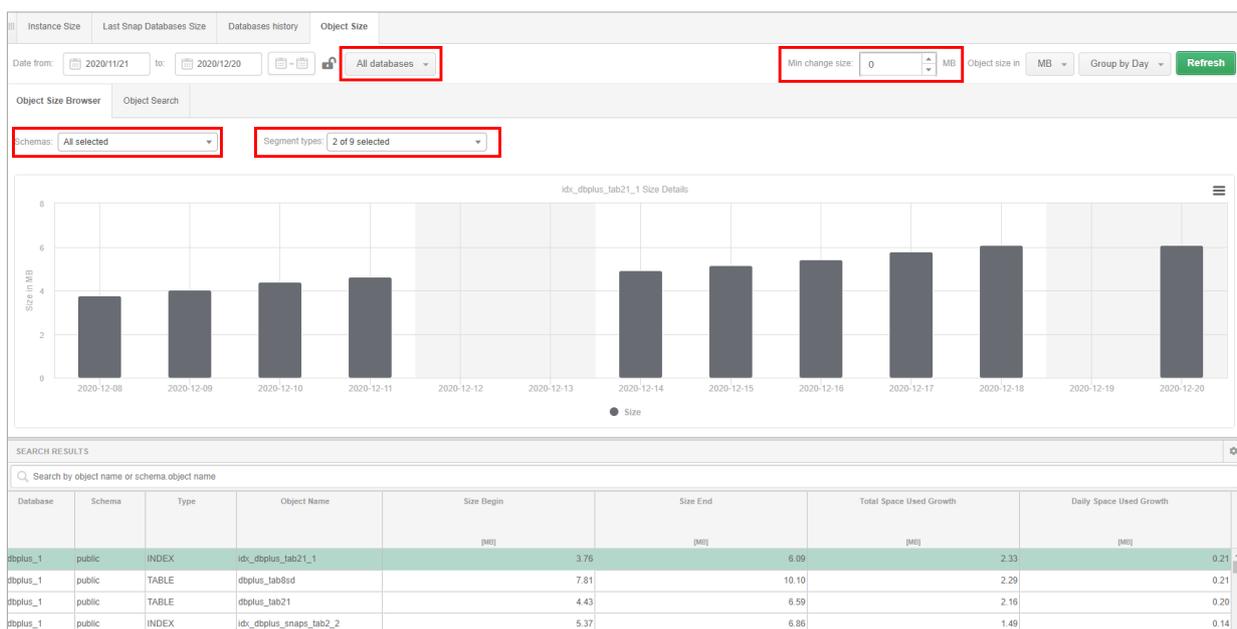
To remove the previously selected database, just click the  (minus) button and save the changes.

Presentation in the DBPLUS application

The data on the occupancy of selected objects can be displayed at the level of details of the monitored PostgreSQL instance, by selecting from the menu **Space monitor** from the **Object Size** tab.

There are two options for searching for objects:

In the **Object Size Browser** tab, it is possible to display information about all objects in the selected databases.

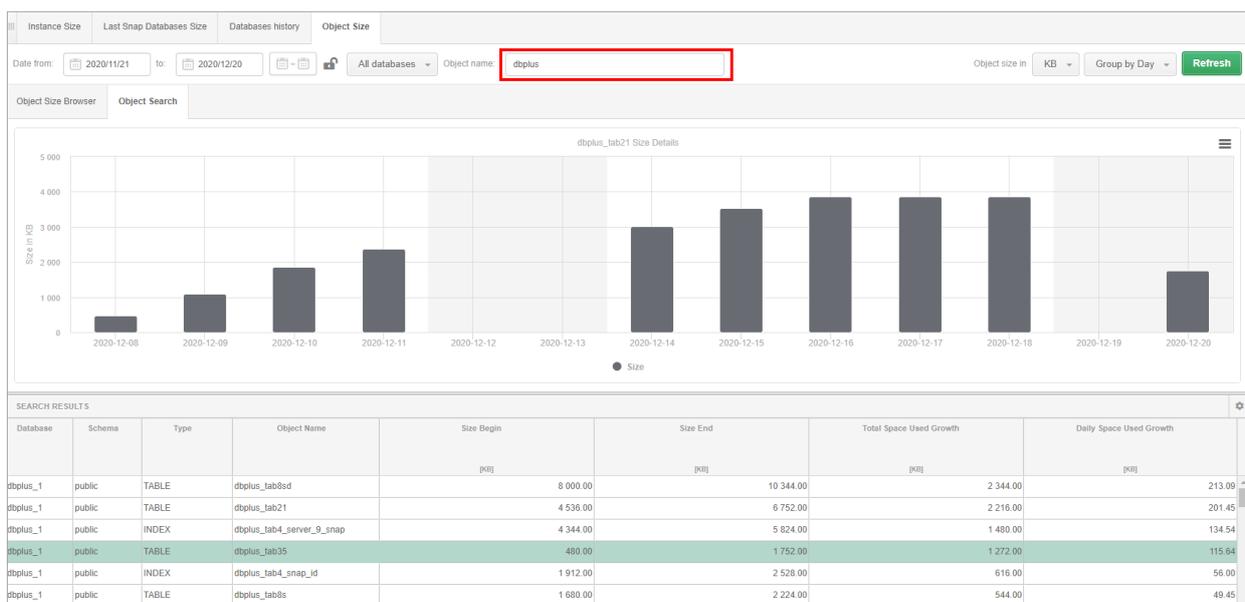


To display data, select for which the User wants to display the objects (**All databases** by default). It is possible to limit the list to a specific scheme (All selected by default). In the next step, it is possible to choose the type of segment that the User wants to display from the list available in **Segments types** (objects of the type: TABLE, INDEX are selected by default). Additionally, User can limit the list by objects larger than the specified size - select a value in the **Min object** size field (given in units of MB). The setting depends on what was set in the configuration under **Settings**. Data can be displayed for a specified period and grouped by hour or day. The history of the object is visible on the chart for the indicated row in the table.

The chart presents information on the occupancy (Size series) for the selected object in the table below the chart. Additionally, the table provides information about:

- Database – name of the database where the object is located,
- Schema – schema name,
- Type – segment type assigned to the object,
- Object Name
- Size Begin – for the first snap of the selected period,
- Size End – for the last snap of the selected period,
- Total Space Used Growth – total size gain over the selected time period,
- Daily Space Used Growth – daily average increase in size of the object over the selected period.

In the **Object Search** tab it is possible to search for an object by name. To do this, enter in the search field the name of the object or the structure consisting of: schema.object_name (case insensitive). After typing, a matching list will be displayed.



1.3. New Settings page

In the latest version, we have modified the current settings screen. The change concerns general settings available in the **Configuration> Settings** menu and the settings for the Dashboard screen available in the **Dashboard thresholds** tab.

General settings

The settings have been divided into areas related to specific parameters:

- General Settings

Contains a parameter that determines the number of days to keep in the repository database.

- Query Statistics

Contains settings for query monitoring.

- Session history Settings

Contains settings for the frequency of collecting information about locks and active sessions.

- Other Settings

Additionally, after making changes, it is always possible to return to the default settings predefined by DBPLUS administrators.

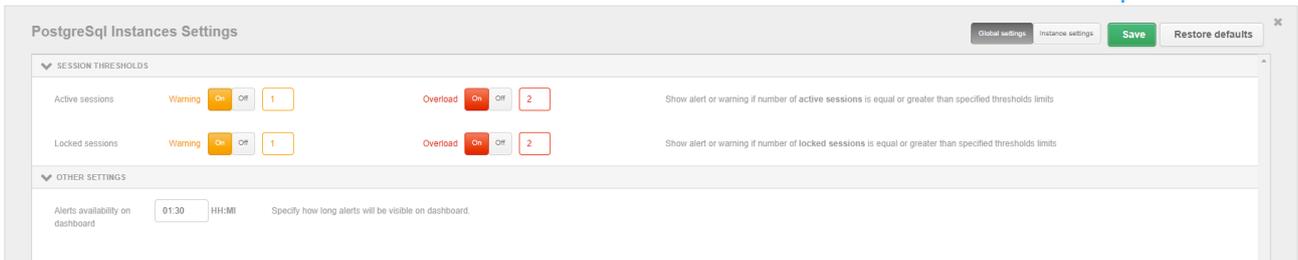
If there is need to change a dedicated instance, select the Instance settings option and overwrite the settings for the indicated PostgreSQL instance.

Dashboard Settings

The settings for the initial screen are available in the same way as for the main settings in the Configuration> Settings menu, the Dashboard Thresholds tab. And by clicking the settings icon in the tile for a given database engine.

Status	Name	Host Name	Version	Startup time	Category	CPU Host [%]	CPU Instance [%]	Waits [s/r/s]	Waits I/O [s/r/s]	Locks [s/r/s]	Active sessions	Locked sessions	Alerts	Space info [GB]
✓	Repository instance_pg11	localhost	11.8	2020-12-02 10:18:08	NOT SPECI...	N/A	N/A	0	0	0	0	0		100 % 0.2
✓	pg13	localhost	13.0	2020-12-07 17:28:03	NOT SPECI...	N/A	N/A	0	0	0	0	0		100 % 0.9

A dedicated parameter settings window contains all available settings for the Dashboard screen. The screen includes the ability to set thresholds for basic parameters that relate to the performance of a PostgreSQL instance. The settings can be modified globally for all instances at the same time, as well as after switching to Instances settings, it is possible to change the settings for a dedicated PostgreSQL instance.



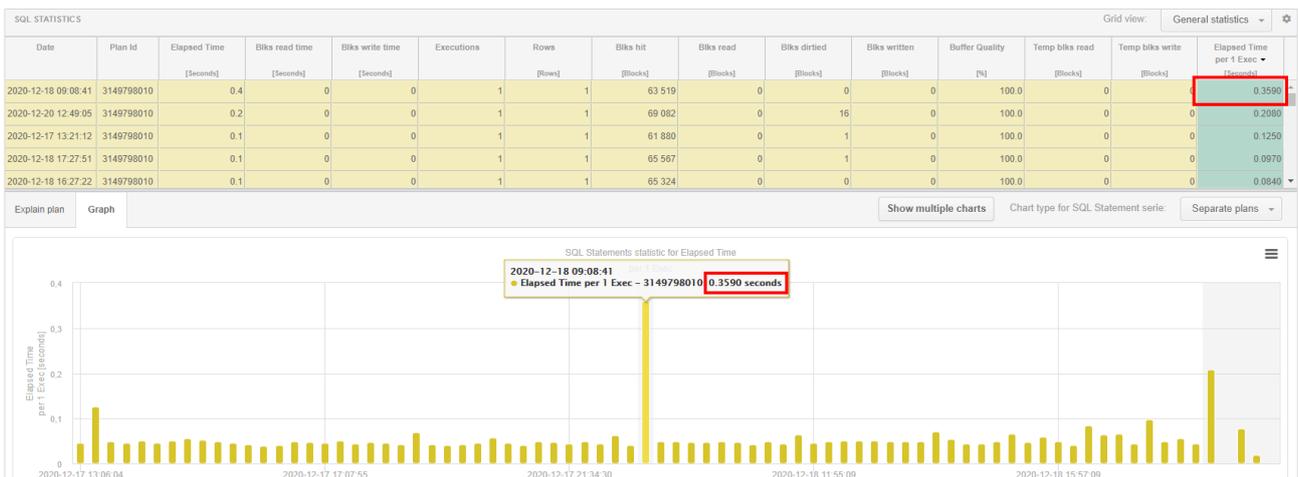
2.1 General Improvements

2.1.1 Improving SQL FIND function

Another change concerns the improvement of remembering previously searched queries. In the latest version, after re-entering SQL Details, the last searches for each of the available search options will be remembered.

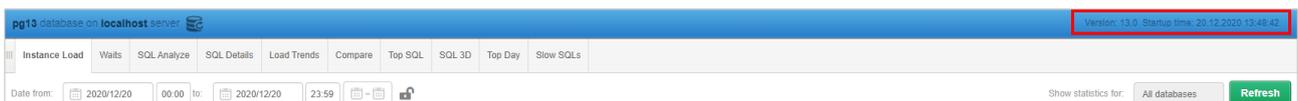
2.1.2 Data precision in the tooltip

In the latest version, we have improved the display of tooltip data on the chart. In previous versions, in some cases, data presented in the tooltip was rounded or displayed with a constant precision to two decimal places. From the latest version, the data in the chart is always displayed with the precision set for a given column in the table based on which the chart is displayed.



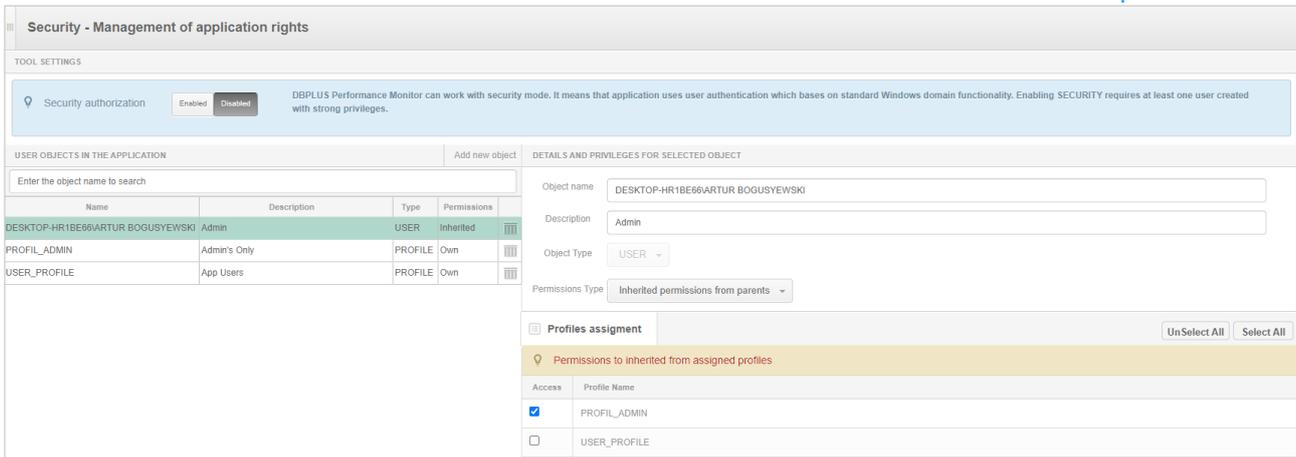
2.1.3 PostgreSQL instance restart date

In the DBPLUS application at the level of the monitored instance, in the upper right corner, the version of the PostgreSQL instance is presented, as well as the date of its last restart. The problem was that in some browsers the information was cached and the actual reboot date was not always displayed. In the latest version we fixed the problem with refreshing information about the last restart of a monitored PostgreSQL instance. The date is updated up to 15 minutes after the instance is restarted.

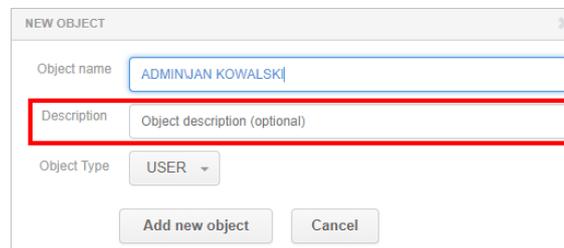


2.1.4 Security Settings

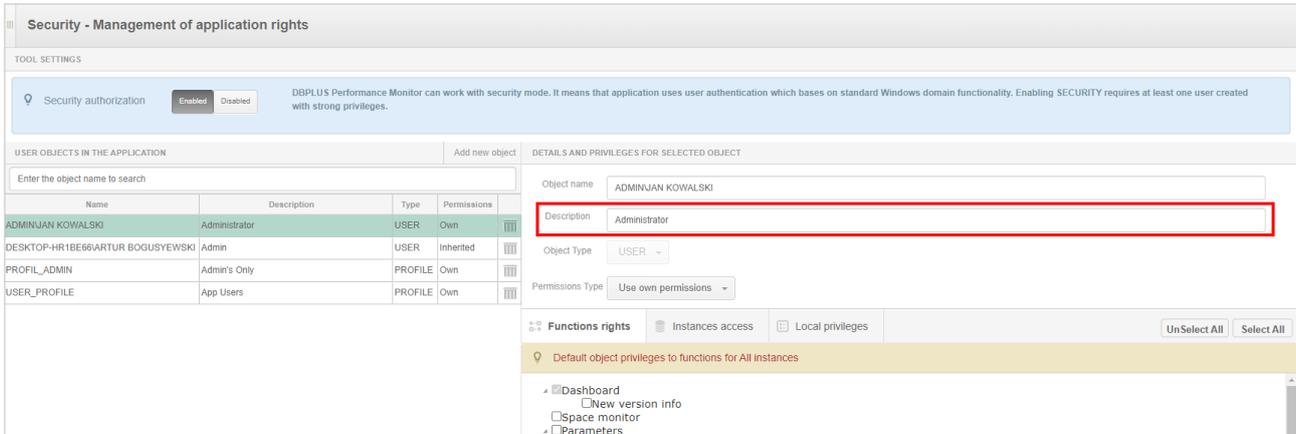
In the latest version of the application, to the Security settings screen available from the Configuration> Security menu, we moved the main parameter of the settings responsible for enabling / disabling the access configuration operation.



Another change concerns the addition of alias information to configured access. From this version on, we can assign additional information to the user / group name or profile. When a new user / group / profile object is created, it is now possible to add additional information. Adding information is optional.



To change the description, select the row to be modified and enter the appropriate value in the Description field.



The last change concerns the speed of writing data to the Security screen. It was a longer wait for a response after saving the changes for customers whose configuration contained many entries. The problem has been fixed in the latest version.

2.1.5 Support for objects on Show Plan Objects page

Support for objects with the name in quotation marks

In the latest version we fixed the bug with the lack of visibility of objects for queries whose names were written in quotation marks. The problem was that the table or index was not displayed on the Show Plan Objects screen in the query details. Support for this type of objects has been added in the latest version.

Object/relation does not exist

On Show Plan Objects screen application verify the query for which the object - table or index does not exists, a message will be presented with information about the missing objects or relation does not exist to given object.

SQL TEXT (QUERY ID: 362842354)				EXPLAIN PLAN (PLAN ID: 347919764)	
<pre> select l_location,l_shipping_agent_service_co,sum(m_invoiced_amount) as m_invoiced_amount,sum(m_old_order_amount) as m_old_order_amount,sum(m_shipped_amount) as m_shipped_amount from sa_ic.prod_nav_bg_sales_estimat_jfjf where true and (((l_location = '836-Know Customer') and (l_shipping_agent_service_co = 'DULOV')) or ((l_location = '836-Know Customer') and (l_shipping_agent_service_co = 'SILISTRA')) or ((l_location = '836-Know Customer') and (l_shipping_agent_service_co = 'TURASOM'))) and (((l_location_code='836') and (l_shipment_time:date) = '2021-01-07') and (l_shipment_time:date) = '2021-01-07') and l_shipment_time::time = '11:15:1' and l_shipment_time::time = '18:15:0' group by l_location,l_shipping_agent_service_co </pre>				<pre> Database: bpladedi_@ @GroupAggregate (cost=42.875 rows=1 width=131) @Group Key: l_location, l_shipping_agent_service_co @Index Scan using prod_nav_bg_sales_estimat_jfjf_l_location_idx on prod_nav_bg_sales_estimat_jfjf (cost=42.875 rows=1) Index Cond: ((l_location)::text = '1'::text) Filter: ((l_shipping_agent_service_co)::text = '1'::text) AND ((l_location_code)::text = '1'::text) AND ((l_sh </pre>	
OBJECTS USED IN EXPLAIN PLAN				INDEXES FOR SELECTED OBJECT .PROD_NAV_BG_SALES_ESTIMAT_JF_JF_L_LOCATION_IDX	
Type	Schema	Object Name	Warning	Name	
INDEX		prod_nav_bg_sales_estimat_jf_l_location	Objectrelation does not exist	No indexes found	
TABLE		prod_nav_bg_sales_estimat_jfjf	Objectrelation does not exist		

2.1.6 PostgreSQL version 13 support

In the latest version of the database, we have added the ability to monitor Postgres instances in version 13. The adaptation was necessary due to changes in the pg_stat_statements system view.