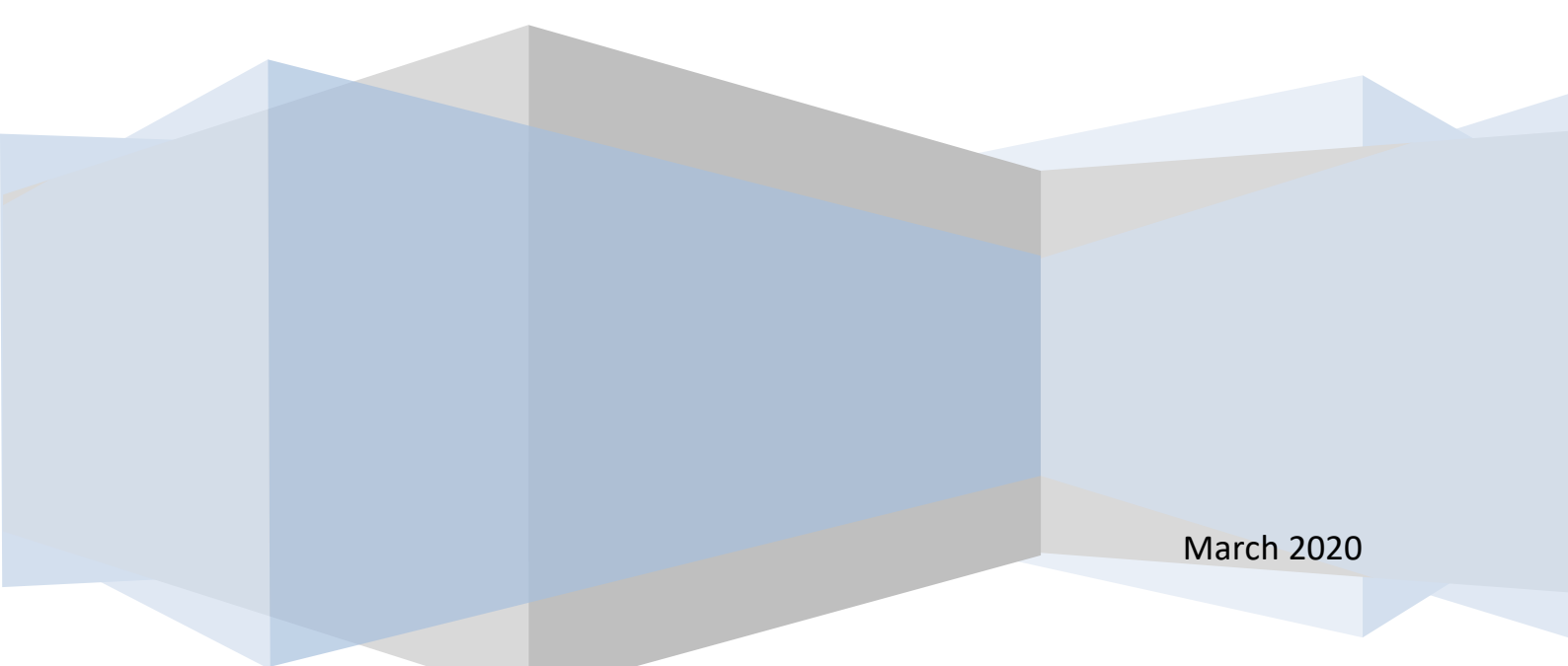




DBPLUS Performance Monitor™ for SQL Server®

User's Manual

An abstract graphic at the bottom of the page consists of several overlapping, semi-transparent geometric shapes in shades of blue and grey, creating a layered, 3D effect.

March 2020

Table of contents

1	Introduction	4
1.1	DBPLUS Technical Support	5
1.2	System architecture	5
1.3	System requirements	6
1.4	Installation of DBPLUS Performance Monitor	7
2	System Configuration	10
2.1	The main configurator screen	10
2.2	Setting up DBPLUSCATCHER monitoring service	12
2.3	System Repository configuration	13
2.3.1	The SQL server name for the database repository	13
2.3.2	Repository database parameters	15
2.3.3	Login/user to connect to the base	15
2.3.4	Add-ons	16
2.4	IIS service configuration	18
2.4.1	Configuration of SSL in the IIS environment	18
2.5	User application configuration	23
2.6	Configuration summary	24
2.7	System configuration file	27
2.7.1	System files for SQL Instance Repository	27
3	Adding a SQL Server instance for monitoring	29
3.1	Import SQL instance from file	33
4	System Upgrade	35
4.1	Setting up for the latest version	35
5	License	39
6	Working with program	40
6.1	Dashboard	40
6.1.1	Information bar	40
6.1.2	The summary area	42
6.1.3	Servers and instances area	42
6.1.4	Details of SQL instance performance	43
6.1.5	Dashboard – various forms of presentation	48
6.1.6	Grid/table options	49
6.1.7	Format SQL text queries	52
6.2	Instance Analysis Menu	53
6.2.1	Performance Menu – Instance Analysis	53
6.2.2	Plan Guides Menu	94
6.2.3	Menu Anomaly Monitor	95
6.2.4	I/O Stats Menu	98
6.2.5	Space Monitor Menu	99
6.2.6	Memory Menu	102
6.2.7	Sessions Menu	105
6.2.8	Jobs Menu	111
6.2.9	Backups Menu	113
6.2.10	Locks Menu	114
6.2.11	Parameters Menu	116
6.2.12	Logs Menu	117
6.2.13	Reports Menu	120
6.3	Space monitor Menu	120
6.4	Accounts Menu	120
6.5	Backups Menu	121
6.6	Parameters Menu	122

6.7	Reports Menu	123
6.8	Servers Monitor Menu	124
6.8.1	Application architecture	124
6.8.2	SQL Server Agent	125
6.8.3	Logs	125
6.9	Configuration Menu	127
6.9.1	Settings	127
6.9.2	Servers	128
6.9.3	Reference lists	129
6.9.4	Security	129
6.9.5	Alert Settings	132
6.10	Help menu	147
6.11	Additional information	147
6.11.1	Export / Import statistics of monitored databases	147

1 Introduction

What is DBPLUS Performance Monitor?

DBPLUS Performance Monitor™ tool is the software used for monitoring and analyzing the SQL instance performance

Using DBPLUS Performance Monitor, you can:

- observe the current database performance,
- track trends of database server load and the individual components: CPU, I / O buffers
- identify performance issues of SQL instances
- track performance trends of individual SQL queries
- analyze data and present them in graphical form
- watch in real time active user sessions
- observe the status of full and incremental databases backups
- troubleshoot a non-optimal SQL queries
- legibly report database problems

and many, many more

Question:

"Why do database work too slow in any specified period of time?"

will never be left without an answer!

1.1 DBPLUS Technical Support








Technical support provides the access to new software updates published 4 times a year as well as to engineers' - help in DBPLUS SQL instance diagnosis (by using **DBPLUS Performance Monitor** software).

1.2 System architecture

The system is designed in client-server architecture and in the presented solution we can distinguish the following components:

- **Databases** - a list of SQL instance covered by the monitoring,
- **Server program** - an application running as a windows service, which consists of a set of procedures performed on individual SQL Instances. The aim of the program is to run periodically procedures, which are responsible for collecting basic data about SQL servers' performance. According to the DBPLUS nomenclature, program is called **DBPLUSCATCHER** and one-up cycle within the service **DBPLUSCATCHER** is called "a snap".
- **Repository** - selected database that stores performance statistics of monitored databases. Collected statistics are the result of the work of **DBPLUSCATCHER** service.
- **Application** - this is a client of the system, which implements user interface which allows to implement functionality of the system, i.e. monitoring review, performance analysis, query execution statistics reports, the current sessions of database, chart of server load, etc. The application is made in web technology using IIS application server and it is accessible from a web browser.

DBPLUS Performance Monitor requires the installation and configuration of each of the elements to ensure full functionality of the solution. Below we present a general model of the system:

Monitored SQL Instances	Monitoring service	Database repository	User application
✓ 3 instances monitored	✗ Service stoped	✓ Configured successfully	✓ Configured successfully
MAQCH\CENTRAL2008  MAQCH\SQLEXPRESS  MAQCH\SQLEXPRESS2012 	 DBPLUS Catcher Status: ● Stopped	 Server: maqch\squlexpress2012 Database: [DBPLUS]	 IIS Service Status: ● Running
			 Application: ● Installed Website: ● running App pool: ● running http://MAQCH/DPM

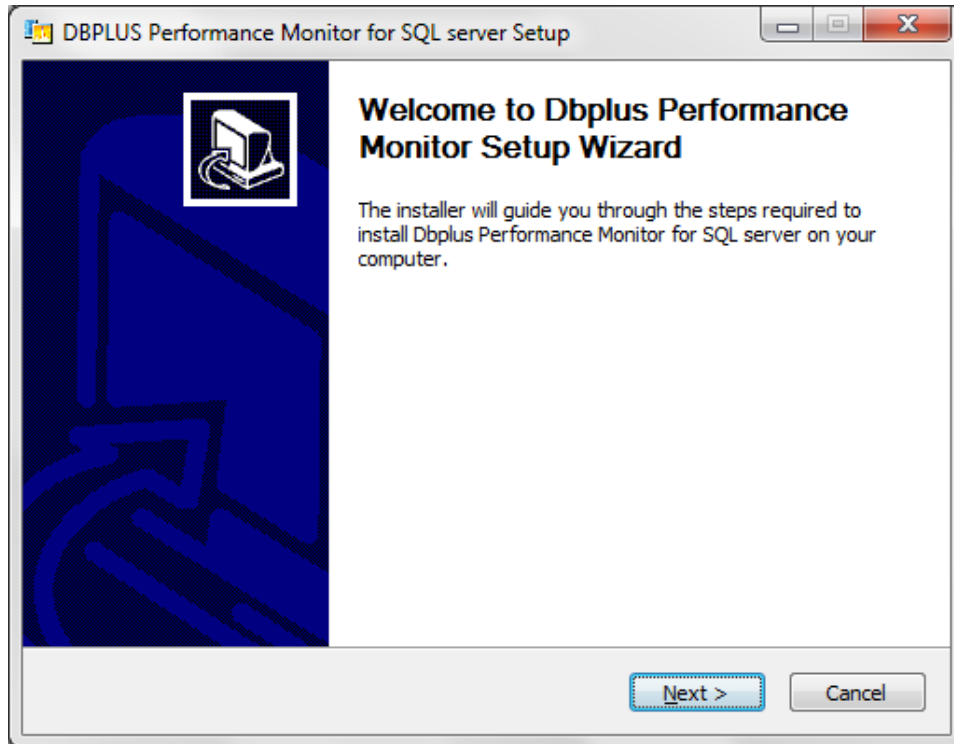
IMPORTANT: DBPLUS System Performance Monitor requires the installation and configuration on any given server / computer in the company. During normal use of application, system does not require any installation on the user's local computers.

1.3 System requirements

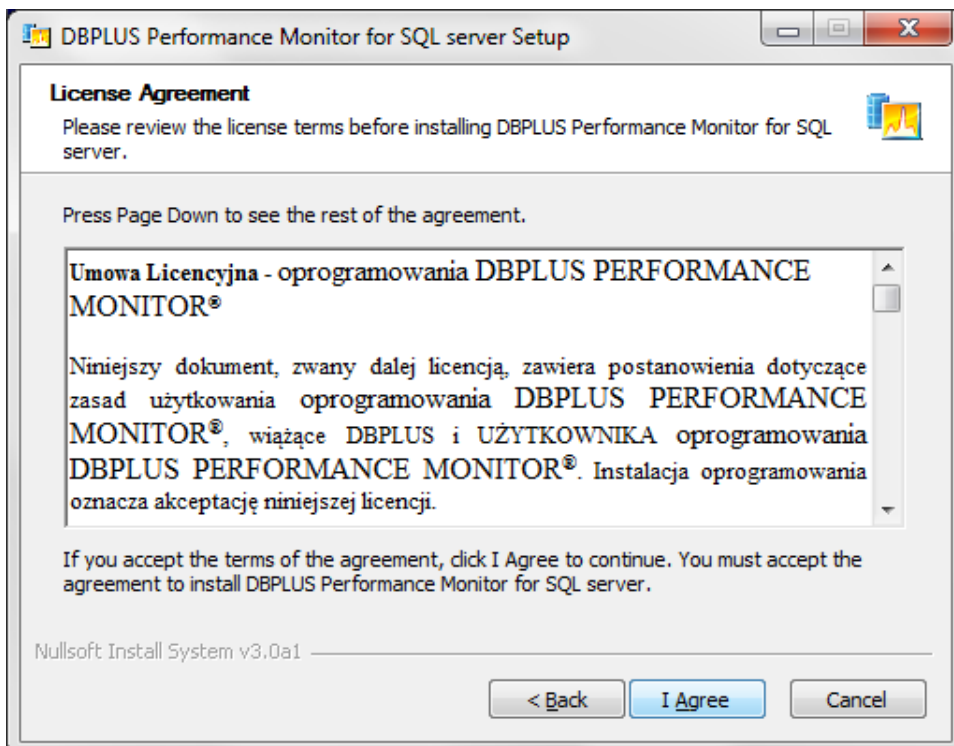
Parameter	Description
Monitored SQL Server	Supported types of monitored SQL Server instance: <ul style="list-style-type: none"> • SQL Server 2005 • SQL Server 2008 • SQL Server 2012 • SQL Server 2014 • SQL Server 2016 • SQL Server 2017 • SQL Azure Edition
Server operating system with installed DBPLUS PERFORMANCE MONITOR software	Servers: <ul style="list-style-type: none"> • Windows Server 2008 and above Also: <ul style="list-style-type: none"> • Windows 7 and above Additional requirements: <ul style="list-style-type: none"> • .NET Framework 4.0 (for DBPLUSCATCHER service) • .NET Framework 4.0 (for the client application). <p>On the server / computer with DBPLUS Performance Monitor software is not required to install MS SQL Server components.</p>
Server's hardware requirements with installed DBPLUS PERFORMANCE MONITOR software	<ul style="list-style-type: none"> • 4 CPU • 8 GB of RAM • HD – no requirements <p>When monitoring 20 instances:</p> <ul style="list-style-type: none"> • DBPLUSCATCHER Monitoring Service consumes at a level 1 GB, IIS to 500 MB of RAM • Assign 4 CPU due to the multithreading services, monitoring a number of instances, plus user applications. • DBPLUS Software is 30 MB, so no special requirements as to the size of the drives on the machine
The impact of the system to SQL Server servers	<p>The system generates an average load of less than 1% dependent on generally accepted "quality" of databases</p> <p>As a result of the installation of repository on a selected database, the system sets up:</p> <ul style="list-style-type: none"> • Database with DBPLUS objects – tables, functions • Login & user with privileges: <ul style="list-style-type: none"> ○ db_owner for repository database ○ privileges allowing for read system views <p>Login is used when connecting to the repository database by the service DBPLUSCATCHER and the user application</p> <p>As a result of inclusion in the monitoring process a specific instance it is set up the user used only to connect with a given instance</p>
User interface	<p>The user application is accessible from a web browser. Supported browsers include:</p> <ul style="list-style-type: none"> • Internet Explorer (ver. 9 and above) • Google Chrome • Mozilla Firefox • Opera

1.4 Installation of DBPLUS Performance Monitor

DBPLUS Performance Monitor is available on DBPLUS server through the provided link. User can install DBPLUS Performance Monitor by double-clicking downloaded EXE file:

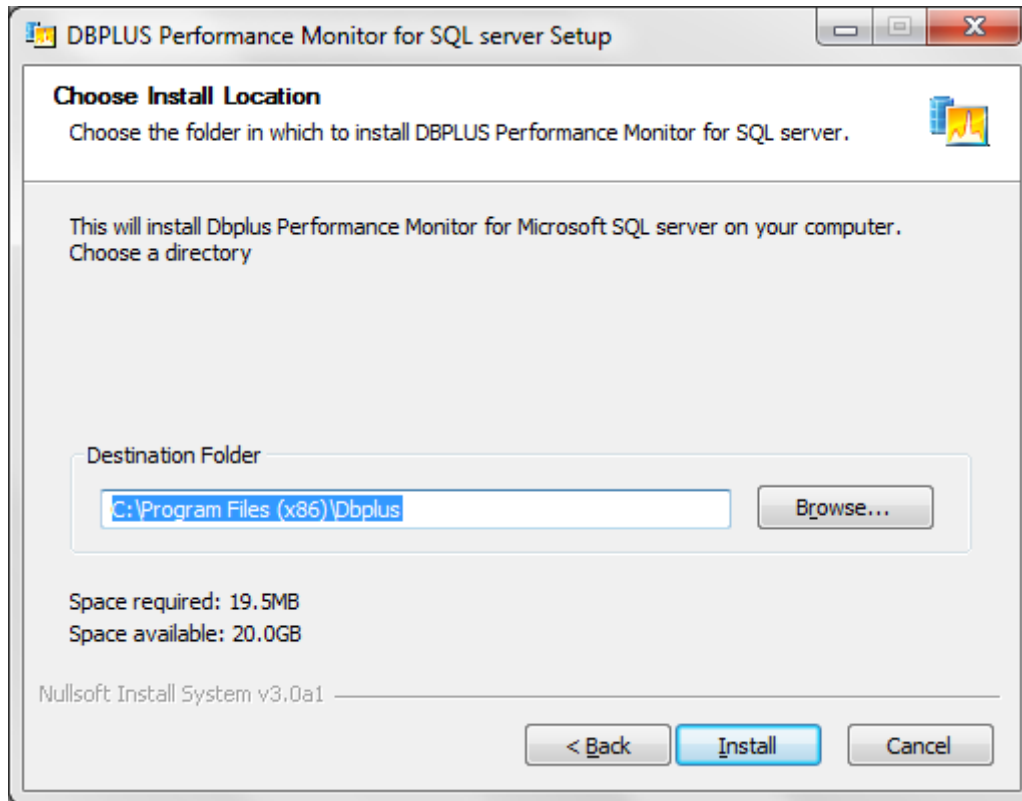


By clicking "Next" we get information about the license:

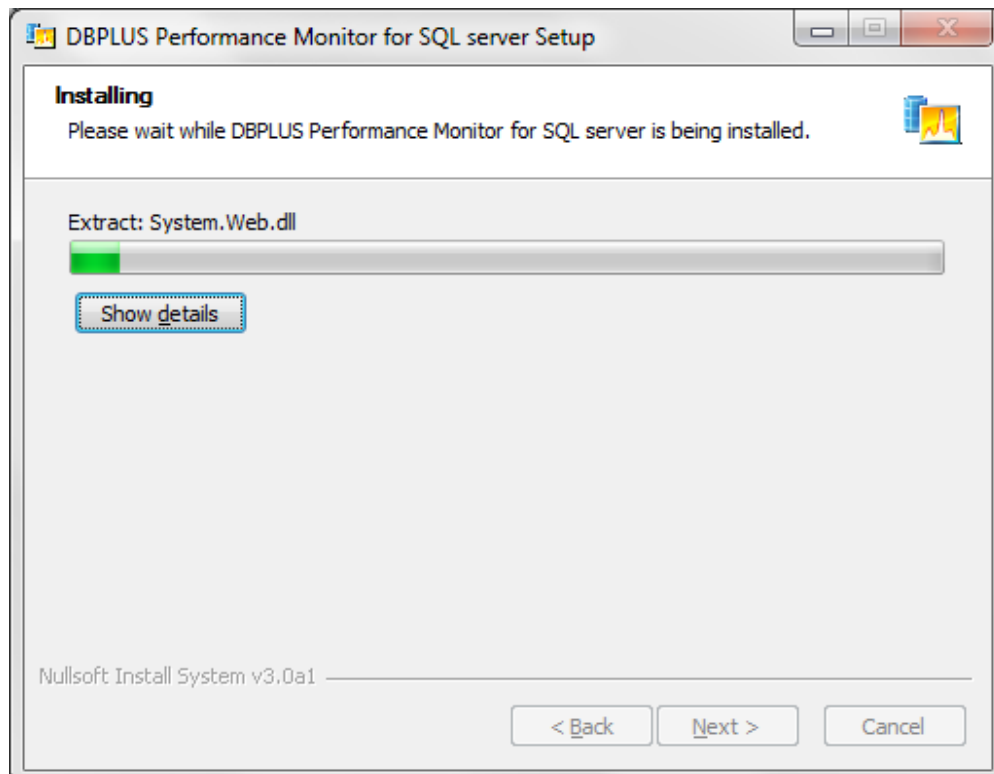


In order to continue installation, you should read and accept the terms of the license. The next step is to select the directory, where DBPLUS Performance Monitor will be installed.

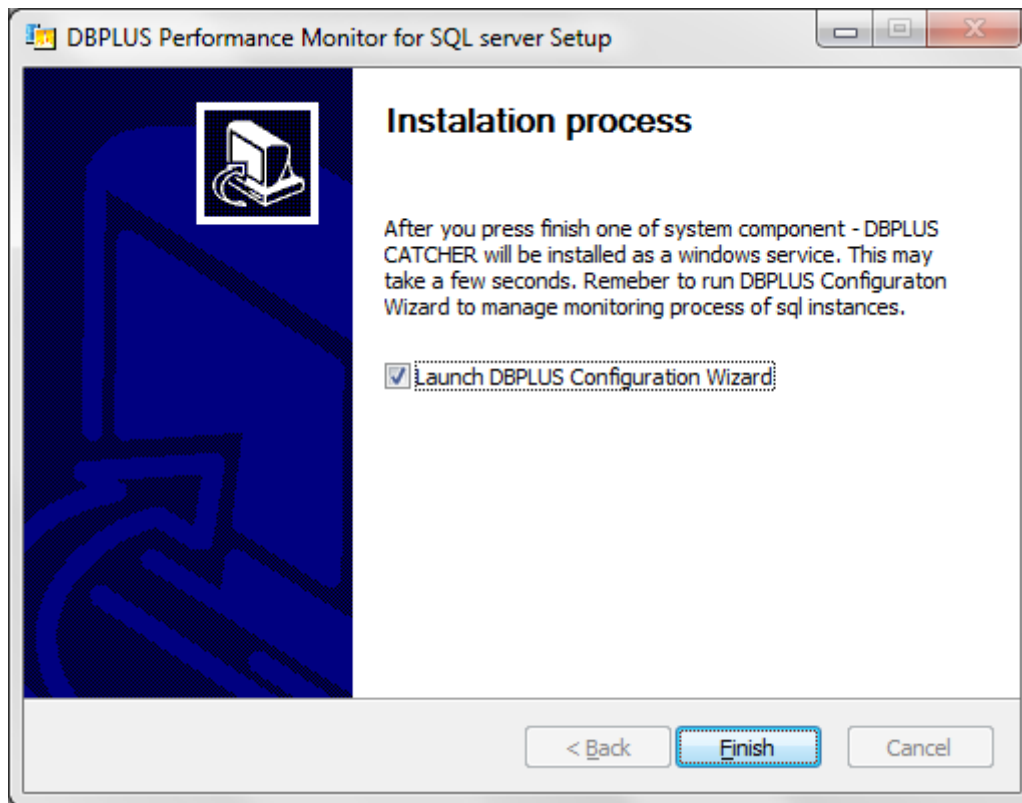
Default directory is C:\Program Files (x86)\DBPLUS



Visible progress of the installation process:



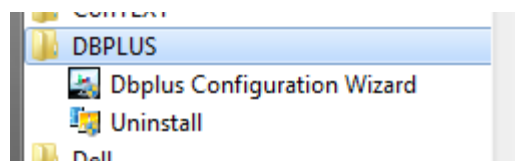
After the correct installation You will receive the following information



The installation process is completed by pressing "Finish" button. By default, we start system configurator, which will carry out the further process of installing individual components of the system or configuration.

Installed DBPLUS Performance Monitor is in the menu:

„Start” -> „Programs”->„DBPLUS”



The following tools are available after the correct installation

1. DBPLUS Configuration Wizard
2. Uninstall

2 System Configuration

In the first stage you must set up a system on the server with **DBPLUS Performance Monitor™** installed, in order to:

- Create a DBPLUS database repository in the selected SQL Server instance, which will store all the information about SQL Server instances performance,
- Inclusion SQL Server servers in the monitoring process,
- Configuration monitoring service DBPLUSCATCHER responsible for gathering information about individual servers' performance,
- User Application Configuration

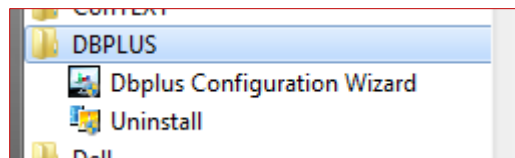
For performing these tasks, system requires permissions:

- Local Administrator on the current server:
- The role of the **sysadmin** on SQL servers, where it is carried out one-time installation of repository or database instances are added to the monitoring

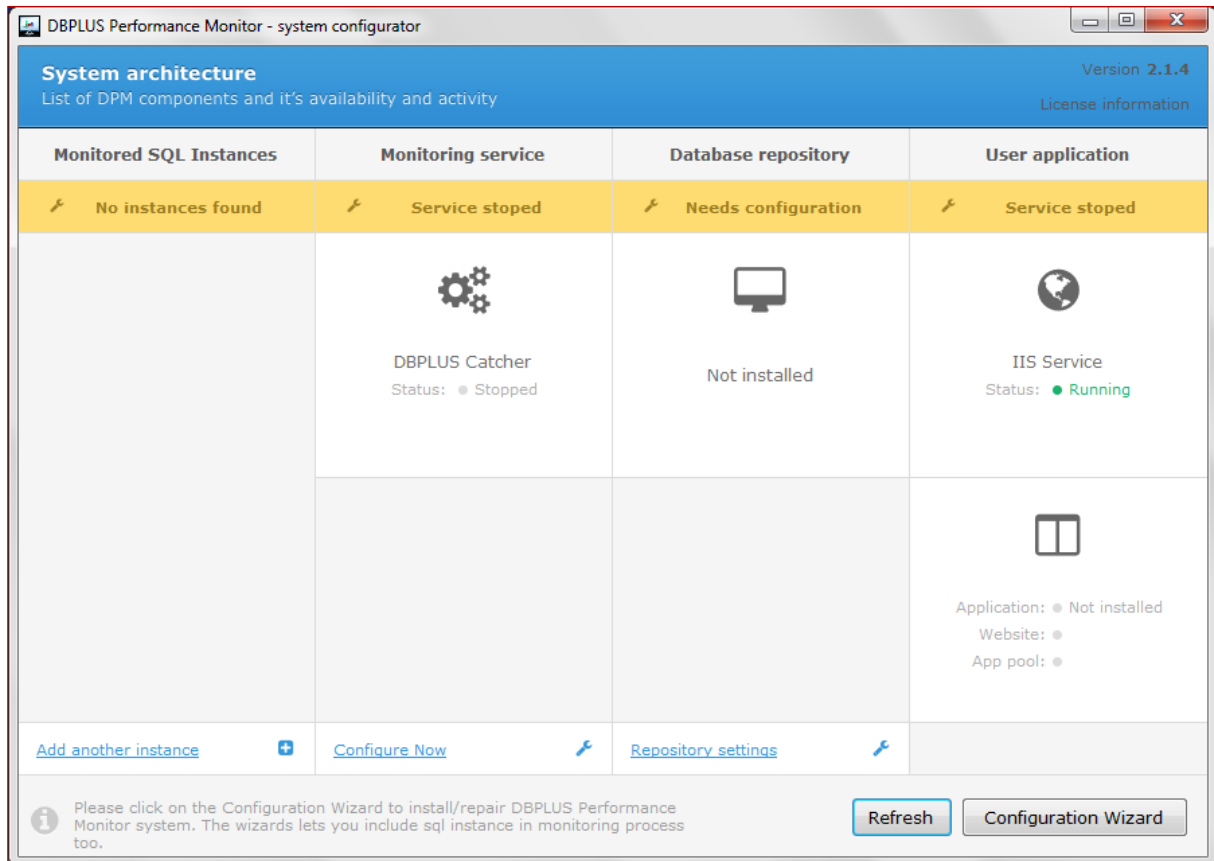
After completing these steps, the application will be available to user from a web browser.

2.1 The main configurator screen

On the server, where software has been installed, by clicking „Start”→”Programs” → DBPLUS→„DBPLUS Configuration Wizard”



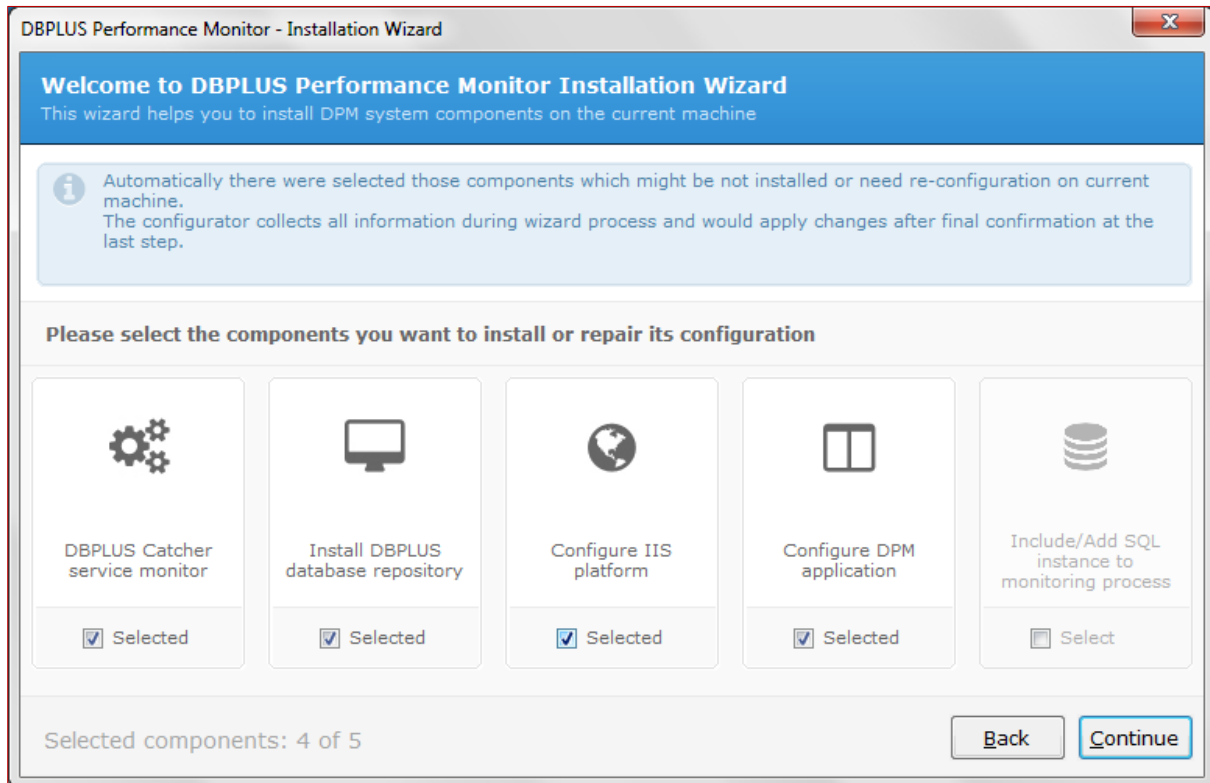
we open a window with system management tool



The main window shows the system architecture and informs among others about:

- number of monitored SQL instances
- place in which monitoring data is stored (Database Repository),
- installation / configuration of individual components of system, for example:
 - lack of monitored SQL Server instances,
 - user application installed or not, application services (IIS website, application pool) running or not,
 - if the monitoring service is enabled.

In order to perform basic system configuration, click **[Configuration Wizard]** button and - as a result - we get this screen to configure individual components.



By default, system selects components that require configuration. You can always reconfigure e.g. a monitoring service or add another (not included so far) SQL database to monitoring.

In the initial stage:

- We create the DBPLUS database repository
- We include / IIS role/service on the current machine
- We configure DBPLUSCATCHER monitoring
- We configure user application

2.2 Setting up DBPLUSCATCHER monitoring service

DBPLUSCATCHER is a program that runs as a service of Windows. When configuring the service, we can set method of authentication and DBPLUSCATCHER might work:

- using a local account (the default setting)
- a domain account / a window

Method of authentication has an influence on the further authentication service on the monitored SQL instances. In the context of a local account, DBPLUSCATCHER service will be making connection with instances of using internal SQL account/s. For domain service account DBPLUSCATCHER will log on to the SQL instance in the context of the account.

Service configuration screen is presented below:

Click on the button **[Continue]** to advance to the next configuration item.

NOTE: All settings - made in the various components of system - are ultimately confirmed in the final step of the creator.

2.3 System Repository configuration

DBPLUS Performance Monitor System Repository is a database that must be created on the selected SQL Server instance. To do this, you have to enter among others.

- Name of SQL Server instance
- Name of database and paths to data and log files
- Login/user name that will be used to connect to a given database

2.3.1 The SQL server name for the database repository

In the first step, system asks for the basic information:

- The name of SQL Server instance
- Determining a user account with **sysadmin** permissions to create a new user that will be used to complete the installation database repository.

DBPLUS Performance Monitor - Installation Wizard

DBPLUS database repository

Specify sql instance where repository database can be installed

Catcher	Repository	IIS	App	Finish

Information: You need to specify the sql instance where dbplus database repository would be located. Database details like name, files and any specific features you can select in the next following steps

SQL Instance/server name: maqch\sqlexpress2012 Retrieve servers

Set an user account with administrator/sysadmin rights.
It will be used to perform database installation on selected instance

Authentication: Windows Authentication

Username: MAQCH\radoslaw

Password:

Connection properties

Step 2 from 8 Back Continue

When you enter the instance name, you have the option to search for all SQL servers on the network - option **[Retrieve servers]**. In the case of custom properties for a connection, you can update them by clicking on the button **[Connection properties]**. In the **[Connection Details]** window we set parameter values, among others Connection Timeout, Application name, etc.

Connection Details

Database Repository

Connection settings

Connection login **Connection properties**

Database

☒ Database name master

☐ Path to MDF file

Connection Pooling

☐ Use connection pooling Min pool size Max pool size Connection lifetime

Other details

Connection timeout 0 Application name DBPLUS Performance Monitor TCP Port 1433

☐ SSL Encryption ☐ Trust server certificate ☐ Use replication

Save Test connection Close

2.3.2 Repository database parameters

The next step is to provide information about database, such as:

- Name
- The path to data and log file
- Initial file sizes and way of incremental growth

The default option is to create a new repository. **The database name can be anything.**

2.3.3 Login/user to connect to the base

Then you have to specify the account parameters that will be used to connect to database. In the case of DBPLUSCATCHER monitoring service configuration in the context of a domain account, the wizard will force the application to give an identical account.

DBPLUS Performance Monitor - Installation Wizard

DBPLUS database repository

Specify login account which will be used by DBPLUSCATCHER service and user application to connect to database

Catcher	Repository	IIS	App	Finish

Information: You need to specify the login which will be used for connection purposes by DBPLUSCATCHER service and DBPLUS Performance Monitor application

We strongly recommend to use the same user account as specified for DBPLUSCATCHER service. If login doesn't exist, then please to create the new one. Please do not use an account with sysadmin privileges. Specified login would be set as an owner for database repository. In addition it would get grants to executes system procedures or read system views on the sql instance

Create new login/user

Authentication: **SQL authentication**

User name: **dbplus_monitoring**

Password: **••••••••**

Use existing login

☐ Use existing user

User name: **aaa**

Password:

Test credentials

Step 4 from 8

Back **Continue**

2.3.4 Add-ons

One of the elements of the database configuration repository is the choice of add-ons:

- **Ole Automation** - a parameter of the server, which is enabled for monitoring disk space of the server
- **db_ddladmin** – access right added to the user, thanks to which we have the ability to view usage statistics of indexes (when selecting a role permission to establish/modify/delete any objects in different databases on the instance are withdrawn)
- Create a task to update user accounts in databases

DBPLUS Performance Monitor - Installation Wizard

DBPLUS database repository

Select special features

Catcher	Repository				IIS	App	Finish

Information: We recommended to select all features to offer better functionality of the system

Ole Automation module and its procedures are used for server disk space usage and estimation analysis.

The role db_ddladmin is required to allow analysis in explain plan and to view statistics (By adding this role, all other privileges like creating, altering, deleting objects in any database are revoke from the specified user)

The job responsible for creating/updating DBPLUS user in any databases is useful particularly in cases where databases are often moved between servers or participate in replication processes

- ☒ Enable **[Ole Automation]** module on the server to monitor disk spaces usage
- ☒ Add **db_ddladmin** role to the dbplus user (recommended to have possibility to view statistics)
- ☒ Enable a job responsible for creating/updating DBPLUS user in any database if required

Step 5 from 8

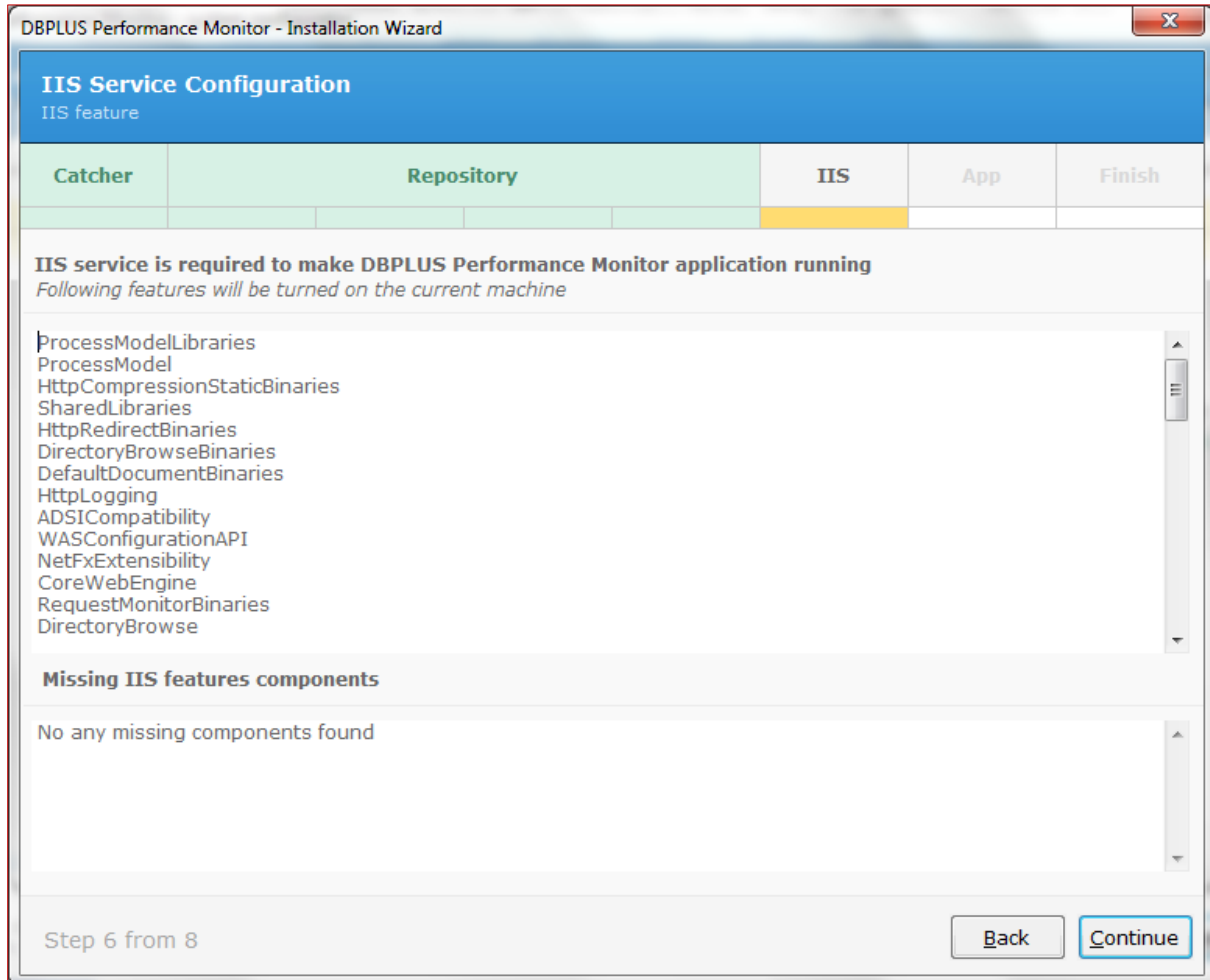
[Back](#) [Continue](#)

We suggest that the choice of all add-ons. After clicking on **[Continue]** go to the next component of the system configuration

IMPORTANT: all settings for the repository database are ultimately confirmed in the final step. At this stage of the database and its objects have not been created yet.

2.4 IIS service configuration

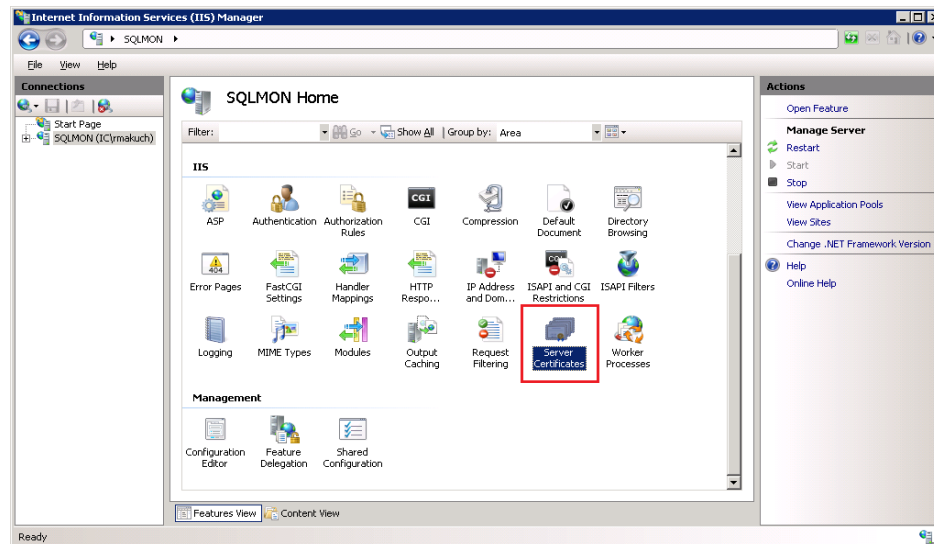
Launching of the IIS role / function on the server is required to run the user interface. The creator window announces additions / services of IIS application server that will be installed. If the “**Missing IIS features components**” box is empty, no configuration is required.



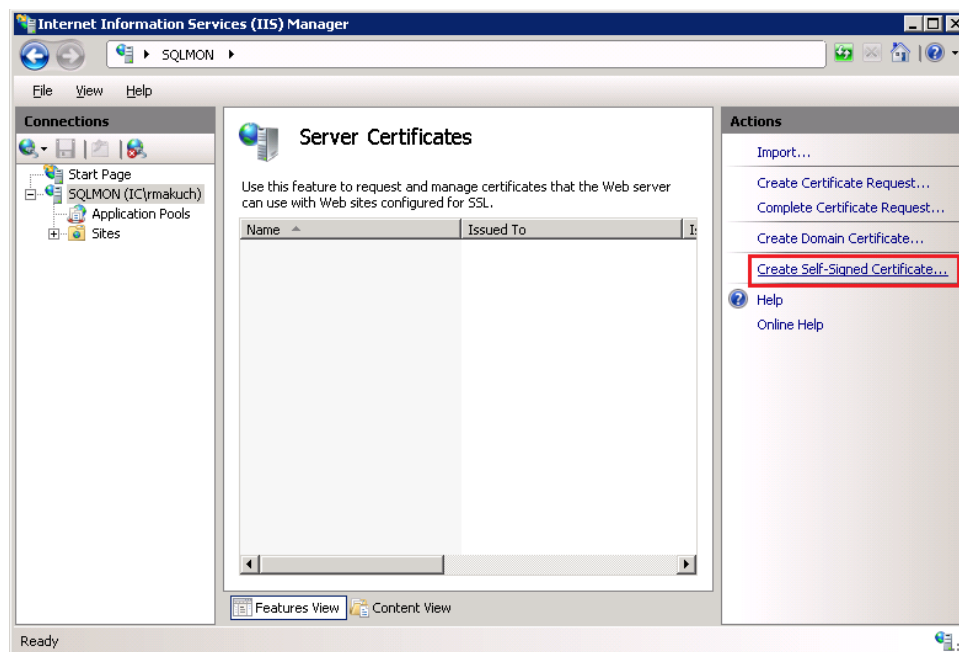
2.4.1 Configuration of SSL in the IIS environment

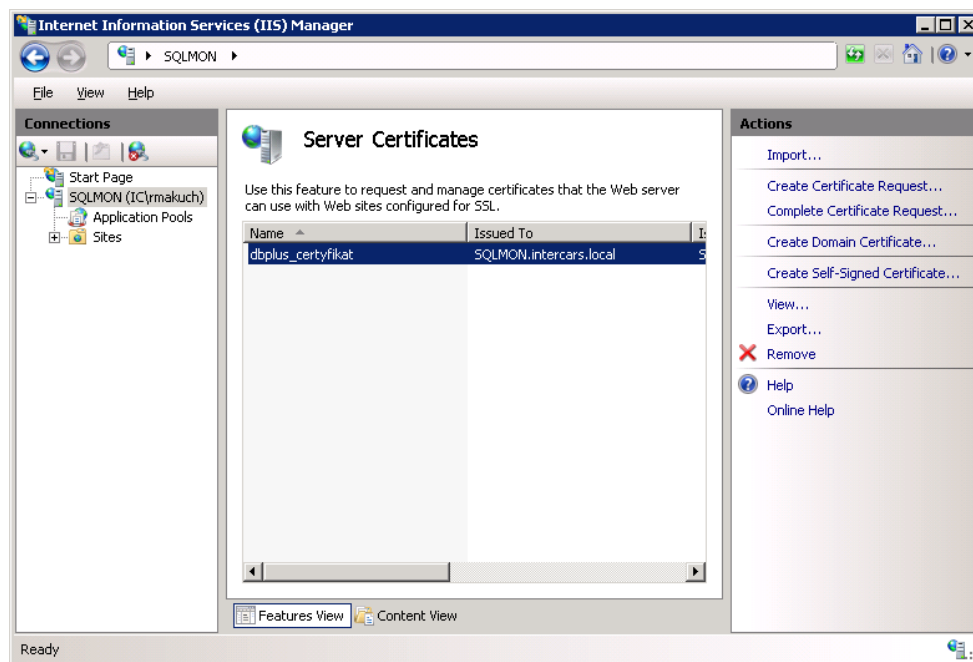
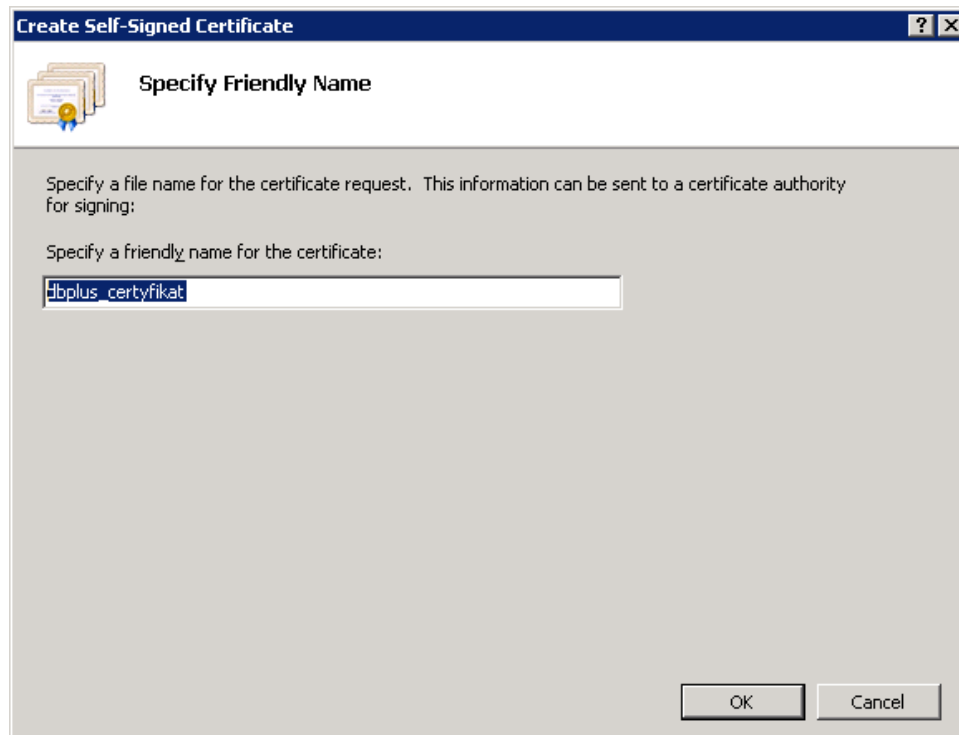
In case you want to enable the SSL functions in the DBPLUS Performance Monitor application, you need to perform the steps on the server with the installed DBPLUS software:

1. Run the IIS Manager (Internet Information Manager) from the command line with the **inetmgr** command
2. For the selected server, find the Server Certificates icon and enter to generate or import a certificate



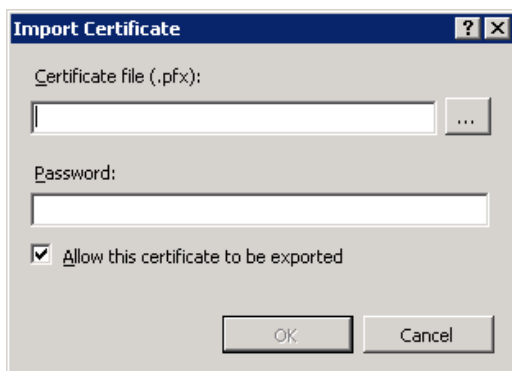
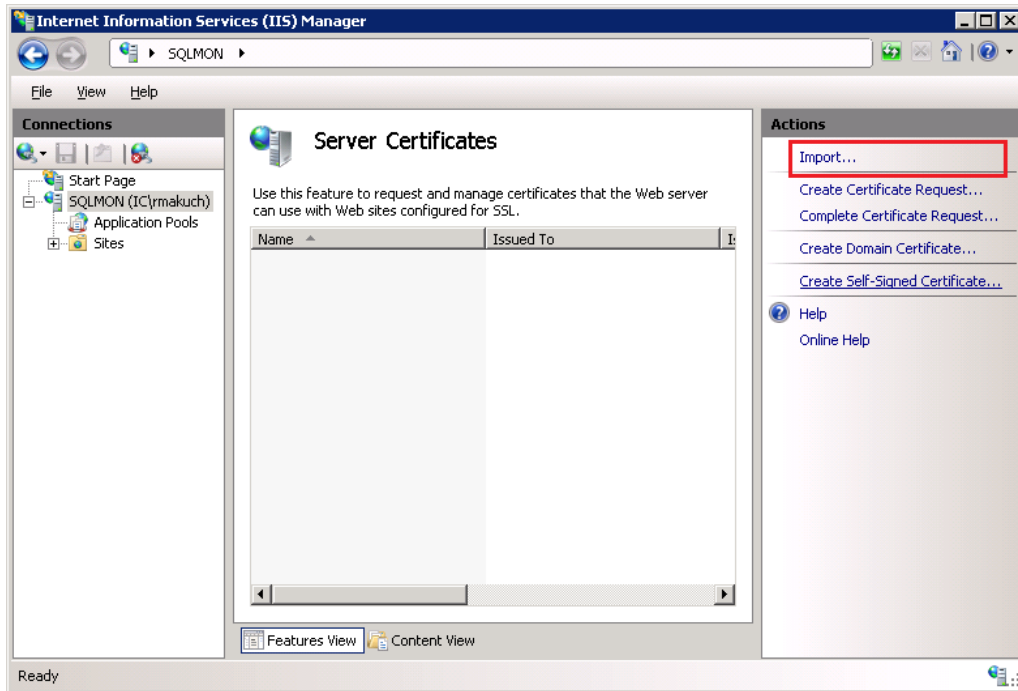
3. Generation of the certificate on the IIS server (in case we do not have it)
We run options according to the below screenshots





4. Certificates import (in case the certificate was not generated directly on the IIS server)

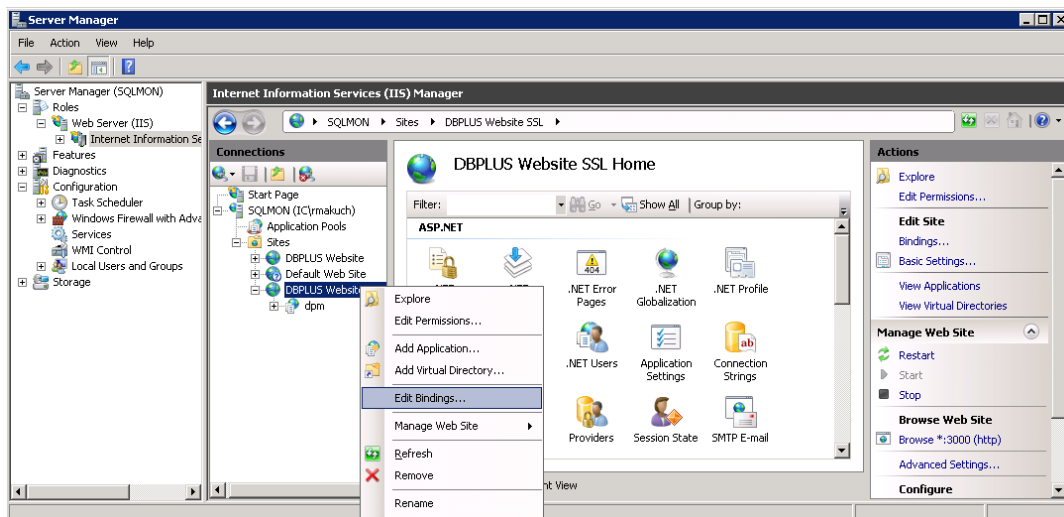
We run according to the following screens:



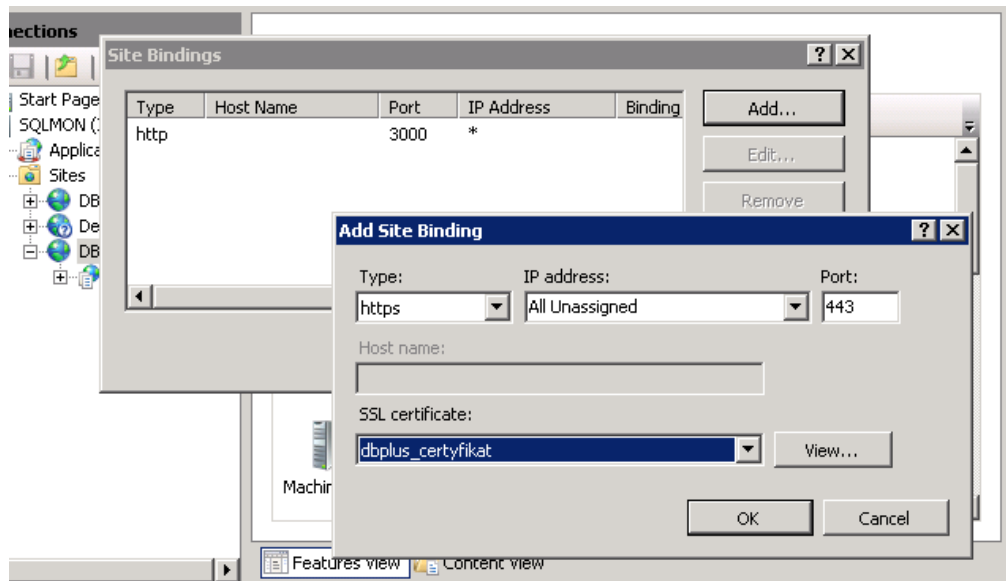
Pass the password if the certificate was exported with a password

5. Addition of the ssl protocol (binding update)

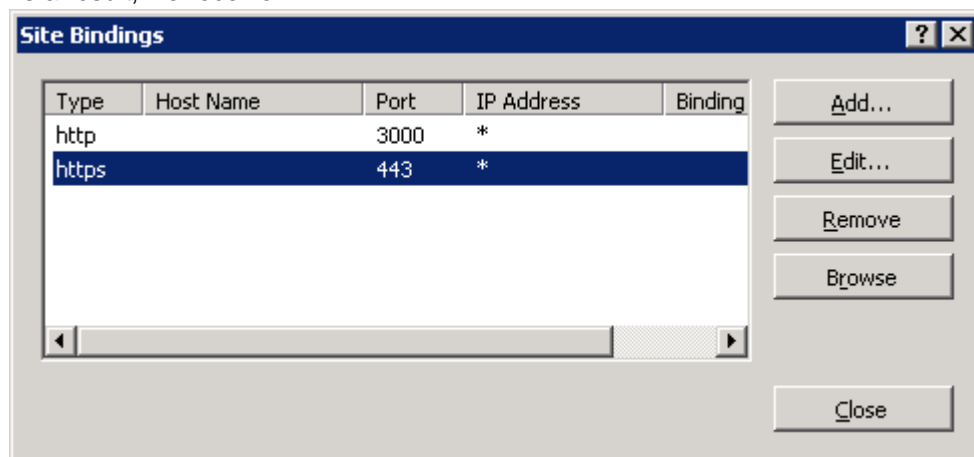
We are updating the link for the DBPLUS Website. Clicking on the site, then Edit Bindings.



In the Site Bindings window, add a new link specifying the SSL protocol and select the certificate previously created or imported as below:



As a result, we receive:



We are removing relationship with the http type.
On the configured DBPLUS Website, we click the restart (Refresh button).

2.5 User application configuration

Another element is the creation of user interface objects. Belong to them:

- Application's website - DBPLUS Website
- Application's pool - AppPoolDPM
- Parameters:
 - Port number
 - Authentication Type - follows the already set method of authentication to an SQL instance (using the SQL or domain account login)
 - The way to access the application - whether users at the login to the site will be asked to authenticate (login and password) or not.

DBPLUS Performance Monitor - Installation Wizard

DBPLUS Performance Monitor - user application
Specify settings for web user application

Catcher **Repository** **IIS** **App** **Finish**

Information: Specifying the user account used for connection purposes by application client, is available during DBPLUSCATCHER configuration or installation database repository

It's required to use the same user type/account for mentioned components (DBPLUSCATCHER service, IIS application and SQL instances monitoring purposes).

You can specify the port for http protocol and turn on/off windows authentication for users who would access the application. Any access and user privileges to the application you can manage directly in the application

Login type:

Username:

Password:

Port:

Access to application:

Application path:

Test user Test port Select application

Step 7 from 8 Back Continue

As a result of the entire setup process completion, your application will be available at the following url:

http://servername:port_number/dpm

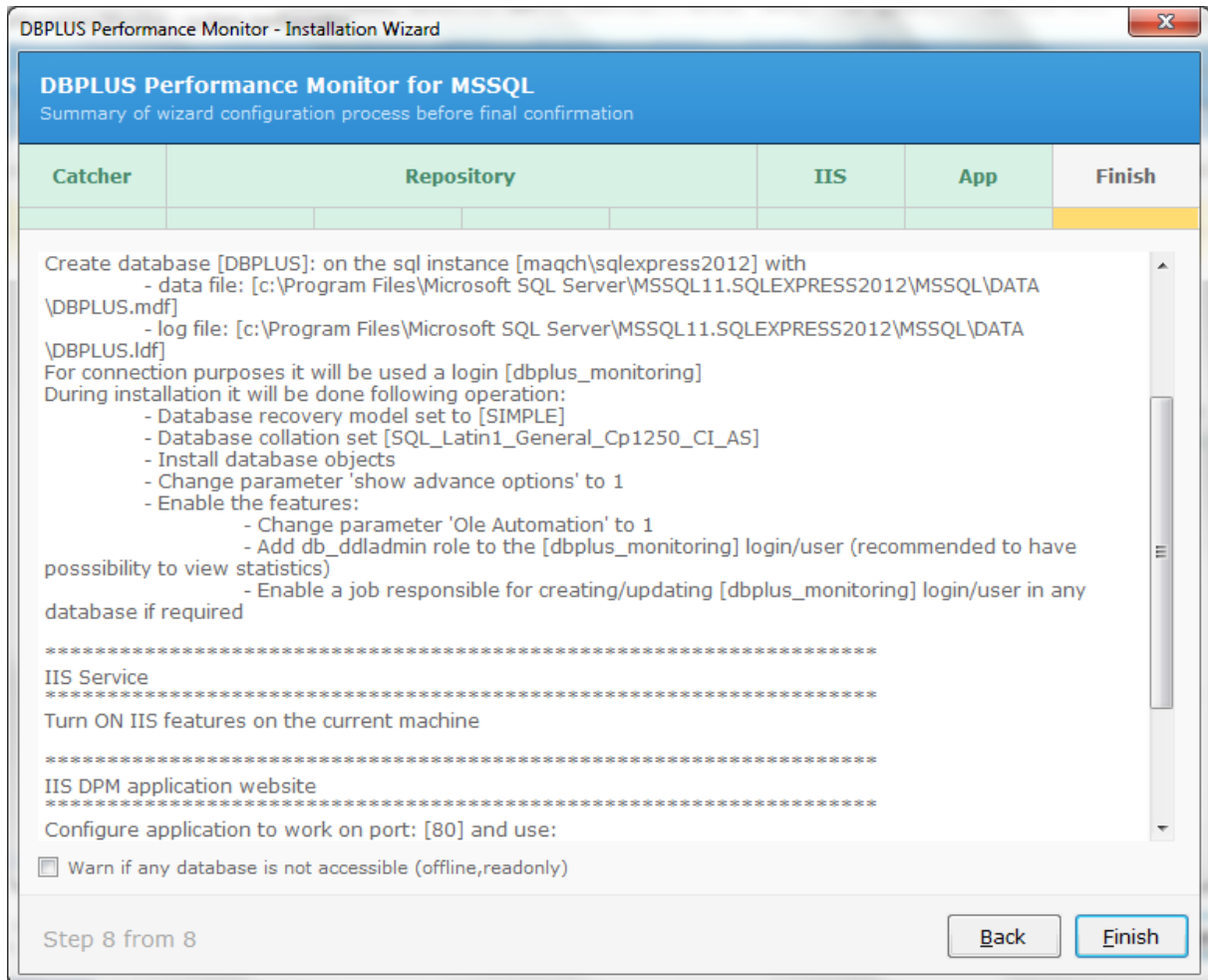
If the system will be running on port 80, link will be as follows:

<http://servername/dpm>

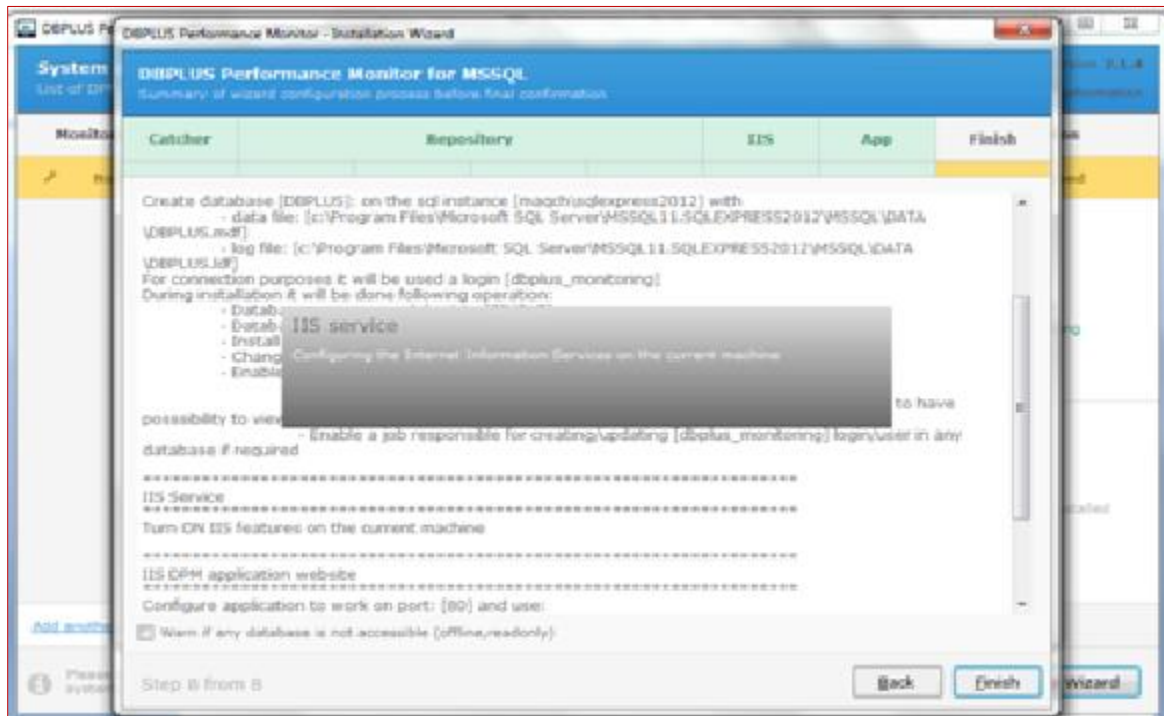
Click on the **[Continue]** button to proceed the next step

2.6 Configuration summary

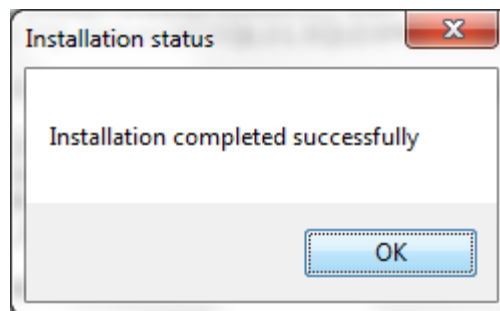
The last configuration step is to confirm all settings according to steps defined in the configurator. The final screen shows a summary. To confirm changes, click on **[Finish]** button.



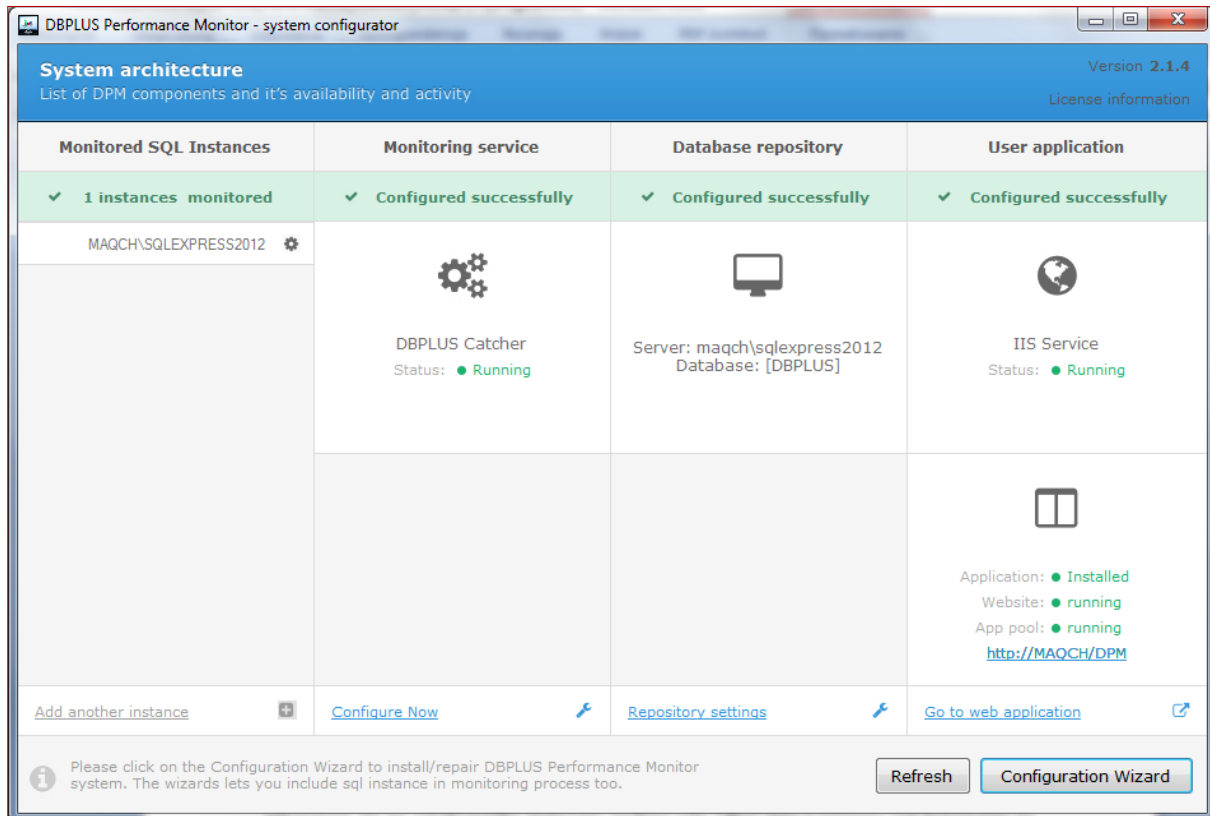
In the meantime, information about the progress of work is displayed



At the end of the completion status of the installation/configuration



As a result, system configuration main window looks like this:



From the above sample screen, we can read that DBPLUS system PERFORMANCE MONITOR is:

- installed on the server MAQCH (link to the application in the lower right corner)
- all components are properly configured (the bar with information “Configured successfully”)
- appropriate services are running:
 - DBPLUSCATCHER - a service responsible for database monitoring
 - IIS, Website, App pool - which means that the application is available to the user
- We have 1 monitored SQL Server instance
- Information from the monitoring of all instances (currently one) are stored in server SQL maqch\sqlexpress2012 in DBPLUS database.

Interface / User application is available at <http://MAQCH/DPM>

2.7 System configuration file

The default path of all system configuration files is:

C:\ProgramData\DBPLUS\DPM.Web.

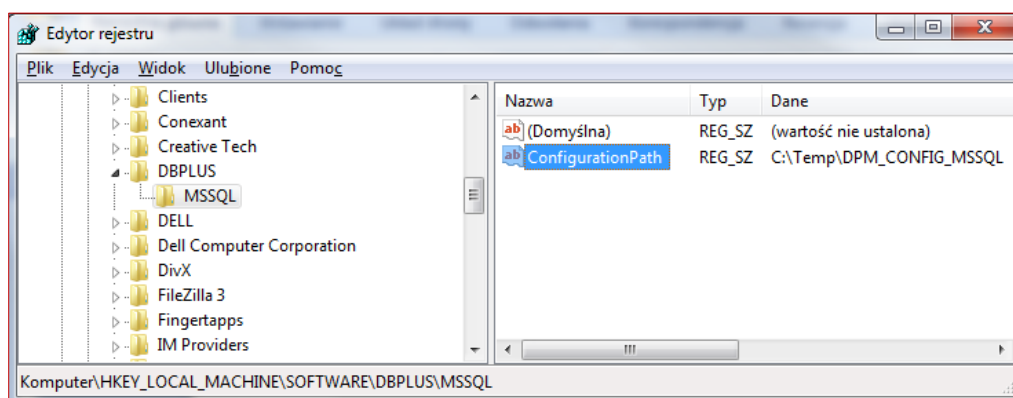
The folder contains:

- ConnectionStrings.con configuration file - connection information for the SQL instance with the repository database
- license file with * .license extension
- the DbplusCatcherServiceErrors.txt log file

In the system it is possible to change the configuration path. User can change this setting in the Windows registry on the machine with the DBPLUS software.

If you change the configuration folder, create the following path in the Windows SYSTEM registry: HKEY_LOCAL_MACHINE -> SOFTWARE -> DBPLUS -> MSSQL

In the defined key, create a ConfigurationPath string in which we give a new directory - below is an example:

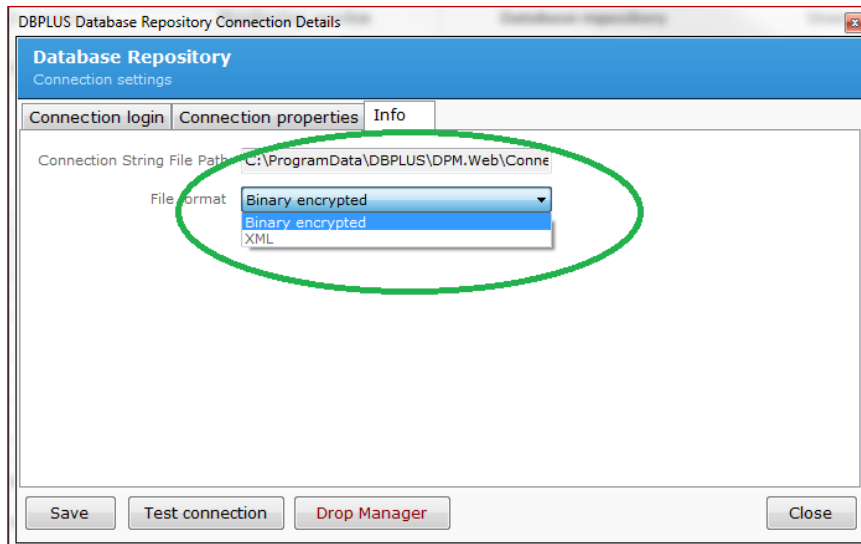


After change, move the indicated configuration files to the new location, restart the DBPLUS configuration Wizard and restart the system services (DBPLUSCATCHER, IIS).

2.7.1 System files for SQL Instance Repository

Configuration information to connect to an instance containing a DBPLUS repository is available from the DBPLUS Configuration Wizard, by clicking the link Repository settings.

As a result, a form appears where data is available to connect to the SQL instance. In the Database Repository form, you can save the file in XML format:



Below is an example of the ConnectionStrings.con configuration file in this format:

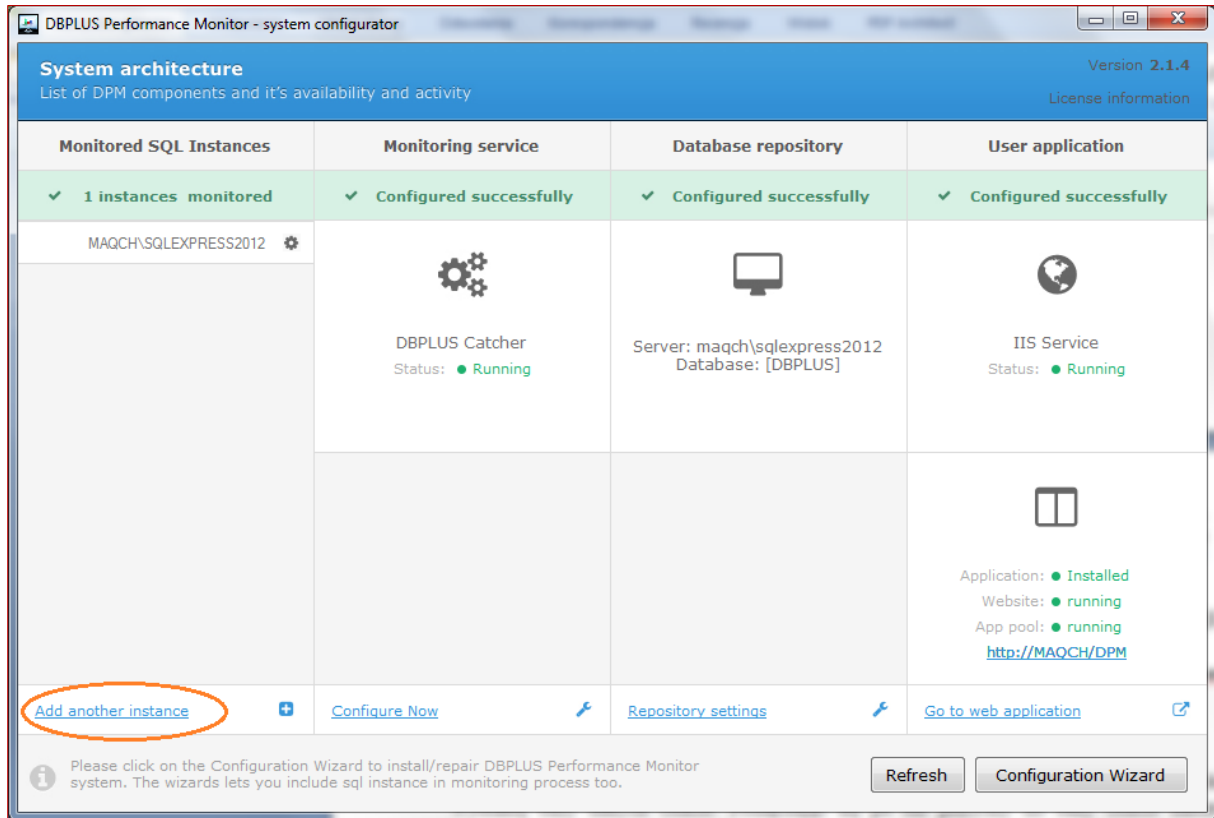
```
<?xml version="1.0" encoding="utf-8" ?>
<ConnectionString xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ConnectionName>DBPLUS MASTER</ConnectionName>
  <ServerId>1</ServerId>
  <ServerName>magch\sql2008</ServerName>
  <TCPPort>1433</TCPPort>
  <DatabaseName>dbdb1</DatabaseName>
  <ApplicationName>DBPLUS Performance Monitor</ApplicationName>
  <ConnectionTimeout>0</ConnectionTimeout>
  <UseSSL>false</UseSSL>
  <TrustServerCertificate>false</TrustServerCertificate>
  <UseReplication>false</UseReplication>
  <WindowsAuthentication>true</WindowsAuthentication>
  <SQLServerAuthentication>false</SQLServerAuthentication>
  <Encryption>XmlText</Encryption>
  <UserId/>
  <Password/>
  <UseConnectionPooling>false</UseConnectionPooling>
  <MinPoolSize>0</MinPoolSize>
  <MaxPoolSize>0</MaxPoolSize>
  <ConnectionLifeTime>0</ConnectionLifeTime>
  <PathToMDFFile/>
  <Index>0</Index>
  <IsMaster>true</IsMaster>
</ConnectionString>
```

It should be noted that the file is not encrypted and the field with the password is explicitly given (refers to the case of using the SQL login).

The option of changing the format of the configuration file is added in case external programs change the connection settings.

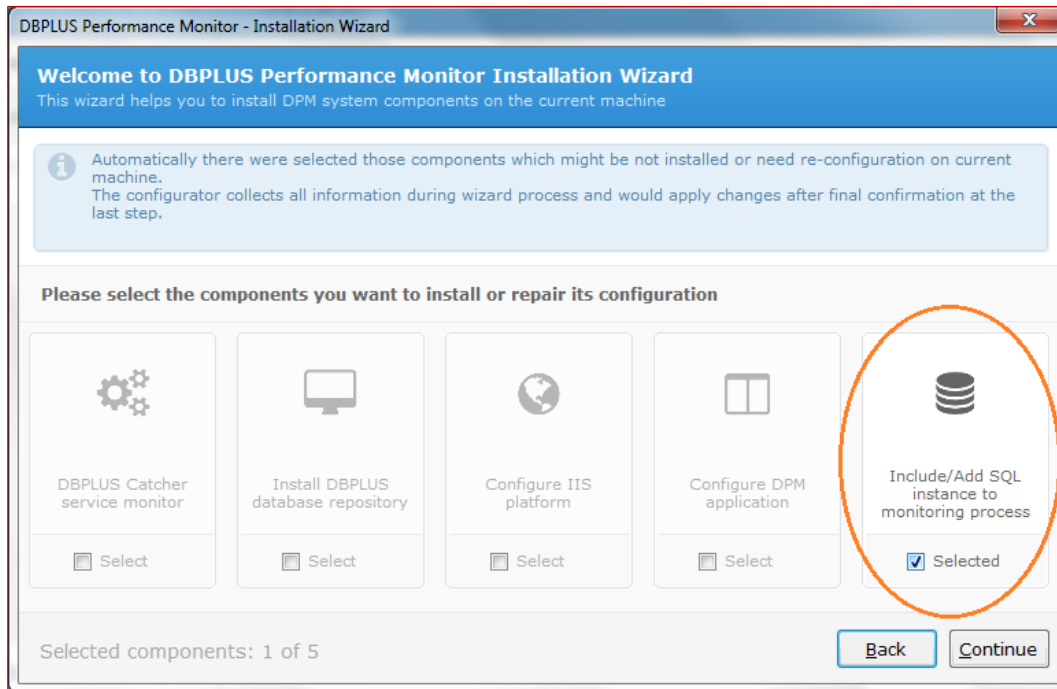
3 Adding a SQL Server instance for monitoring

After the initial configuration, you can proceed to add more SQL Server instances for monitoring. For this purpose, in the main System Configurator Window we click **[Add Another instance]** button.



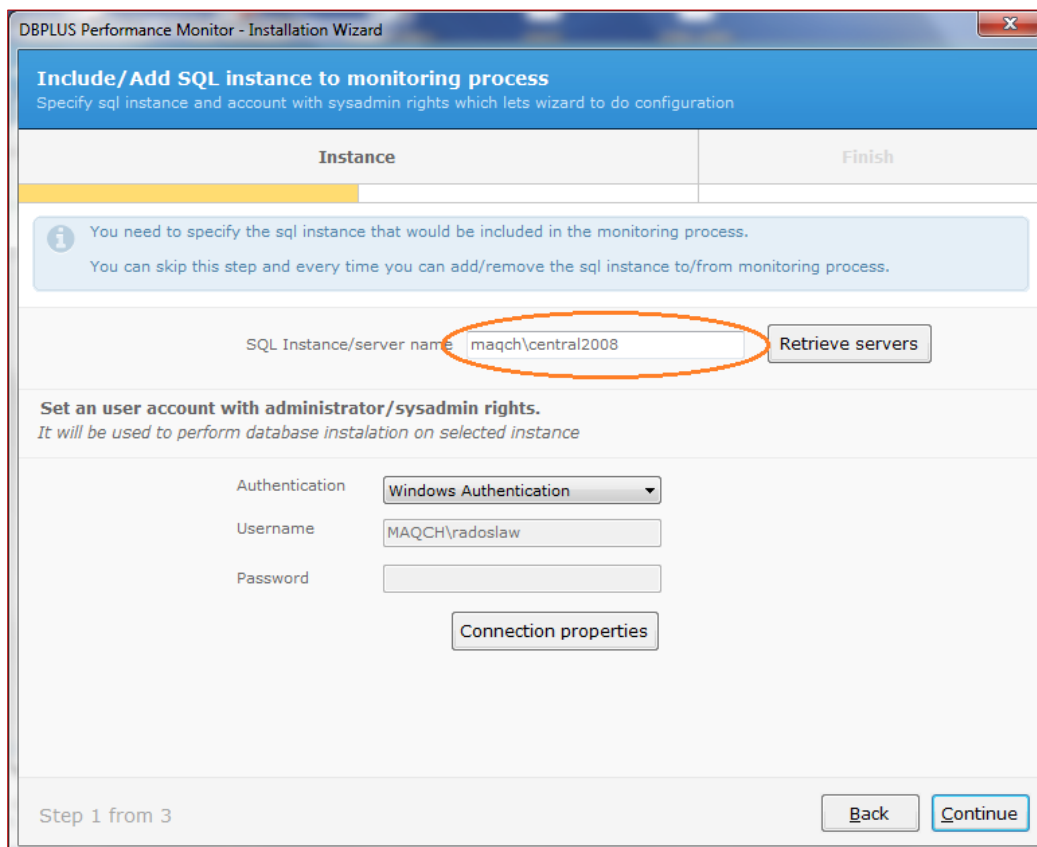
IMPORTANT: If the **[Add another instance]** button is not available it is the result of a license for a specified number of instances.

The second option to add an instance is clicking **[Configuration Wizard]** button and select the component **[Include / Add SQL instance to monitoring process]**



As a result, we go to the wizard to add a new database. In the first place we provide basic information:

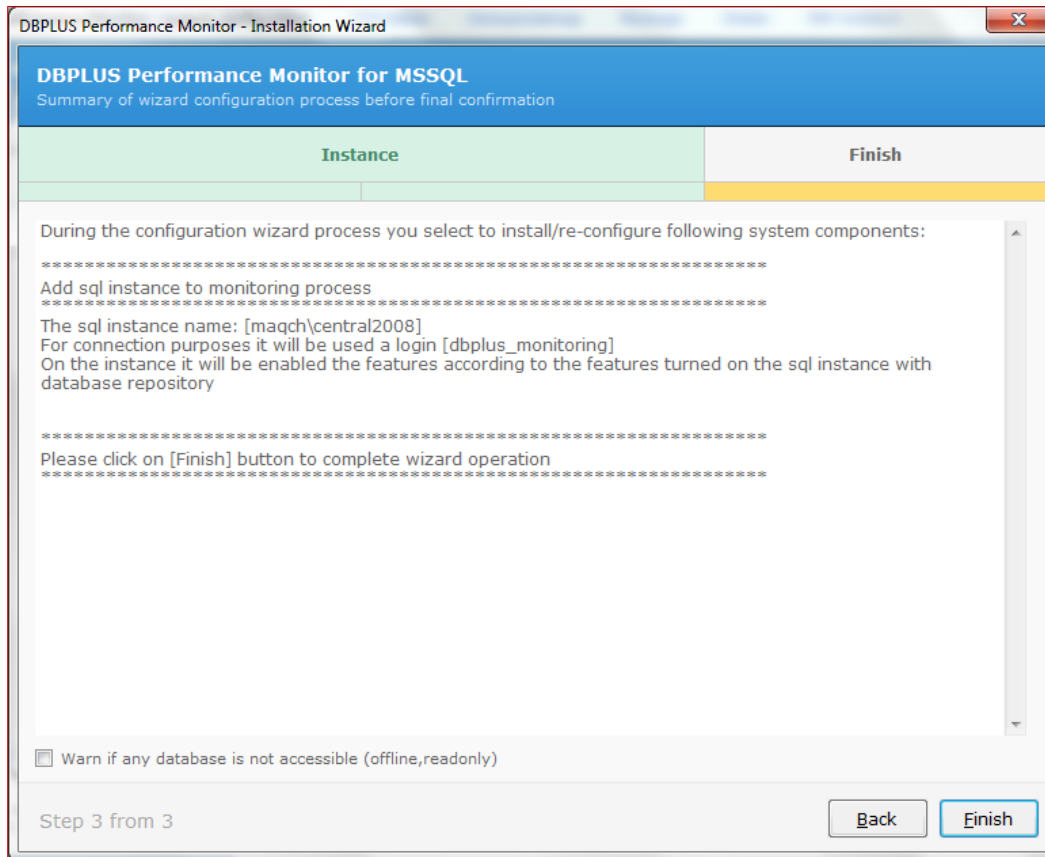
- Name of the SQL Server instance
- The name of database account with the privileges of **sysadmin**, which will be used to carry out the process.



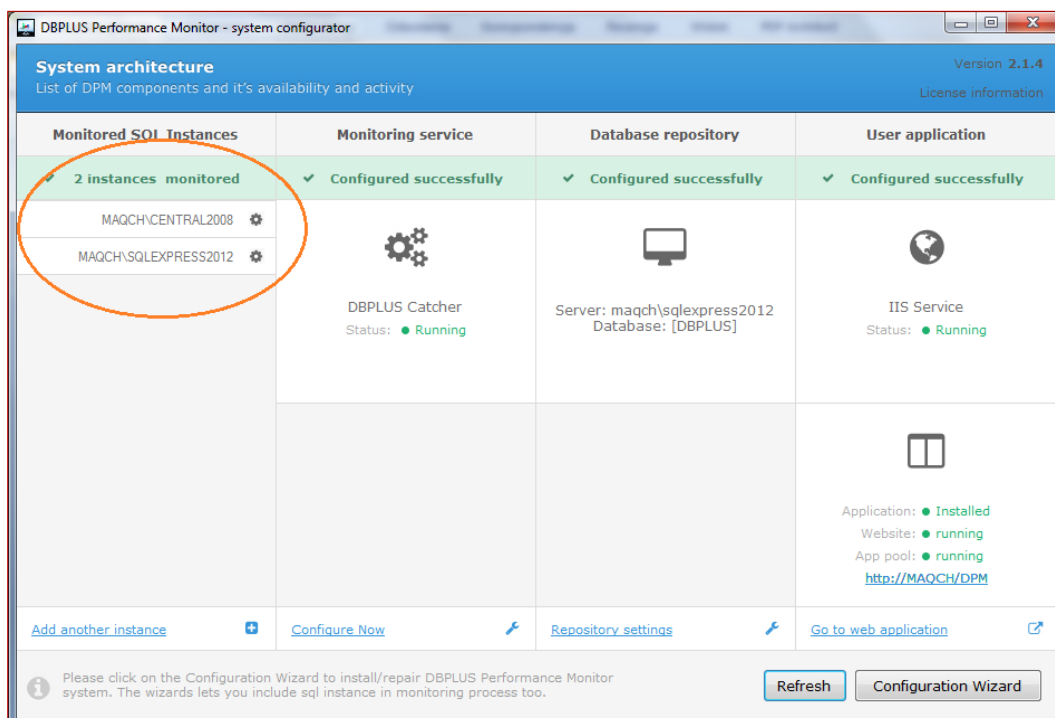
When you enter the instance name, you have the option to search for all SQL servers on the network - option **[Retrieve servers]**. In the case of custom properties for a connection, you can update them by clicking on the button **[Connection properties]**. In the **[Connection Details]** window we set parameter values, among others Connection Timeout, Application name, etc.

In the next step, you specify the account information, which will be used to connect to the database. By default, the wizard will prompt the login data used when configuring an instance of the repository database.

After clicking **[Continue]** button, it shows the final screen being the summary of previous steps.



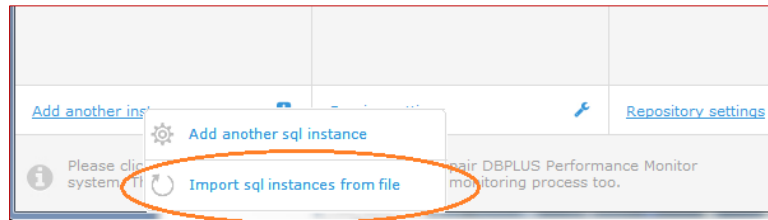
Click **[Finish]** button to add an instance to monitoring. As a result, changes are visible in the system configuration main window - DBPLUS Performance Monitor supports two SQL Server instances



3.1 Import SQL instance from file

From the DBPLUS Configuration Wizard console, it is possible to import sql instance configuration into the list of monitored objects.

To do this, in the main system configurator window, click the [Add Another instance] button - select [Import sql instances from file] from the pop-up menu.



As a result, the form appears:

The screenshot shows the 'Import SQL Instances form' for MSSQL. The title bar says 'Import SQL Instances form'. The main header is 'DBPLUS Performance Monitor for MSSQL' with the subtitle 'Import instances to monitoring list'. Below this is an information icon and a message: 'Please select a file to import sql instances connection details and include them in monitoring process. You can use csv file containing following columns: SERVERNAME, TCPPORT, DBPLUS_USERNAME, DBPLUS_PASSWORD, SYSADMIN_USERNAME, SYSADMIN_PASSWORD [,SERVER TYPE NAME]. File should be without header and with comma separator. For domain username, PLEASE USE EMPTY PASSWORD. IF SYSADMIN user is not specified, then system will do the steps in context of currently logon user.'

Below the message is a section 'Select a file' with a text input 'File for import' and a 'Select file' button.

Below that is an 'Option' section with four checkboxes:

- ☒ Create DBPLUS login/users if not exists
- ☒ Enable [OLE Automation] module to monitor disk spaces usage
- ☒ Add ddl_admin role to the dbplus user to view statistics
- ☒ Enable a job responsible for creating/updating DBPLUS user

Below the options is a table with the following columns: 'Marked For Import', 'Server Name', 'DBPLUS Username', 'DBPLUS Password', 'Create DBPLUS user', 'SYSADMIN Username', 'SYSADMIN Password', and 'Database'. The table is currently empty.

At the bottom right, there are two buttons: 'Test connection' and 'Import'.

The file format is as follows:

- The name of the SQL server
- Port number
- DBPLUS username
- Password for the DBPLUS user
- User name with SYSADMIN privileges,
- Password for a user with SYSADMIN privileges,
- SQL server type - optional value (The server type must be in the managed list from the web application level, i.e. in Configuration-> References lists)

Additional information:

- When using a domain user, the password in the file is empty
- File separator: comma
- We recommend selecting the proposed options:
 - Create DBPLUS login / user if not exists,
 - Add ddl_admin role to the DBPLUS user - the system is able to analyze objects from the execution plan,
 - OLE Automation module, allowing monitoring the server disk usage,
 - Incorporating the task responsible for the DBPLUS user and mapping with the login.

After loading the file, the system automatically checks:

- Possibility to connect imported SQL instance
- Whether the instance is already monitored and, for example, does not require import
- Whether the DBPLUS login specified for use with the connection exists in the monitored instance or not

If the above criteria are met, the system automatically selects specific instances for import - below is an example:

Marked For Import	Server Name	DBPLUS Username	DBPLUS Password	Create DBPLUS user	SYSADMIN Username	SYSADMIN Password	Database
<input type="checkbox"/>	maqch\sqlxp...	db_mon987	db_mon987	<input type="checkbox"/>	maqch\radosl...		master
<input checked="" type="checkbox"/>	maqch\sqlxp...	db_mon987	db_mon987	<input checked="" type="checkbox"/>	maqch\radosl...		master

After moving the table slider to the right hand side, we have additional information:

TCP Port	Server Type	Connection Status	Import status
1438	TEMPORARY	Wystąpił błąd związany z siecią lub wystąpieniem podcz...	
1433	TEMPORARY	connection success (db_mon987 to create)	

After clicking the **[Import]** button, the selected instance has been added for monitoring.

4 System Upgrade

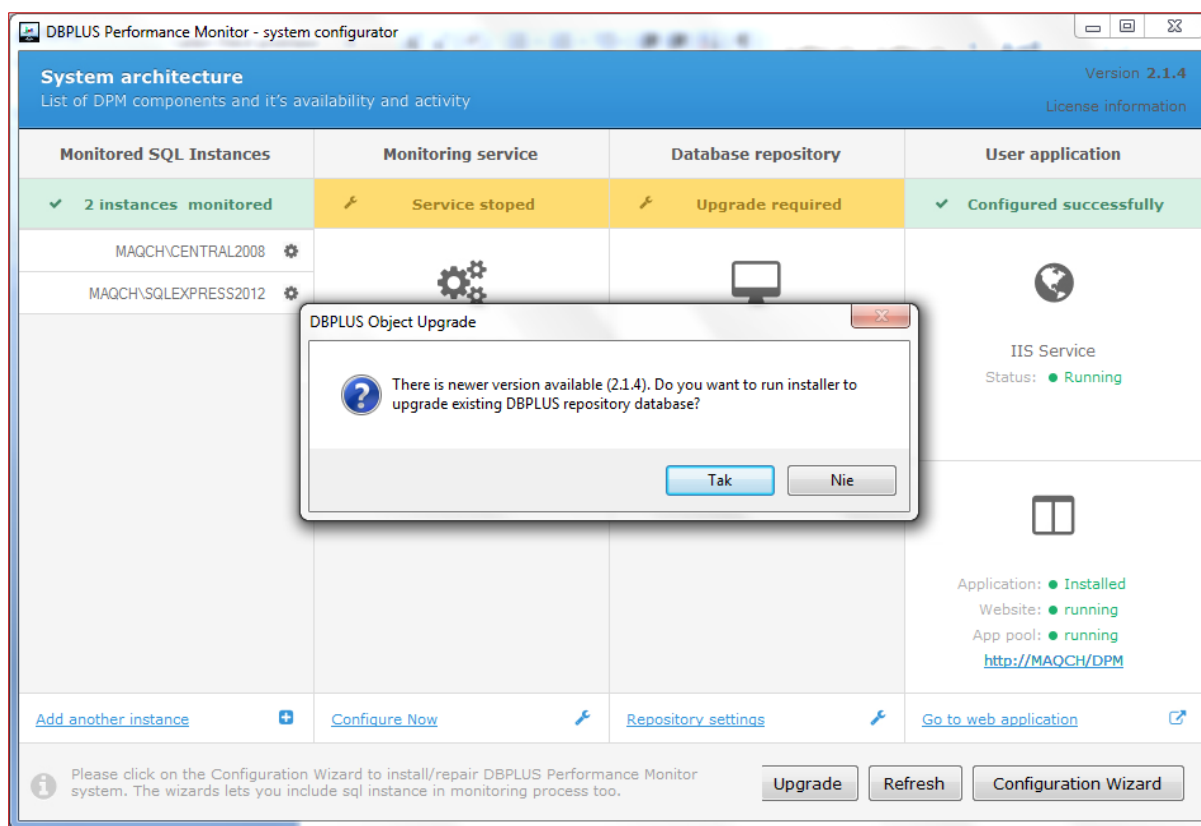
Technical support provides the access to new software updates that are published 4 times a year, as well as to DBPLUS engineers' help in SQL Server servers diagnosis, using **DBPLUS Performance Monitor™** software.

Upgrading system combines with two steps:

- Run the installation file (which goes the same as the first installation)
- Upgrade of database objects repository on DBPLUS user to the latest version

4.1 Setting up for the latest version

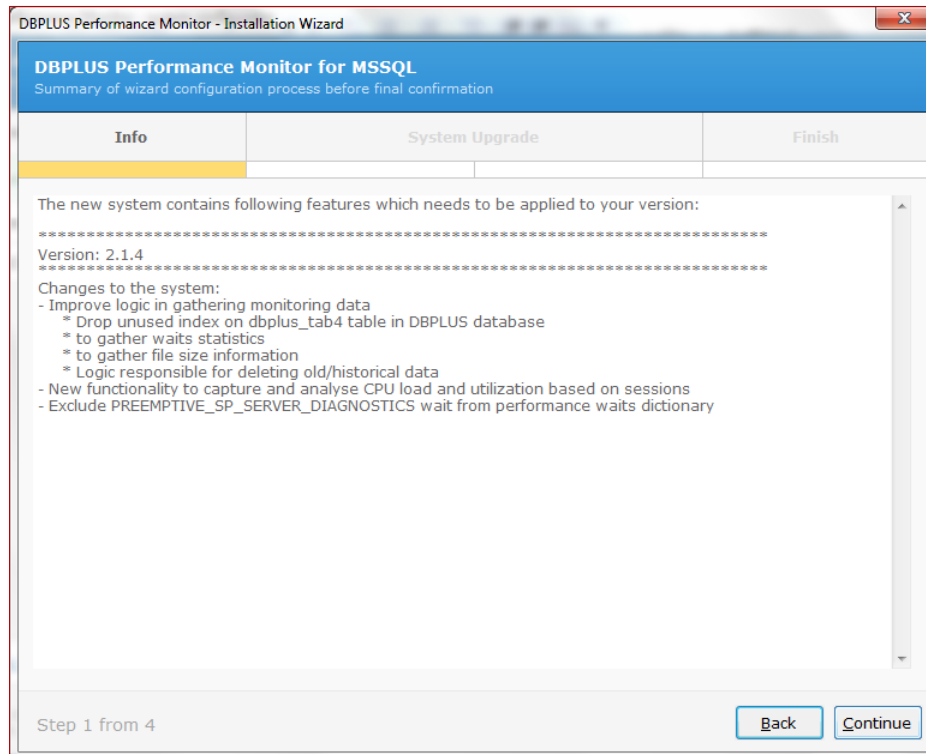
In order to go through the upgrade process, you have to run DBPLUS Configuration Wizard, which also runs automatically after installation. In the result:



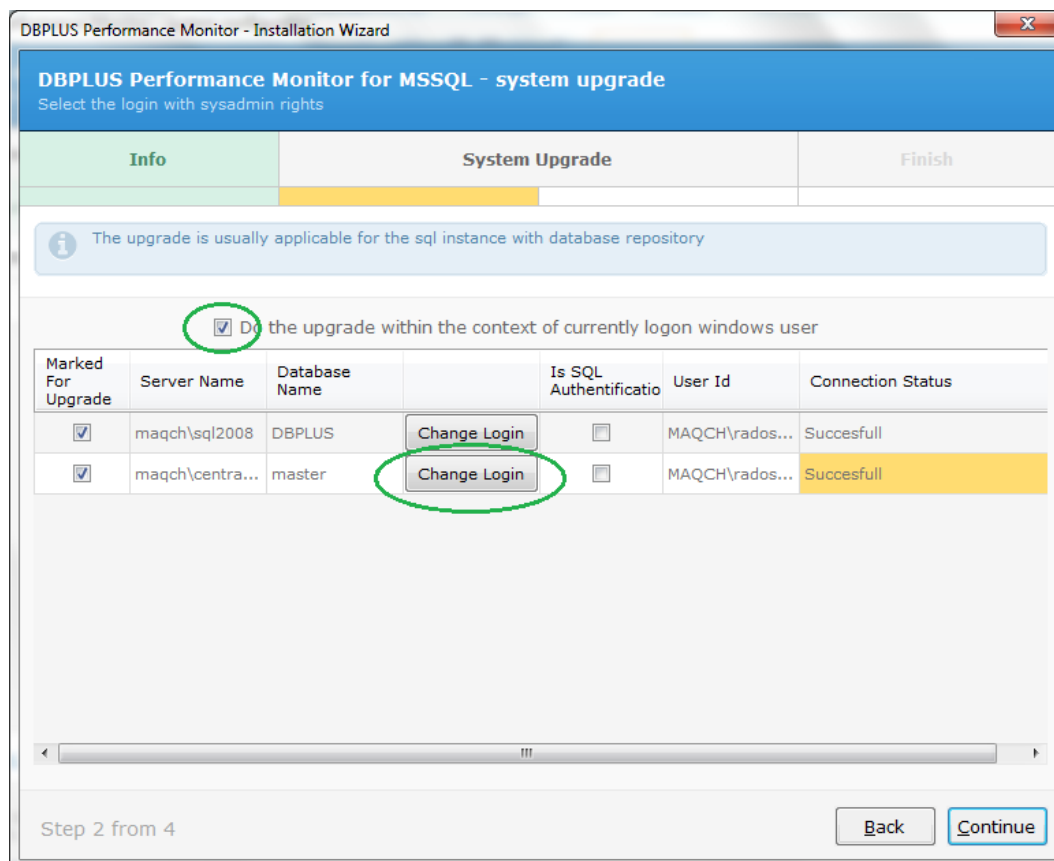
System automatically detects the need to update to the latest version. We accept the dialog box and we run the wizard that will guide us through the upgrade process.

In case of withdrawal from the operation we can always return to it by clicking **[Upgrade]** in the main configurator's window.

As the first screen we have information about system version, to which application will be updated (with system functionality description):

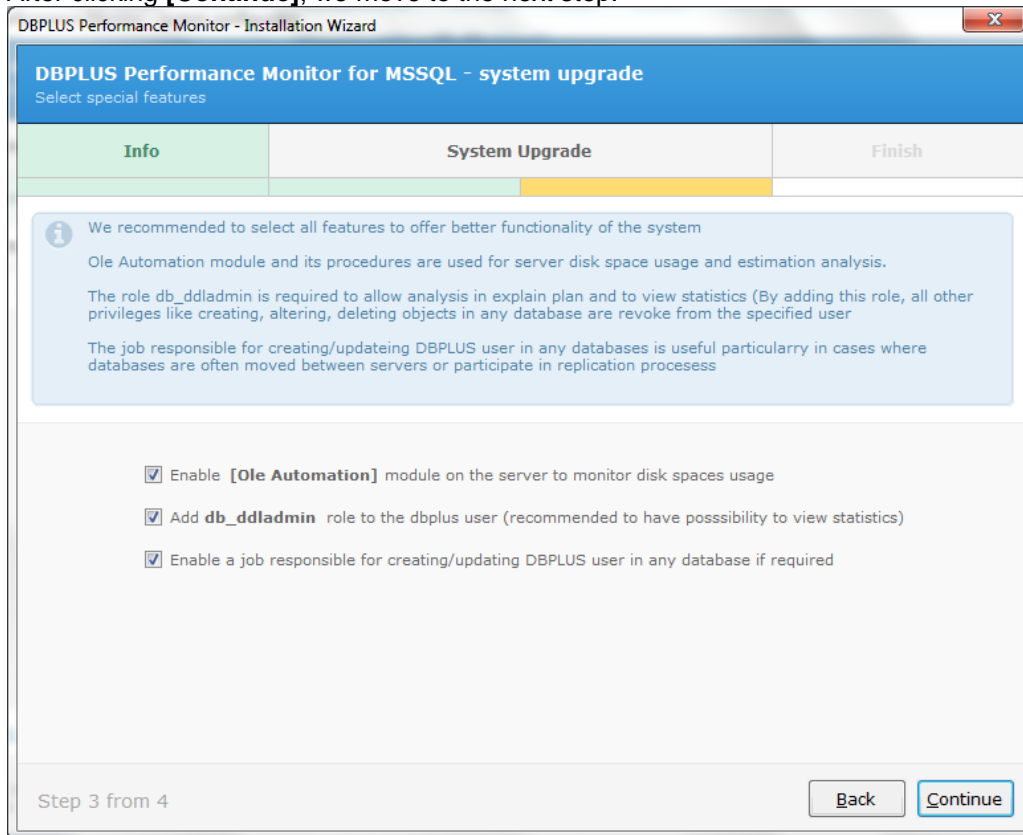


In the next screen you see the list of monitored instance for which we provide a user with **sysadmin** rights. In the most common case, you should tick the box **[Do the upgrade within the context of currently logon user]**. If the current domain /Windows account does not have the appropriate permissions, you must change the selected instance by clicking **[Change login]**.

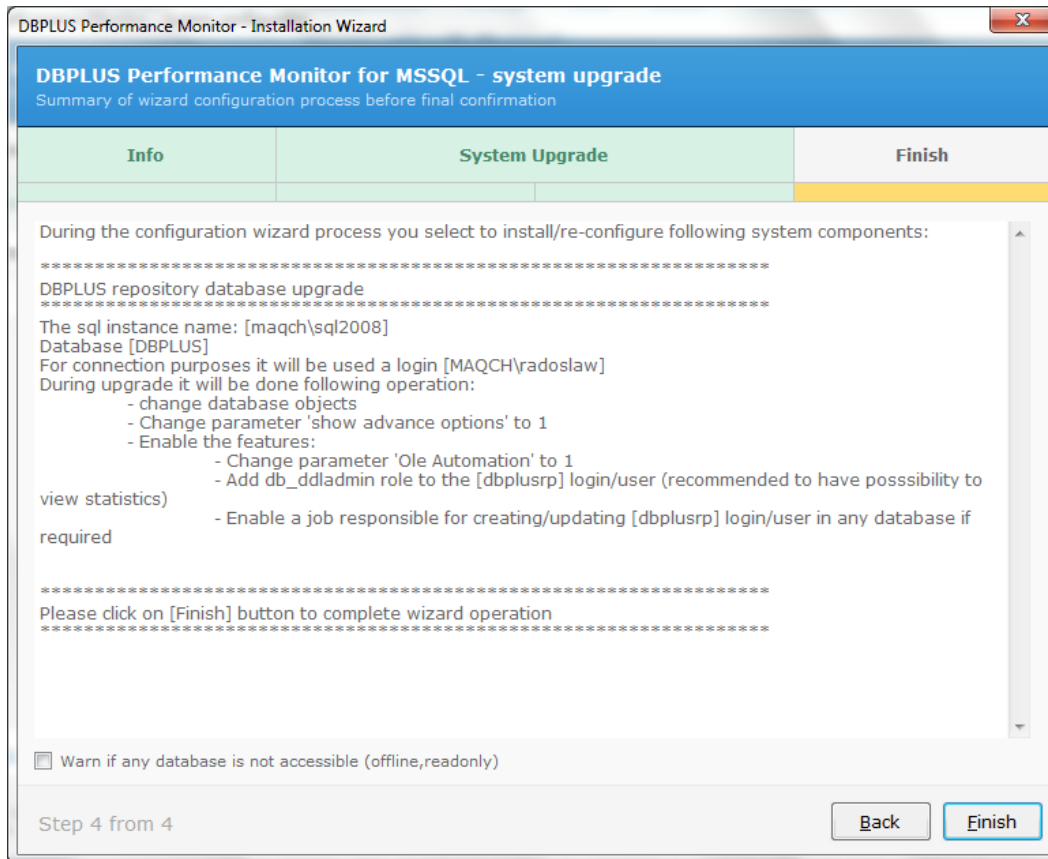


Upgrade procedure applies only to objects of the instance on which there is the DBPLUS repository database. In specific cases, may need updating user permissions DBPLUS used when connecting to the monitored instances.

After clicking **[Continue]**, we move to the next step.

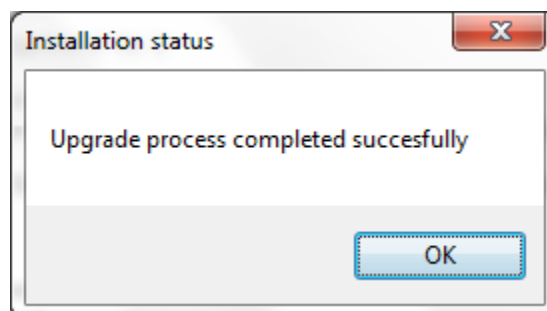


The system remembers what additional features are enabled in the system. We recommend that you leave these options enabled. Go to the next - the last step in the procedure.



System informs about the operations, which will perform at the instance of the database containing DBPLUS repository. We accept by clicking **[Finish]**.

Depending on version of the update process can take 1 to 3 minutes. At the end we get the information about success of the process.



We close the configurator's window.

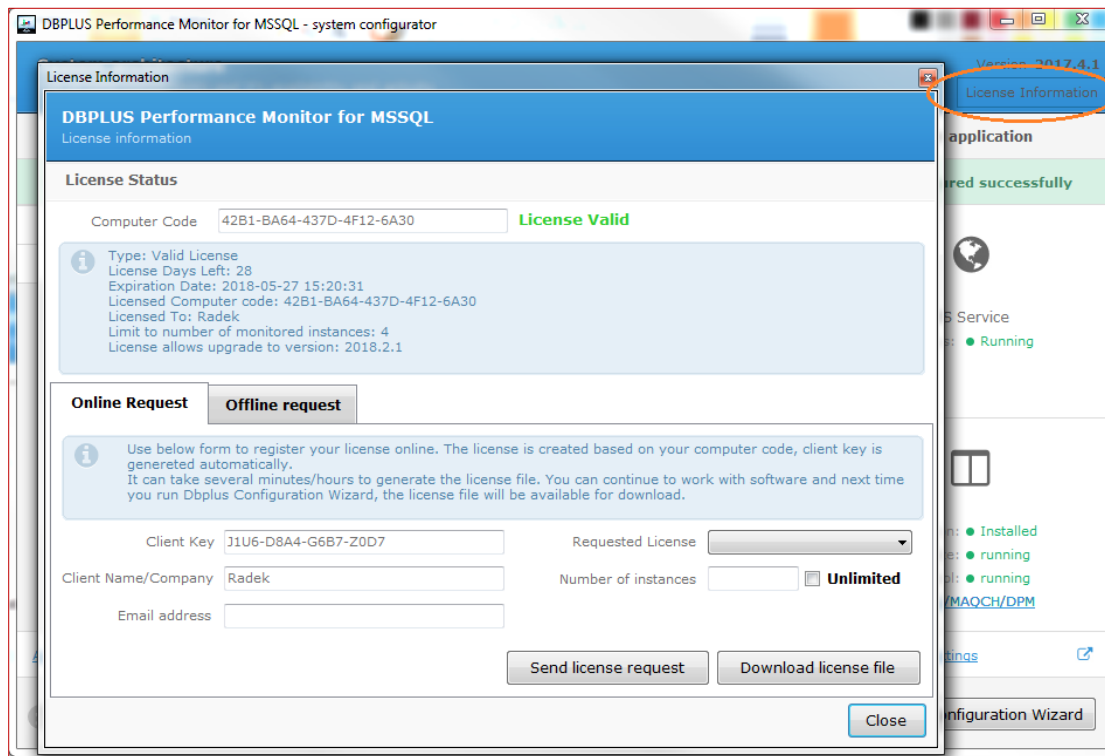
5 License

The license is generated always for the server with **DBPLUS Performance Monitor™** software and not for the database.

Standard system license includes:

- System uptime
- The number of monitored SQL Server instances

Information about licensing is available from the configurator, i.e. **DBPLUS Configuration Wizard**

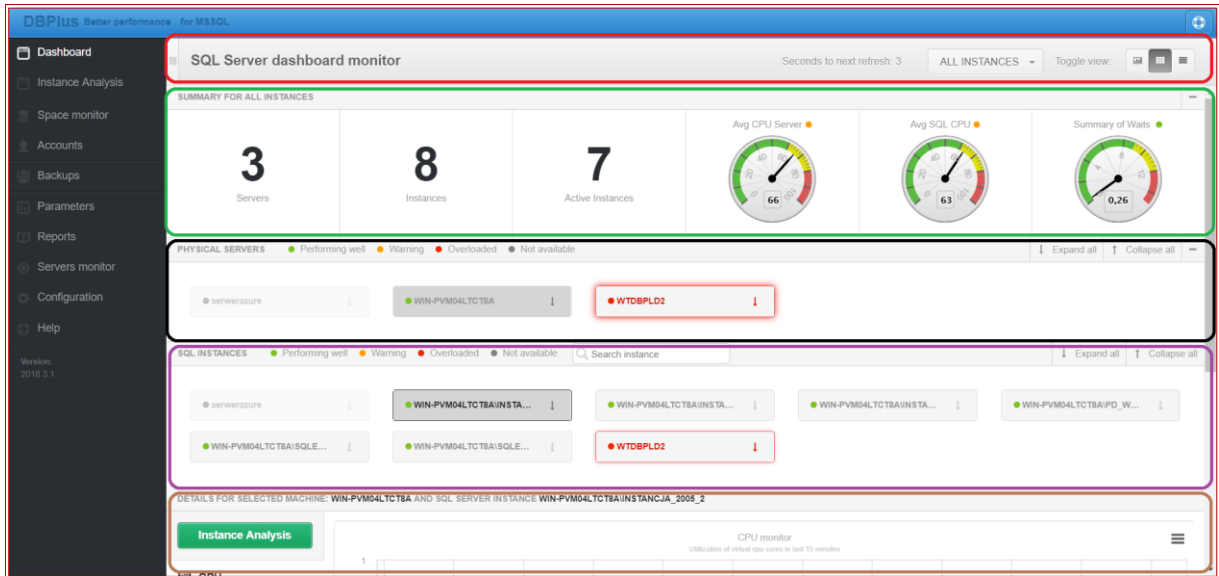


6 Working with program

The user interface is accessible from a web browser at the previously configured address. The default page of the system is a dashboard showing the current performance of the monitored databases.

6.1 Dashboard

After starting **DBPLUS Performance Monitor™** web application, it opens a dashboard showing the current performance of monitored SQL Server instances. Default starting page is a dashboard showing the currently monitored SQL Instances and how they perform.



Dashboard is divided into the following areas:

- information banner
- summaries area
- servers' area,
- instance area,
- details for the selected SQL Server instance.

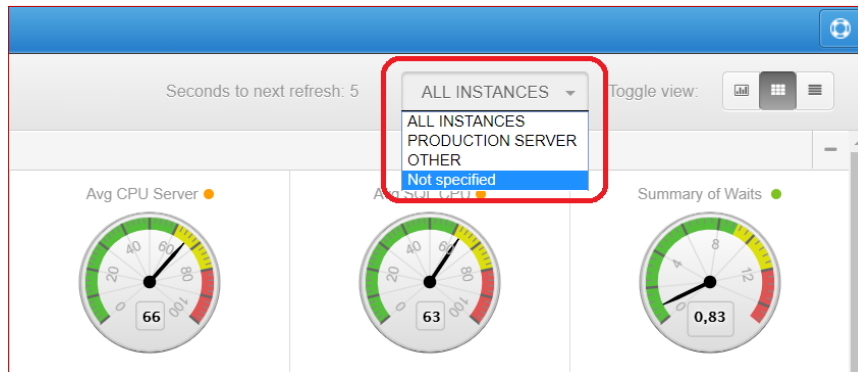
6.1.1 Information bar

User can switch between dashboard and different mode using the information section. List of available modes:

- **Icon View** - displays monitored servers / SQL instances as icons (default)
- **Grid View** – SQL instances are displayed in a grid/ table view
- **Television Mode** - shows SQL instances in the form of developed icons with automatically switching performance indicators.

Additionally, User is informed how much time is left until next dashboard refresh with the most recent data about the current performance of all monitored instances.

Users can change display information's about SQL instances via the drop-down menu in the information bar. Various types of SQL Instance can be defined and attributed in the 'Configuration' menu, which is described in the next chapter.

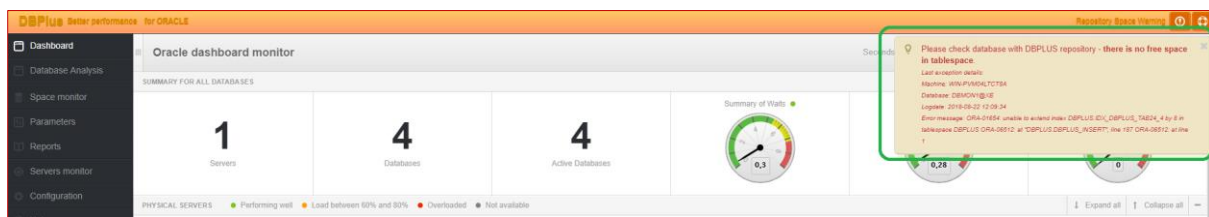


In case when information bar changes colour from blue to orange – this implicates insufficient space in the DBPLUS repository base or that DBPLUSCATCHER service doesn't work.

Insufficient space at repository database

In the instance of space in the pattern database, which repository is intended for DBPLUS to collect data, a warning message will show.

The toolbar on Dashboard page turns orange and a lack of space message „Repository Space Warning” about lack of space will be shown.

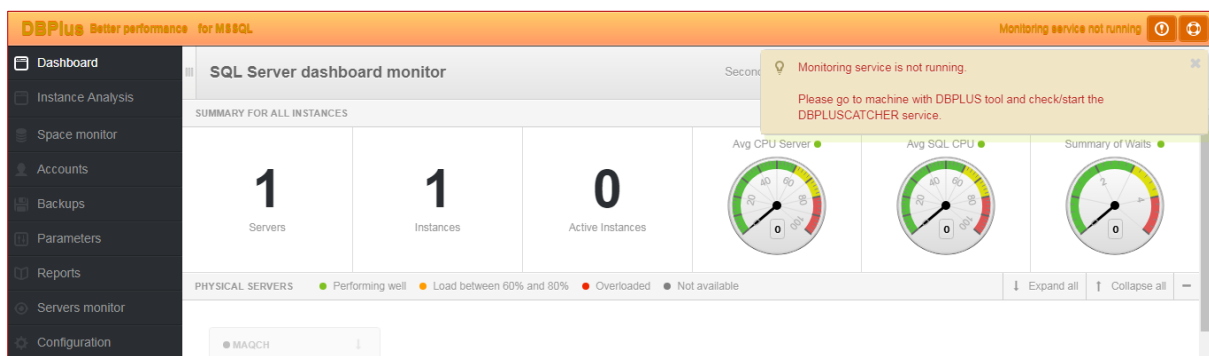


Information about the error is stored in the Logs tab. If in the database Repository there is no free space, the logs are stored in the DbplusCatcherServiceErrors.txt file in the C: \ ProgramData \ DBPLUS \ * folder on the server with the DBPLUS software.

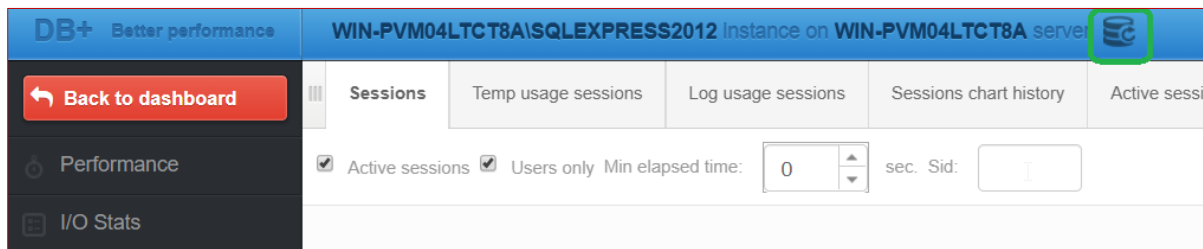
DBPLUSCATCHER service is not running

In the event of monitoring service problem, the toolbar on Dashboard page turns orange and information „Monitoring service not running” will appear.

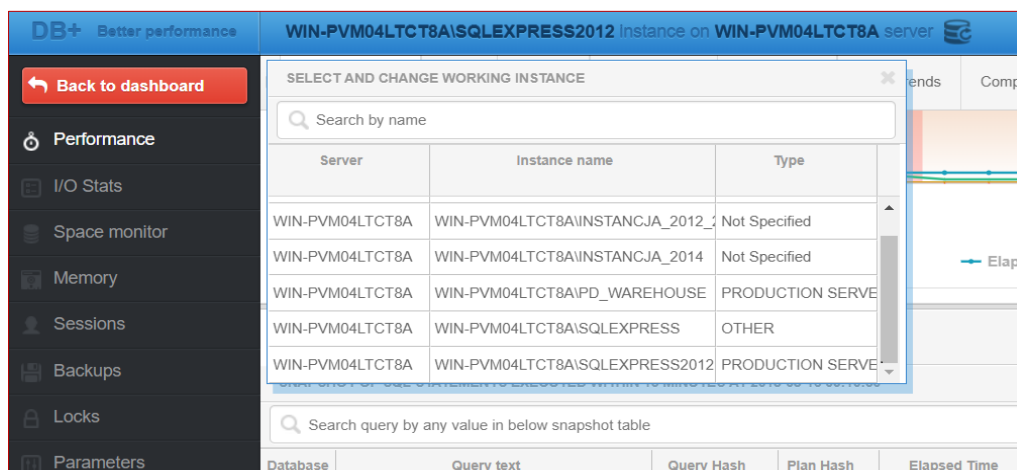
To fix the problem you need to check for any issues with the server on which the client DBPLUS is installed, after which restart the DBPLUSCATCHER service.



In addition, the information bar after entering the module of the Performance system by Instance Analysis presents the identifiers of the selected SQL instance for each screen.



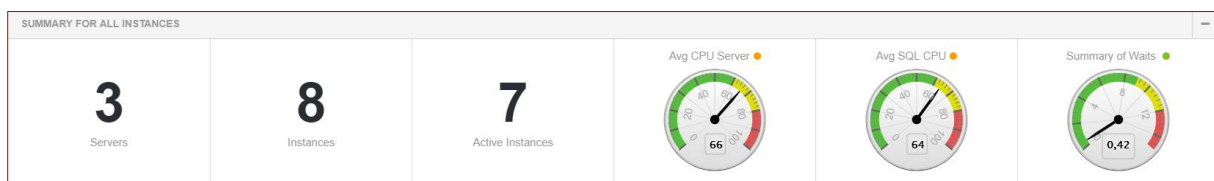
As a result of clicking on the "database" icon, a table with a list of monitored SQL instances is shown, clicking on other instances from the list changes the view to the indicated instance, while remaining on the given screen.



6.1.2 The summary area

The main area presents a general summary of:

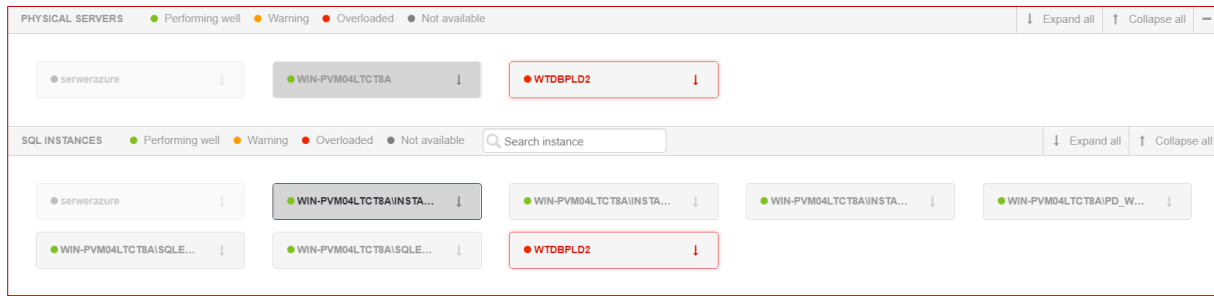
- number of monitored servers and instances
- number of active instances
- number of databases on all instances
- current load of virtual processors
 - by all processes on server
 - by all SQL instances
- summary of waits



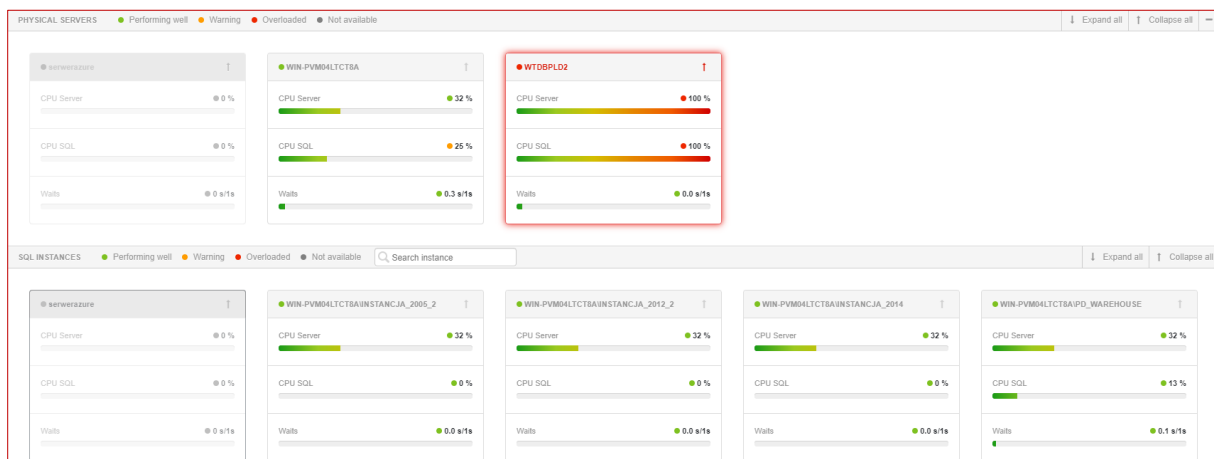
From the main area, you can already determine whether utilization of virtual processors of all servers comes from the SQL instance, or, in turn, is caused by other non-SQL processes running on machines.

6.1.3 Servers and instances area

In server's area, we see the icon of servers running SQL instance. After clicking on the server, in the area below these SQL Server instances will illuminate which work on the machine.



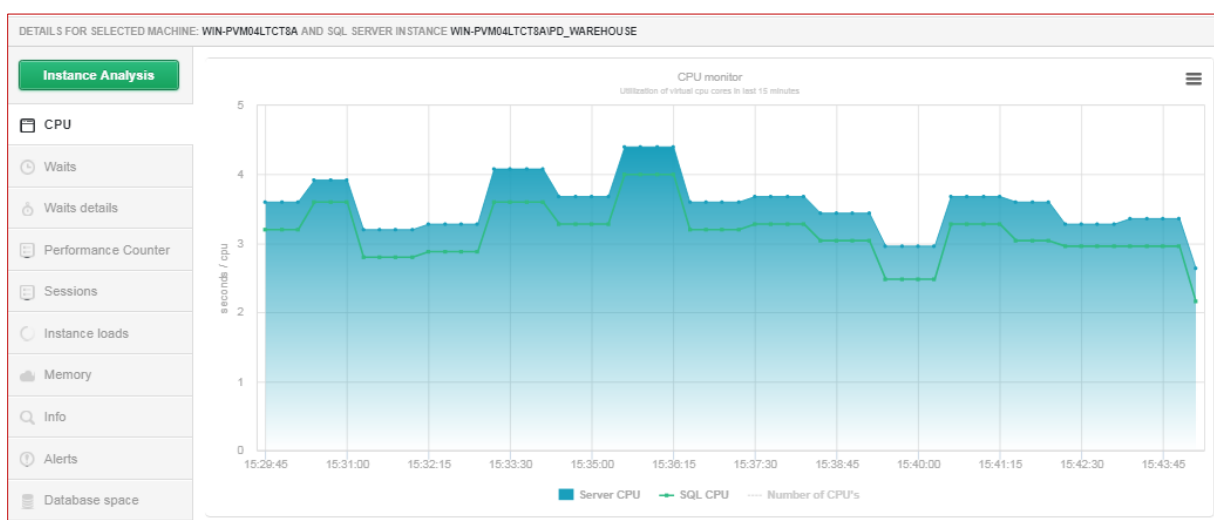
Icon on each server or database can be expanded by clicking on the "arrow" button or **[Expand All]** button



After clicking **[Expand All]** button at the level of SQL instances area, we see exactly which SQL instance has the highest level of waits.

6.1.4 Details of SQL instance performance

In order to analyze the current load, click on the icon of a particular SQL instance. As a result, the lower dashboard area reloads presenting details of the selected SQL instance.

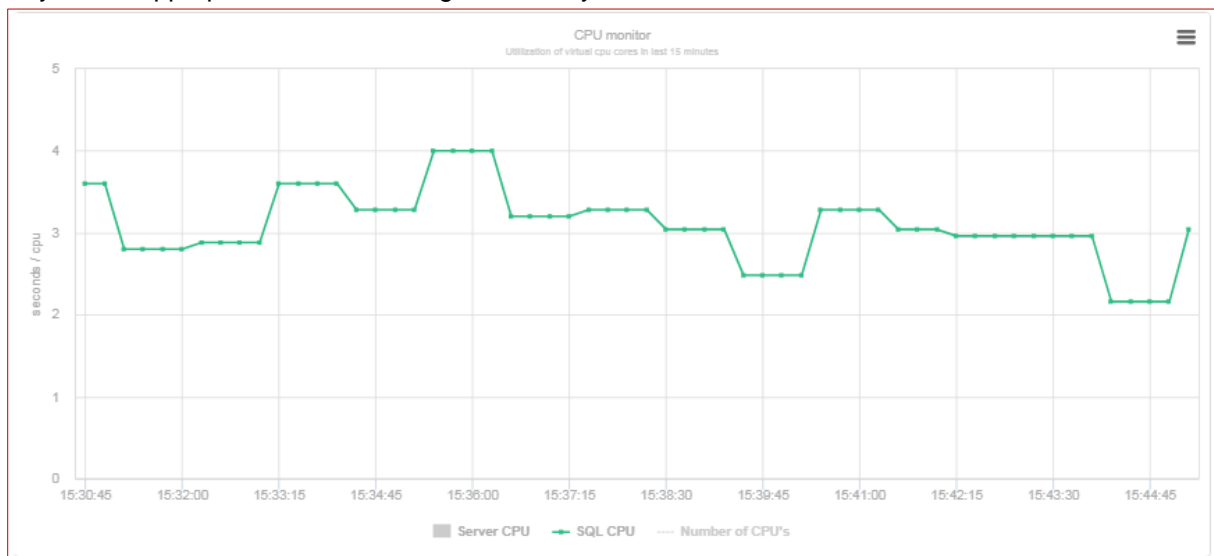


Here dashboard lets to:

- observe the current CPU usage

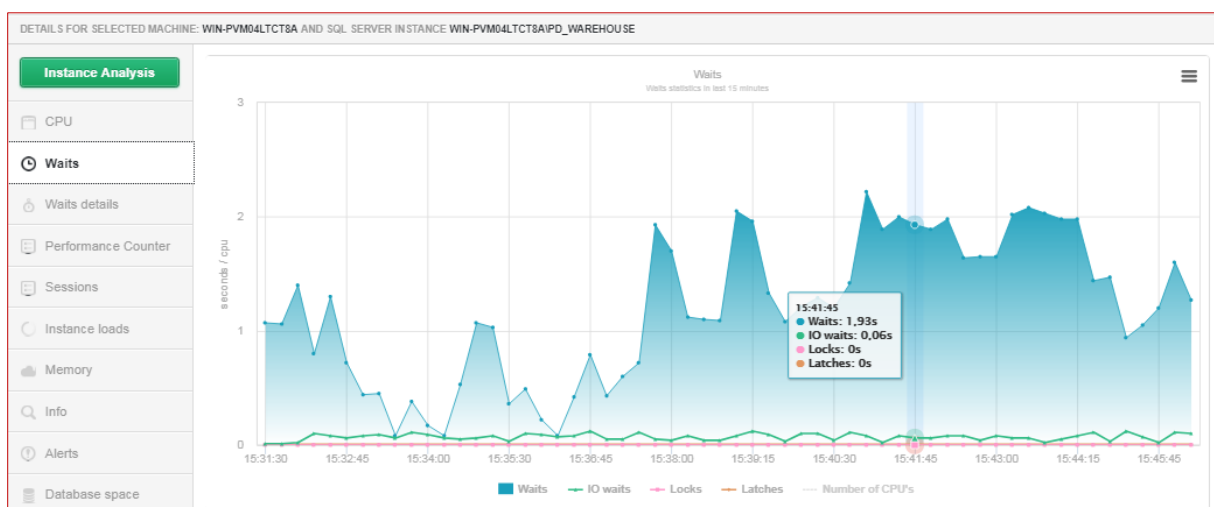
- determine on what SQL instances currently spends his time – **Waits bookmark, Waits details bookmark**
- analyze performance indicators of the last 24 hours - **Performance Counters**
- check the level of the session / locks, active transactions - **Sessions**
- to see the load of the instance from the last 24 hours - **Instance Load**
- verify memory utilization - **Memory**
- check the size of SQL Instances - **Info**
- view basic information about instance - **Info**
- check alerts - **Alerts**

Information about CPU utilization, Waits, sessions are presented here in the horizon of the last 15 minutes. For example, on the CPU load chart – after clicking in series **[Server CPU]** it remains active only series appropriate for utilization generated by an SQL instance.



In the current example, we see that the instance of SQL used a level 3-4 CPUs.

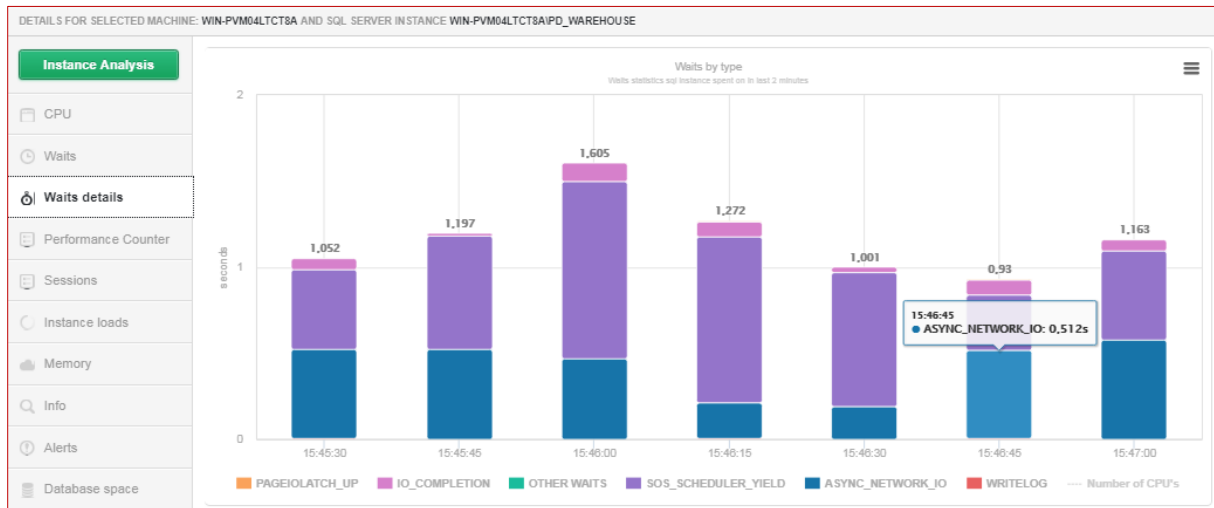
Chart means that at a certain point in time (the time read from the X-axis), all users (active sessions) are waiting for the outcome of the query indicated number of seconds (the result read from the Y axis). Categories IO waits, Locks, Latches help to state why sessions are staying idle. By default, all series are visible.



Waits are divided into following types:

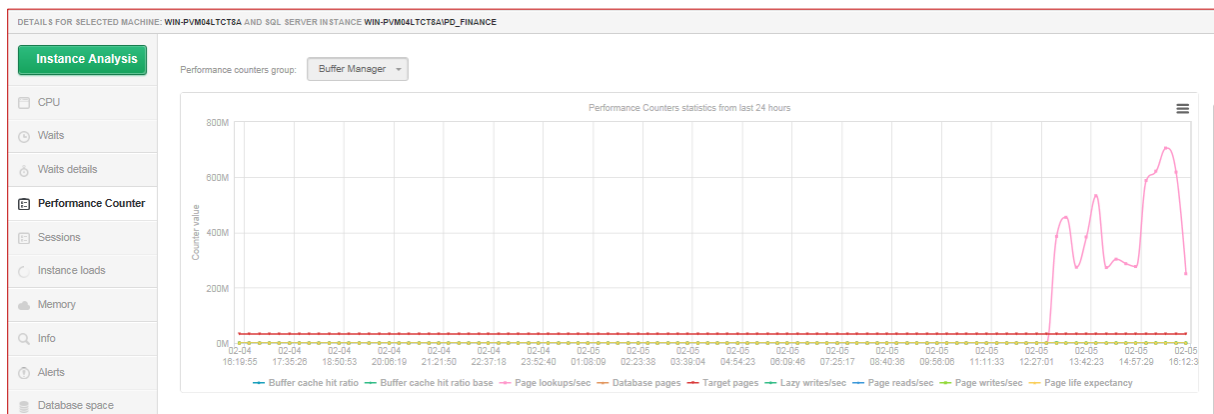
- I/O waits - readings of disk devices
- Locks - locks of database records by session
- Latches - waits for access to database buffers

Data information about waits can be known in the next tab **[Waits details]**.

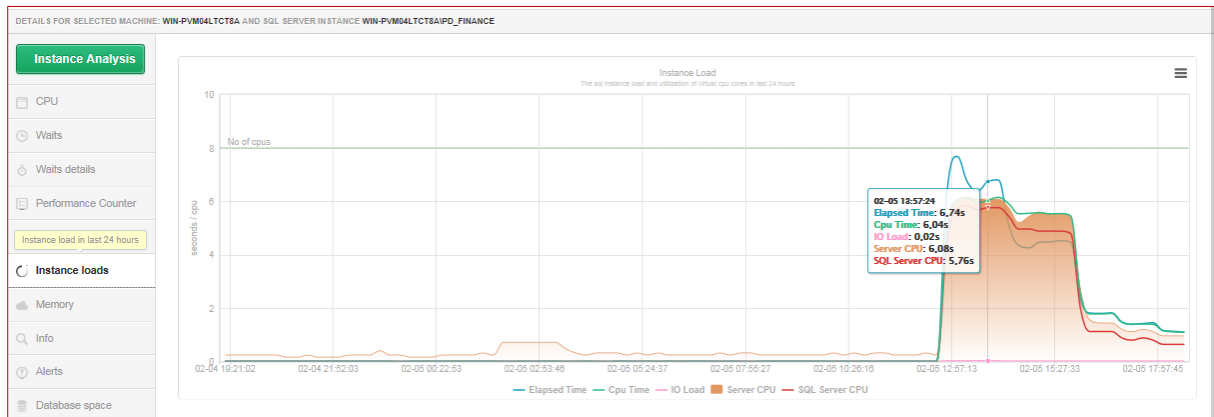


On the dashboard screen for the selected SQL service, you can also check the load, SQL indicators - so called performance counters and memory utilization of the last 24 hours.

To do this, click on the appropriate tab on the left side. In the **Performance Counter** we should be aware of the additional selection filter of the group:



The load from the last 24 hours we get after clicking on **[Instance Load]** button.



Instance Load is one of the core modules used by DBPLUS engineers to analyze performance. Chart consists of the following series:

- **Elapsed Time** - shows the summary time all users are waiting on the result of a query at a given second of time. On the graph for the displayed point Elapsed Time is 6.74 seconds, which can be interpreted as follows:
 - 7 users launched different queries - 6 users waited for one second, 7th waited 0.74 second.
- **CPU Time** – the utilization of server processors by all queries in a given second in time.
- **IO Load** – utilization of processors for I/O operations
- **Server CPU** – load of server's processors
- **SQL Server CPU** - CPU load on the server's processors by selected SQL instance. Series should be identical or imitating "Server CPU" line.

For better readability of the graph:

- you can click to disable (or enable) given series of the chart - we do it in the area of the chart legend
- you can zoom the chart

Here is an example showing series of Elapsed Time and CPU Time in the narrower time horizon:



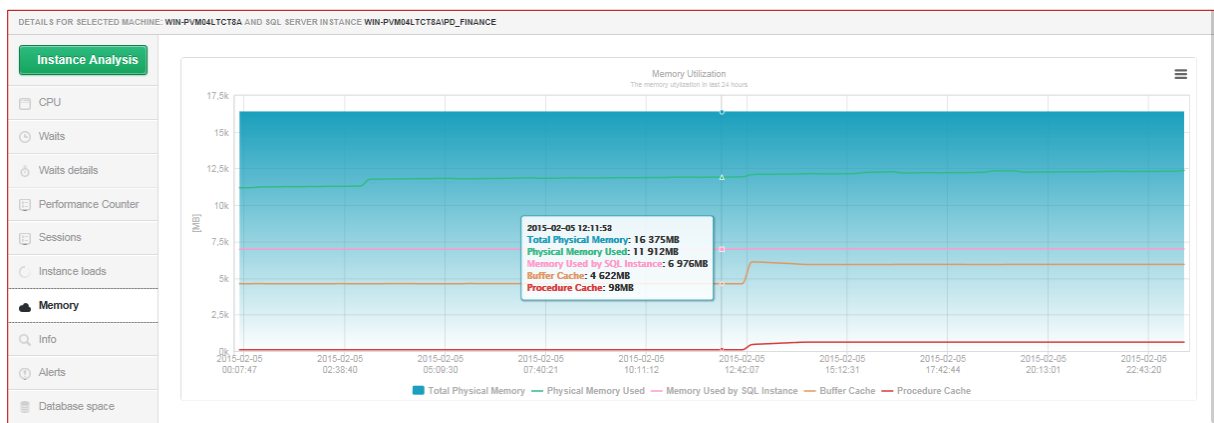
Dashboard also allows you to view basic information about SQL instances, among others:

- SQL instance version
- the number of available virtual processors
- amount of assigned memory
- recent changes to instance and SQL instances parameters

We get it after clicking on the **[Info]** tab:

DETAILS FOR SELECTED MACHINE: WIN-PVM04LTCT8A AND SQL SERVER INSTANCE WIN-PVM04LTCT8A\PO_FINANCE				
Instance Analysis		Q SQL Instance information		Last changes
CPU	Parameter	Value	Date change	Description
Waits	Server type	PRODUCTION SERVER	2014-03-06 09:18:12	Server parameter Parameter affinity mask changed to 0
Waits details	SQL Version	2012	2015-02-04 12:48:29	Server properties Property ProcessID changed to 4748
Performance Counter	Product/Version	11.0.2100.60	2015-02-05 12:57:18	Database parameter for adv_works Parameter SQLSortOrder changed to 52
Sessions	Edition	Enterprise Evaluation Edition (64-bit)	2014-09-18 15:00:59	Login created/updated Login: otm_distributor, type: SQL_LOGIN
Instance loads	Product/Level	RTM	2014-09-18 14:40:45	Last database created distribution
Basic information about the instance		Instance startup date	2015-02-04 12:38:37	Last backup executed Backup executed for database adv_works
Q Info		Databases count	10	
Alerts		Logins count	15	
Database space		Hard disk free space [MB]	343159	
		Number of virtual cores	8	

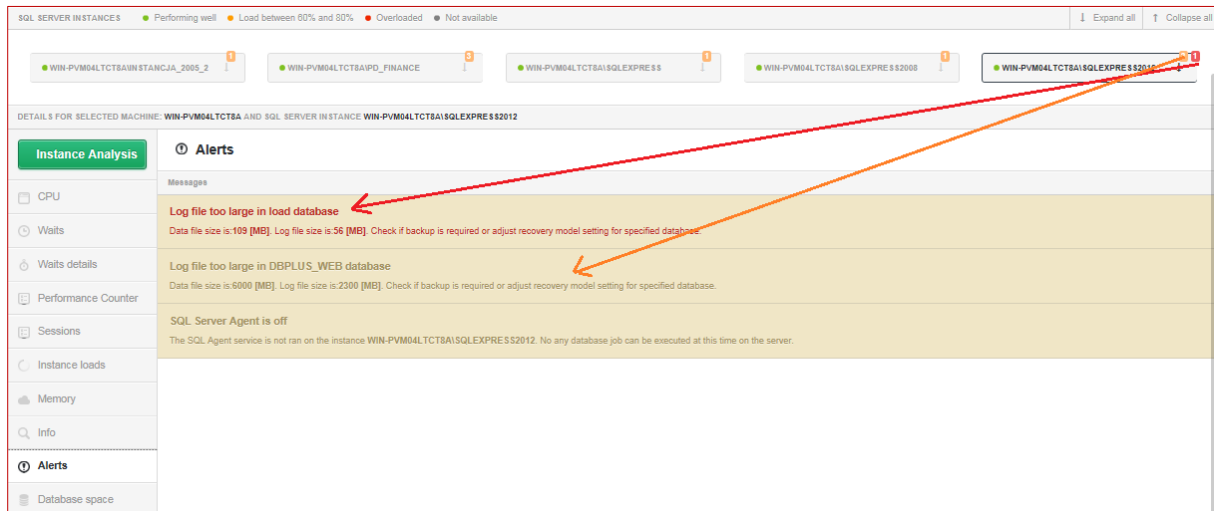
When you click on **[Memory]**, you have information about memory utilization on the server and by SQL instance. In addition, we see what level of memory usage is kept by **Buffer pool** and **Procedure cache**:



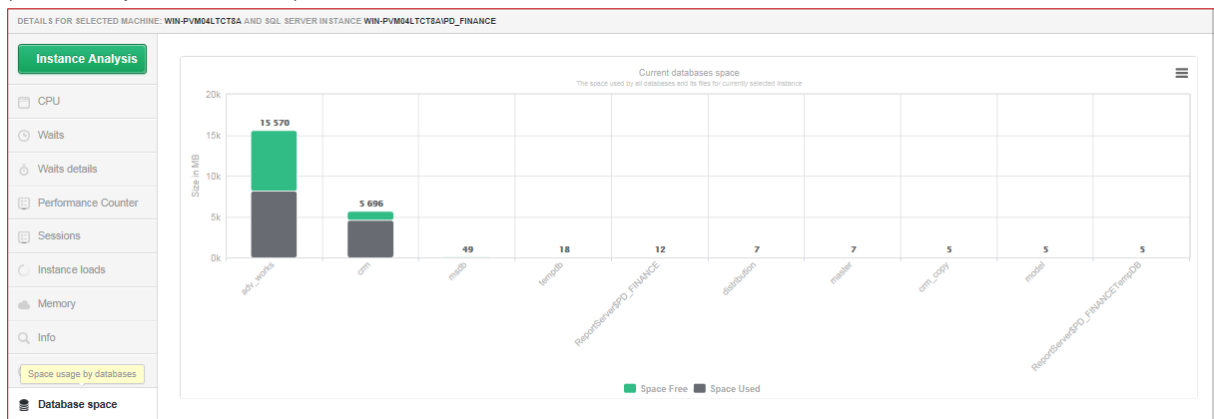
An additional feature of the dashboard is **alerting** about more or less critical performance events on server, among others:

- Increased CPU utilization on server or CPU
- Locks
- The decrease in Buffer Cache Hit Ratio
- The increase in the number of sessions or open transactions
- Lack of free memory
- Increased number of sessions or open transactions
- Stopped SQL Agent service
- Lack of space on server's disks

Example of **Alerts** screen is shown below:



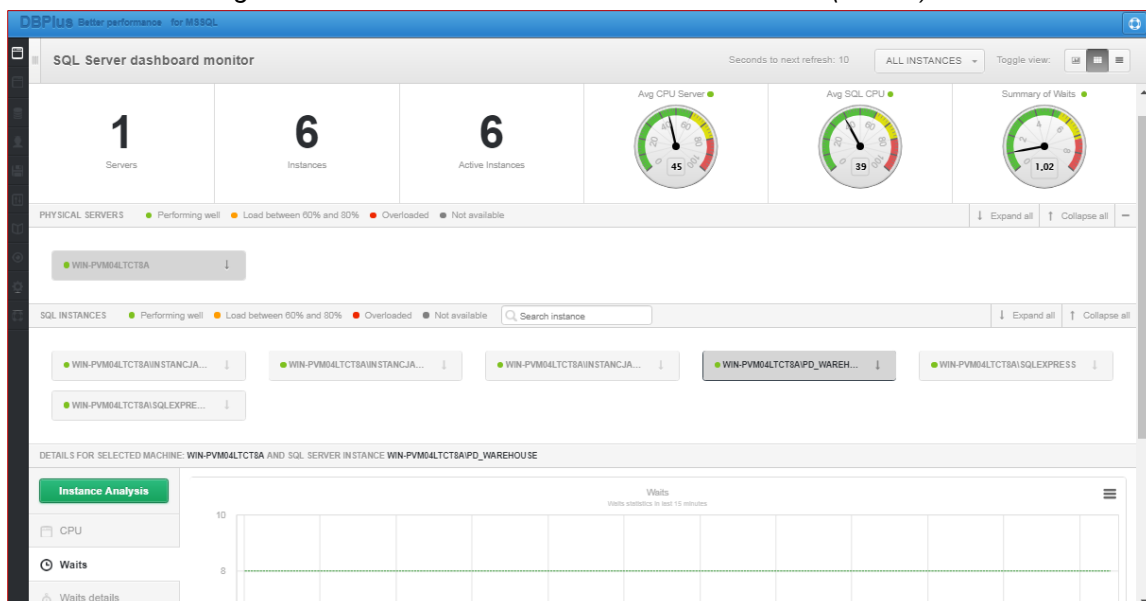
When you click in the **[Database space]**, you can get to know the current size of database on instance (size is expressed in MB):



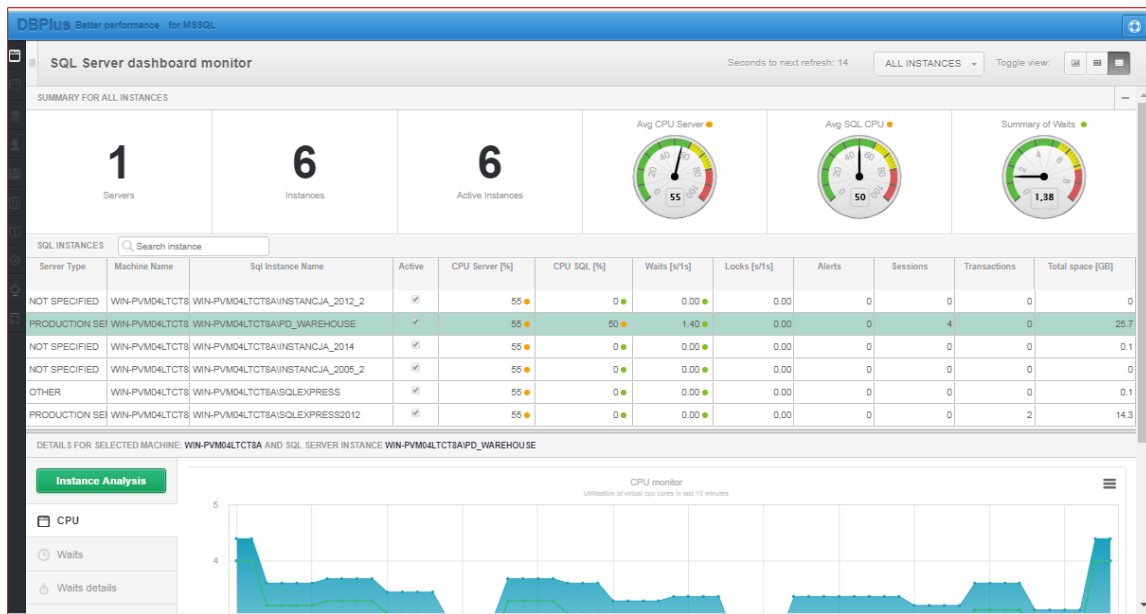
6.1.5 Dashboard – various forms of presentation

Dashboard is available in 3 modes, which are switched by clicking [Toggle View] icon in the top right corner. Available modes are:

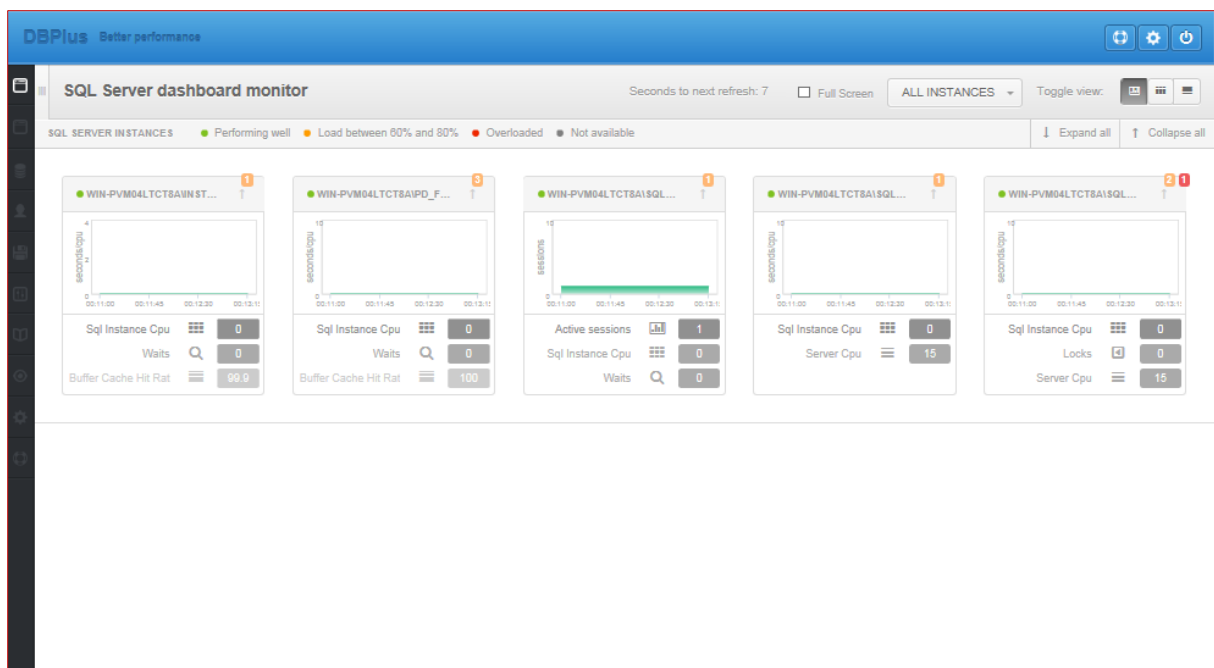
Icon View - showing monitored servers/instances in the form of icons (default)



Grid View - showing instances in the form of a table



Television Mode - showing instances of SQL Server in the form of icons with automatically switching performance indicators



6.1.6 Grid/table options

Data in the DBPLUS Performance Monitor system are presented in tables. In most cases, each table has the following options:

- Ability to export to a file in CSV format
- Formatting columns in tables

The export function is available by right-clicking on the table:

Database	Query text	Query Hash	Plan Hash	Elapsed Time [seconds]	Cpu Time [seconds]	Io, Wait Time [seconds]	Time per 1 exec [seconds]	Executions	Disk reads [Blocks]	Buffer gets [Blocks]	Buffer writes [Blocks]	Rows processed
adv_works	select top(10) L* from Production.Transaction	0x84C102F23329	0x31F605092	79 033.42	69 827.71	9 205.70	2.5828	30 600	13 305	4 577 999 159	0	408 632 400
adv_works	select * from (select p.ProductID, p.ProductN	0x0FE47590673F	0xE8DB3456	33 517.99	32 190.95	7 322.05	5.5155	7 164	11 384 253	4 791 018 132	0	924 327 936
adv_works	SELECT Production.GetProductName(Produc	0xDF1D6547F4E	0xCB74A269C	2 84	0.0320	839 630	6	2 715 363 420	0	894 205 950	0	894 205 950
adv_works	select @qty = isnull(sum(OrderQty),?) from Pr	0x41E7652AF	0x41E7652AF	66	0.0000	824 327 936	656	2 890 258 488	0	894 205 950	0	894 205 950
adv_works	select @name = Name from Production.Produ	0x797529C73920	0xD3EB48A8	6 973.81	6 432.56	541.25	0.0000	824 327 936	0	1 848 741 840	0	924 327 936
adv_works	select @qtyPI = isnull(sum(Quantity),?) from F	0x57F2C80F8BB	0x524FD52Cf	1 904.13	1 729.33	174.80	0.0101	187 851	1 062	164 933 178	0	188 023
adv_works	select top ? * from Production.TransactionHist	0x677E3020F458	0x60305CE48	1 312.97	1 059.80	253.17	0.0067	195 019	38	6 045 589	0	0

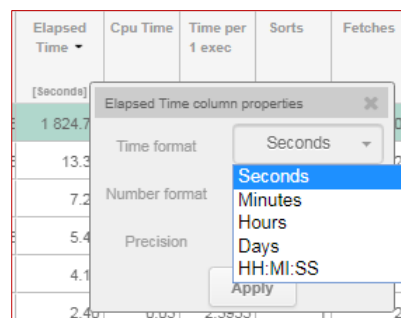
We have the possibility to export data with original values or with formatting for individual columns.

Another table functionality allows formatting of numerical values in columns. The option is available by right-clicking on the column header and min. allows you to:

- unit selection – e.g. Elapsed Time in seconds, minutes, Disk readouts in GB, etc.
- selection of a shortcut for large numbers - e.g. kilo, Mega, ...
- determining the precision of the number

The formatting of columns applies to all grids in the system and is set independently (i.e., the formatting set for the table with the statistics of queries in the Instance Load screen will not copy automatically e.g. to the Sql Details screen).

After clicking on the column header, the settings window appears



➤ Grid manager

The ability to change settings will be introduced in stages. First part are introduced changes on the **Load Trends** and **SQL Details** pages.

The User for these tables on the pages can change for each of the columns:

- Order of displayed columns
- Visibility of columns
- Change the format
- Change of precision
- Change of width

Additionally, it is now possible to hide the Summary row on each page, using the settings available after press the **[cog]** icon. As before, the data contained in the grid can be freely exported to a file.

The order of displayed columns

To change the order of columns, click on the header of the column, hold down the mouse button, drag the columns and drop them to the desired place on the table.

Logdate	Elapsed time	Executions	Active sessions	Blks read	Blks written	Temp blks read	Temp blks written	Wait time	IO time	Lock time	Rollbacks	Tuples returned	Rows	No of temp files	Data written to temp	Blk read time	Blk write time	Blks hit	
	[Seconds]			[Blocks]	[Blocks]	[Blocks]	[Blocks]	[Seconds]	[Seconds]	[Seconds]			[Rows]		[MB]	[Seconds]	[Seconds]	[Blocks]	
2020-02-28	409.030	86 827	0	689	0	28 188	0	0	0	0	3 751	22 097 964	455 787	37	220 MB	0	0	2 331 374	
2020-03-02	78.360	55 821	0	313	22 705	0	22 127	0	0	0	2 760	16 272 699	308 416	29	173 MB	0	0	1 497 210	
2020-03-03	158.780	60 155	0	344	22 259	0	21 274	0	0	0	2 974	17 803 563	310 732	28	166 MB	0	0	1 599 230	
2020-03-04	217.670	59 481	0	6 151	20 756	0	20 668	0	0	0	2 937	20 901 482	338 158	27	161 MB	0	0	1 796 034	
2020-03-05	151.890	63 469	0	1 003	23 097	0	22 954	1	0	0	3 087	23 357 777	352 929	30	179 MB	0	0	1 863 128	
2020-03-06	166.140	61 587	0	633	22 912	0	22 924	0	0	0	2 839	25 747 499	359 203	30	179 MB	0	0	1 987 850	
2020-03-09	77.110	51 877	0	496	21 084	0	20 720	0	0	0	2 574	15 619 704	279 140	27	162 MB	0	0	1 466 907	
2020-03-10	110.700	59 781	0	369	24 667	0	23 790	0	0	0	2 944	18 680 150	306 760	31	186 MB	0	0	1 665 009	
2020-03-11	83.500	48 204	0	525	20 303	0	19 157	0	0	0	2 397	16 038 215	260 606	25	150 MB	0	0	1 411 514	
2020-03-12	107.870	67 061	0	12 030	26 495	0	26 892	0	0	0	3 119	25 940 476	369 647	35	210 MB	0	0	2 059 855	

Visibility of columns

To hide a column, right-click on the column header to be hidden. A popup menu will open where the Hide column button should be selected. The indicated column is hidden.

Logdate	Elapsed Time	Rows	Executions	Blks hit	Blks read	Blks dirtied	Blks written	Temp blks read	Temp blks written	Wait time
	[Seconds]	[Rows]		[Blocks]	[Blocks]	[Blocks]				[Seconds]
2020-03-09	77.11	279 140	51 877	1 466 907	496	21 084	20 720	0	0	0
2020-03-02	78.36	308 416	55 821	1 497 210	313	22 705	22 127	0	0	0
2020-03-11	83.50	260 606	48 204	1 411 514	525	20 303	19 157	0	0	0
2020-03-12	107.87	369 647	67 061	2 059 855	12 030	26 495	26 892	0	0	0
2020-03-10	110.70	306 760	59 781	1 665 009	369	24 667	23 790	0	0	0
2020-03-05	151.89	352 929	63 469	1 863 128	1 003	23 097	22 954	0	0	1
2020-03-03	158.78	310 732	60 155	1 599 230	344	22 259	21 274	0	0	0
2020-03-06	166.14	359 203	61 587	1 987 850	633	22 912	22 924	0	0	0

To reveal a column, click the **[cog]** button in the upper right corner of the table. After the popup menu open, select the **[Show hidden columns]** option, then indicate the column you want to rediscover in the table. The uncovered column will appear last on the right side of the table.

Logdate	Elapsed Time	Rows	Blks hit	Blks dirtied	Temp blks read	Temp blks written	IO time	Active sessions	Sessions	Connectio.	Commits	Rollbacks	Tuples returned	Tuples fetched	Tuples inserted	Tuples updated	Tuples deleted	Conflicts
	[Seconds]	[Rows]	[Blocks]	[Blocks]	[Blocks]	[Blocks]	[Seconds]											
2020-03-09	77.11	279 140	1 466 907	21 084	20 720	0	0	0	10	221	34 014	2 574 15 619	1 696 148	23 612	13 382	0	0	0
2020-03-02	78.36	308 416	1 497 210	22 705	22 127	0	0	0	11	259	36 694	2 760 16 272	1 680 252	25 262	11 356	0	0	0
2020-03-11	83.50	260 606	1 411 514	20 303	19 157	0	0	0	10	205	31 745	2 397 16 038	1 369 643	21 825	9 237	21 418	0	25
2020-03-12	107.87	369 647	2 059 855	26 495	26 892	0	0	0	10	292	43 992	3 119 25 940	2 164 687	29 606	12 373	24 629	0	35
2020-03-10	110.70	306 760	1 665 009	24 667	23 821	23 790	0	0	10	257	39 351	2 944 18 680	1 594 995	26 992	11 495	13 382	0	31
2020-03-05	151.89	352 929	1 863 128	23 097	22 994	22 964	0	0	16	421	41 915	3 087 23 357	1 867 391	28 029	11 356	20 597	0	30
2020-03-03	158.78	310 732	1 599 230	22 259	21 302	21 274	0	0	12	278	39 813	2 974 17 803	1 414 794	27 081	11 200	12 110	0	28
2020-03-06	166.14	359 203	1 987 850	22 912	22 954	22 924	0	0	14	359	41 954	2 839 25 747	2 045 668	25 653	10 320	21 641	0	30
2020-03-04	217.67	338 158	1 796 034	20 756	20 695	20 668	0	0	16	377	41 716	2 937 20 901	1 660 824	23 436	9 518	19 064	0	27

Change of data format / precision

To change the data format settings, precision, right-click on the column heading where you want to change the data. After making changes, save the changes by click **[Apply]** button.

Logdate	Elapsed Time	Rows	Blks hit	Blks dirtied	Temp blks read	Temp blks written	IO time	Active sessions	Sessions	Connectio.	Commits	Rollbacks	Tuples returned	Tuples fetched	Tuples inserted	Tuples updated	Tuples deleted	Conflicts
	[Seconds]				[Blocks]	[Blocks]	[Seconds]											
2020-03-02	78.36	308 416	1 497 210	22 705	22 127	0	0	0	0	1	87	10 974	47	893 425	46 707	0	0	0
2020-03-03	158.78	310 732	1 599 230	22 259	21 274	0	0	0	0	1	94	10 642	51	908 387	47 850	0	0	0
2020-03-04	217.67	338 158	1 796 034	20 756	20 695	20 668	0	0	0	2	122	11 043	67	909 159	48 982	0	0	0
2020-03-05	151.89	352 929	1 863 128	23 097	22 994	22 964	0	0	0	2	108	11 673	68	965 965	55 008	0	0	0
2020-03-06	166.14	359 203	1 987 850	22 912	22 954	22 924	0	0	0	1	90	11 349	62	943 042	53 902	0	0	0
2020-03-09	77.11	279 140	1 466 907	21 084	20 720	0	0	0	0	1	81	10 212	44	831 409	43 034	0	0	0
2020-03-10	110.70	306 760	1 665 009	24 667	23 821	23 790	0	0	0	1	93	11 712	48	949 987	49 221	0	0	0
2020-03-11	83.50	260 606	1 411 514	20 303	19 157	0	0	0	0	1	75	9 442	40	765 231	39 834	0	0	0
2020-03-12	107.87	369 647	2 059 855	26 495	26 892	0	0	0	0	1	105	13 232	61	1 085 776	57 771	0	0	0
2020-03-13	21.96	6 680	6 982	0	0	0	0	0	0	1	36	4 524	23	368 729	19 262	0	0	0
2020-03-16	33.76	8 902	9 152	0	0	0	0	0	0	1	48	6 038	28	494 780	26 137	0	0	0

Change of width

To change the column width, click the column edge, hold and move it to the right or left to change the width. The current solution used in the DBPLUS application adjusts the width of the columns to the width of the screen. Therefore, with many columns in the table, the width of the columns will always be converted in proportion to the width of the screen.

POSTGRES SQL TRENDS STATISTICS																			Clear selection		
Logdate	Elapsed Time	Rows	Blks hit	Blks dirtied	Temp blks read	Temp blks written	IO rate	Sessions	Tuples inserted	Tuples updated	Tuples deleted	Conflicts	No of temp files	Data written to temp	Deadlocks	Blk read time	Blk write time	Executions	Blks written		
	[seconds]	[rows]	[blocks]	[blocks]	[blocks]	[blocks]	[rows]							[MB]		[seconds]	[seconds]		[blocks]		
2020-03-02	49.93	16 120	16 510	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8 109	0		
2020-03-03	55.78	15 605	17 161	0	0	0	0	1	0	0	0	0	0	0	0	0	0	7 842	0		
2020-03-04	69.25	18 808	48 363	0	0	0	0	2	0	0	0	0	0	0	0	0	0	8 305	0		
2020-03-05	63.48	18 113	27 936	0	0	0	0	2	0	0	0	0	0	0	0	0	0	8 668	0		
2020-03-06	55.19	16 672	17 567	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8 379	0		
2020-03-09	52.05	15 005	15 398	0	0	0	0	1	0	0	0	0	0	0	0	0	0	7 547	0		
2020-03-10	72.16	17 223	17 570	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8 663	0		
2020-03-11	52.77	13 902	17 053	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8 304	0		
2020-03-12	71.77	19 490	20 008	0	0	0	0	1	0	0	0	0	0	0	0	0	0	9 816	0		
2020-03-13	21.96	6 680	6 982	0	0	0	0	1	0	0	0	0	0	0	0	0	0	3 362	0		

Storage of table configurations

The configuration for each of the tables is saved in two ways: at the browser cache level on the user's computer or in the repository database.

In order to permanently save the settings to the repository database, Windows authorization must be enabled in the DBPLUS Performance Monitor application (enabled at the Configuration Wizard level), and the Security module (Menu Configuration> Settings: Security "ON") must be started. The settings are saved for all monitored instances for each user separately.

Restore default settings

If User need to return to the default settings, they can do this by click the cog button and select **[Restore grid defaults]**.

SQL STATISTICS												Grid options
Date	Plan Id	Elapsed Time	Blks read time	Blks write time	Executions	Blks hit	Blks read	Blks dirtied	Blks written	Rows per 1 Exec	Blks hit per 1 Exec	
		[seconds]	[seconds]	[seconds]		[blocks]	[blocks]	[blocks]	[blocks]	[rows]	[blocks]	
2020-03-16 10:09:01	2626426938	2.9	0	0	60	60	0	0	0	1.00	1.00	Restore grid defaults
2020-03-16 10:39:21	2626426938	2.8	0	0	60	60	0	0	0	1.00	1.00	
2020-03-16 10:24:11	2626426938	2.6	0	0	60	60	0	0	0	1.00	1.00	
2020-03-16 13:10:11	2626426938	2.4	0	0	59	59	0	0	0	1.00	1.00	

At any time, the User can restore the default setting for a given column by click on the **[Restore defaults]** button for a given column.

6.1.7 Format SQL text queries

On each page where the query text is presented, a **[SQL Format]** button has been added, after which query text will be formatted.

Sql Statements														Statements filter: All statements
SNAPSHOT OF SQL STATEMENTS EXECUTED WITHIN 15 MINUTES AT 2020-03-16 12:07:48														
Search query by any value in below snapshot table														
Database	Query text	Query Hash	Plan Hash	Use plan guide	Elapsed Time	Cpu Time	Time per 1 exec	Executions	Disk reads	Buffer gets	Buffer writes	Rows processed	Elapsed Time Load	Cpu Time Load
					[seconds]	[seconds]	[seconds]		[blocks]	[blocks]	[blocks]		[%]	[%]
Not spe...	select @cpu_sq = isnull(SQLProcess...	0x3D7B0EF32...	0x2314681...		5.08	5.08	0.0862	59	0	0	0	59	82	82
master	set @cpu_sq_pc = (select cntb_value...	0x692F1B5AE...	0x238D37B...		0.53	0.53	0.0009	596	0	0	0	596	9	9
master	select qst_query_hash, qst_query_plan...	0x8AEB5C268...	0x33A3AF8...		0.83	0.83	0.0010	29	0	0	0	0	0	0
Not spe...	select q_wait_type, sum(s.wait_time...	0x0E675F72F...	0x556AD1...		0.08	0.08	0.0014	59	0	0	0	4 838	1	1
Not spe...	SELECT target_data FROM sys.dm...	0x52B3B6CED...	0x99E4203...		0.23	0.22	0.0762	3	0	0	0	3	4	3
Not spe...	select ? as rec_type, s.session_id, s.pr...	0xF08DA1D55...	0xB14512F...		0.09	0.09	0.0031	30	0	8 678	0	30	1	1
Not spe...	select null as var0, total_elapsed_tim...	0xC4569E40B5...	0xD48ABA...		0.16	0.16	0.0054	30	0	0	0	5	3	3

```

STATEMENT TEXT FOR QUERY HASH: 0X0E675F72F7AC2FB
SELECT
  q_wait_type,
  SUM(q_wait_time_ms) AS wait_time_ms
FROM
  (SELECT
    s_wait_type,
    s_wait_time_ms
  FROM
    SYS.dm_os_wait_stats a
  WHERE
    s_wait_time_ms > 0
  UNION ALL
  SELECT
    o_wait_type,
    o_wait_duration_ms AS wait_time_ms
  FROM
    SYS.dm_os_waiting_tasks o
  WHERE
    wait_duration_ms > 1000)

```


6.2 Instance Analysis Menu

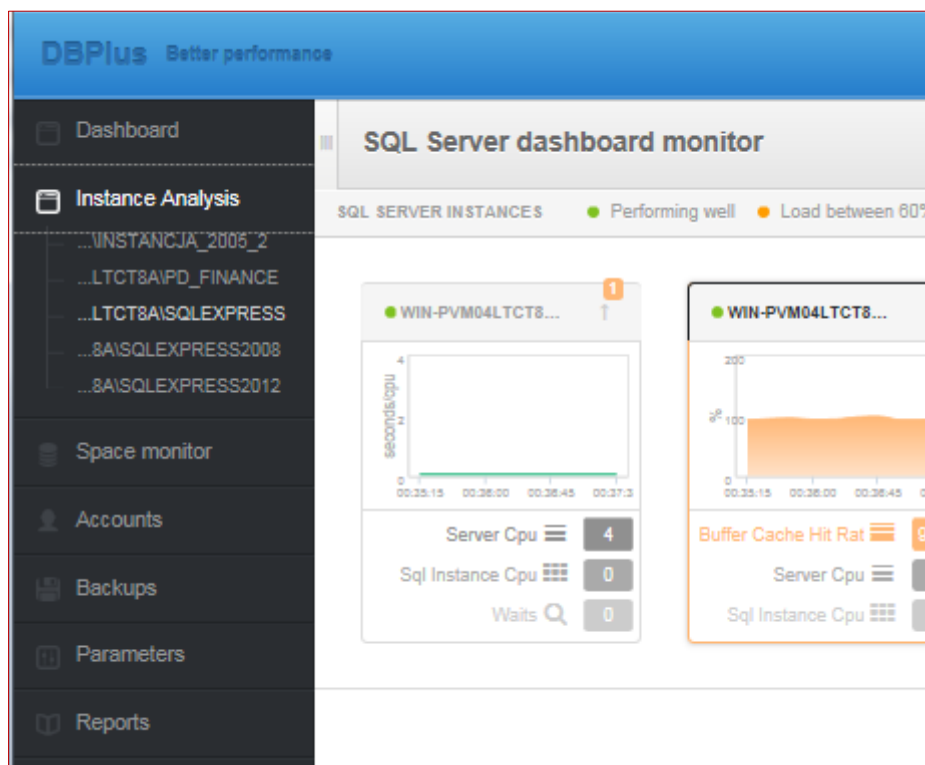
6.2.1 Performance Menu – Instance Analysis

Dashboard of the DBPLUS System Performance Monitor allows you to track the performance of SQL Server instances, and show how it looked over the last 15 minutes or the last 24 hours. For a detailed analysis of the load at any given moment in time, and seek answers to questions like:

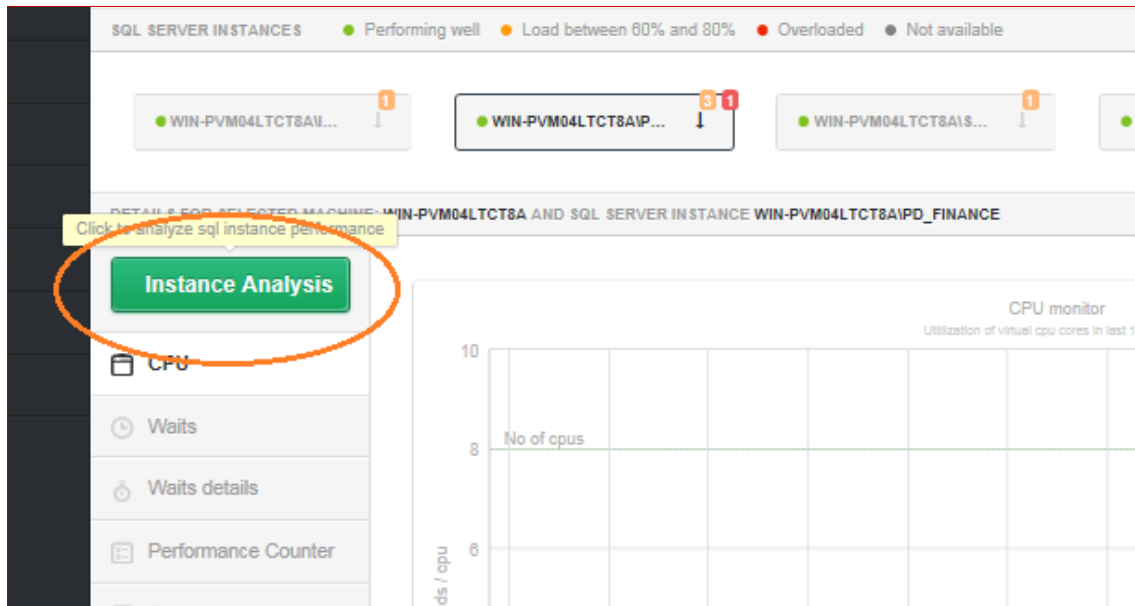
- why SQL instances is running slow?
- why user had problems in the application 3 days ago at 15:48?
- why my report performed 15 minutes?
- etc.

... You need to enter the module **[Instance Analysis]** and have two possibilities:

- On the left side of the menu, click on **[Instance Analysis]** shows a list of SQL instances



- Displaying the details of the instance after it has been selected on the Dashboard



6.2.1.1 Instance Load Tab

Instance Load is a screen showing Instance load time, which was partially discussed in the previous section on the **Dashboard**.

In the **Performance** module functionality of the graph is greater. First of all, we can here:

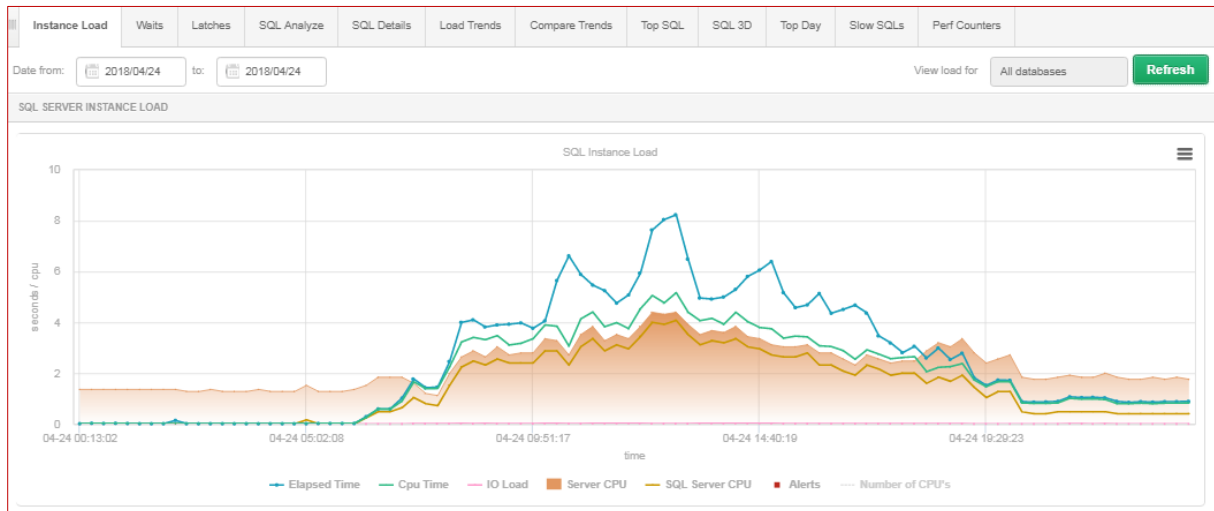
- check the load of the instance in the wider horizon e.g. today, yesterday, a month or even 3.5 years ago.
- look at the SQL queries / commands, which generated the load
- assess what SQL instance did at this time among others if performed a lot of disk operations, whether there were locks, etc.

Instance Load screen consists of following areas:

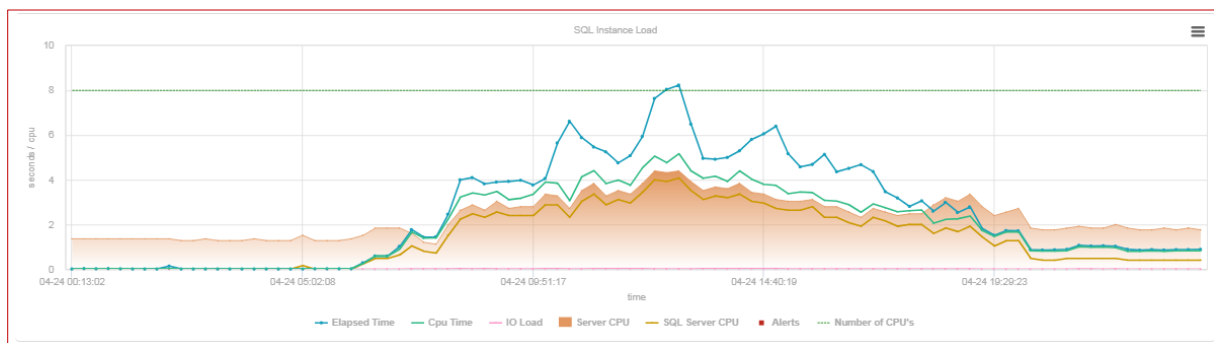
- filtration fields - fields of dates by which we define the period in which we want to check- the load
- the graph presenting the load
- the load information at a given moment of time:
 - list of queries with execution statistics
 - waits - what database was doing at the time to perform queries
 - load from the point of view of databases on instance

Chart consists of following series:

- **Elapsed Time** - shows the summary time all users are waiting on the result of a query at a given second of time.
- **CPU Time** – the utilization of virtual server processors by all queries in a given second in time.
- **IO Load** – utilization of processors for I/O operations
- **Server CPU** – load of server's processors
- **SQL Server CPU** - CPU load on the server's processors by selected SQL instance. Series should be identical or imitating "Server CPU" line.
- Number of CPU's – (default hidden)
- Alerts – the number of alerts for a given snap.



After clicking on the legend of the chart on the Number of CPU's label, an additional series appears:



The chart is "clickable" - click on the selected part / section will refresh the bottom of the screen with information about requests and waits that generated the data load.

IMPORTANT: The screen shown that SQL Server instance is on a server with 8 CPUs.

Data for chart of the Instance load is calculated by monitoring service **DBPLUSCATCHER** - a component of the **DBPLUS Performance Monitor**. Monitoring Service performs a number of procedures examining SQL instance performance. The result of the operation of these procedures is a snapshot (snap) that is created every 15 minutes.

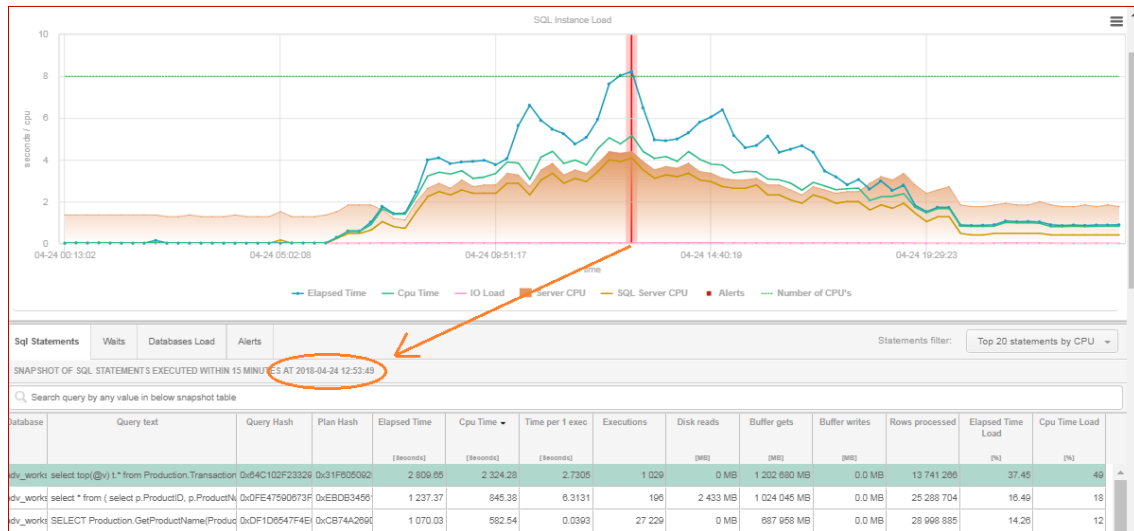
If one session for 15 minutes performed 3 queries:

- The first query 5 minutes
- The second query 14 minutes and 59 seconds
- The third query 1 second

The graph for a given situation will present

- 1 CPU usage by Elapsed Time - so long the session owner was waiting for the result of the three queries
- occupancy of less than 1 CPU by the line CPU Time - if any inquiry carried out in one thread
- occupancy of more than 1 CPU by the line CPU Time - if one of the queries was executed the in multithreaded mode (wait CXPACKET)

After clicking on the selected point in time, the lower part of the screen is refreshed, information requests and waits.



If you scroll down the screen:

Database	Query text	Query Hash	Plan Hash	Elapsed Time [seconds]	CPU Time [seconds]	Time per 1 exec [seconds]	Executions	Disk reads [MB]	Buffer gets [MB]	Buffer writes [MB]	Rows processed	Elapsed Time Load [%]	CPU Time Load [%]
adv_works	select top(10) t.* from Production.TransactionHistory	0x64C102F23326	0x31F005062	2 809.85	2 324.28	2.7305	1 029	0 MB	1 202 680 MB	0.0 MB	13 741 266	37.45	40
adv_works	select * from (select p.ProductID, p.ProductName	0x0FE47590873F	0xE8DB3456	1 237.37	845.38	6.3131	196	2 433 MB	1 024 045 MB	0.0 MB	25 288 704	16.49	18
adv_works	SELECT Production.GetProductNames(Products	0xDF1D6547F4E	0xCB74A286	1 070.03	582.54	0.0393	27 229	0 MB	687 668 MB	0.0 MB	28 998 885	14.26	12
adv_works	select @city = isnull(sum(Quantity), 7) from P	0xF9C0C8784D6	0x41E7652AF	414.51	355.89	0.0000	25 243 612	0 MB	617 066 MB	0.0 MB	25 243 612	5.53	8
adv_works	select @name = Name from Production.Product	0x797529C7362D	0xD3EB49A8	445.52	299.30	0.0000	28 997 820	0 MB	686 018 MB	0.0 MB	28 997 820	5.94	6
adv_works	select @cityPI = isnull(sum(Quantity), 7) from F	0x57F2C80F8B8	0x524FD52Cf	206.68	177.21	0.0000	25 243 612	0 MB	394 450 MB	0.0 MB	25 243 612	2.78	4
adv_works	select top 2 * from Production.TransactionHist	0x67E3020F458	0x60305CE45	69.69	57.02	0.0104	6 726	0 MB	46 136 MB	0.0 MB	6 726	0.93	1
adv_works	SELECT p.Name AS ProductName, NonDisco	0x00D46959695A	0x6943A3963	37.03	28.36	0.2572	144	0 MB	157 517 MB	0.0 MB	0	0.49	1

STATEMENT TEXT FOR QUERY HASH: 0x64C102F23326DC38

```
select top(10) t.* from Production.TransactionHistory t where t.ProductID = @p ORDER BY t.TransactionDate option(optimize for (@p=7))
```

EXPLAIN PLAN FOR PLAN HASH: 0x31F005092B25E86A

Show plan objects for 0x31F005092B25E86A

```
Database: adv_works
Missing indexes
  /Missing index impact: 96.7274% use [adv_works]; create index [missing_index_TransactionHistory_05042018] on [Production].[TransactionHistory] ([ProductID]) include ([TransactionID],[Reference
SELECT (Cost = 3.07821, Rows = 0, CPU = 0, IO = 0)
```

There are 4 additional sub-tabs:

- SQL Statements
- Waits
- Databases Load
- Alerts

SQL Statements is the query statistics presented in the form of a table. By default, system displays the most aggravating question for the duration of Elapsed Time or CPU utilization. The method of display can be changed after clicking on the field [Statement filter] - you can also view a complete list of queries that participated in the load.

The table with queries:

- You can sort on any column
- Change the number of queries per page
- Search e.g. after a part of the query text

The table contains the following information:

- **Database** – name of the database where the SQL statement was executed
- **Query Text** - full text of SQL command
- **Query Hash/Query Plan Hash** - an identifier of a query and an identifier of execution plan
- **Elapsed Time** - the duration in seconds for all query executions within last 15 minutes. It may happen that the time is not for the last 15 minutes only for the last execution, which could take more than 15 minutes.
- **CPU time (sec)** - CPU utilization time in seconds by the query during last 15 minutes. It may happen that the time is not for the last 15 minutes only for the last execution, which could take more than 15 minutes.
- **Time per one Exec (sec)** – duration of query in seconds for one execution,
- **Executions** - number of executions of the query in last 15 minutes. It may happen that the time is not for the last 15 minutes only for the last execution, which could take more than 15 minutes.
- **Disk Reads** - number of disks reads for the query in last 15 minutes. It may happen that the time is not for the last 15 minutes only for the last execution, which could take more than 15 minutes.
- **Buffer Gets** - number of buffers utilized by the query during last 15 minutes. It may happen that the time is not for the last 15 minutes only for the last execution, which could take more than 15 minutes.
- **Buffer Writes** - number of buffers writes by the query during last 15 minutes
- **Rows processed** - number of rows returned by the query in last 15 minutes.
- **Elapsed Time Load** - the percentage of total instance load caused by the query during last 15 minutes.
- **CPU Time Load** - the percentage of total SQL instance servers' CPUs load caused by the query during last 15 minutes.

IMPORTANT - in MS SQL statistics are counted after the query. In the case of a long-time query (e.g. more than 1 hour), the information about the query will appear only in the snap in which the query has been completed and all statistics will be counted for the entire query.

In the column Query Hash / Query Plan Hash (each line presenting statistics of the execution) shows **[+]** ("plus") button

Database	Query text	Query Hash	Plan Hash	Elapsed Time	Cpu Time	Time per 1 exec
				[Seconds]	[Seconds]	[Seconds]
adv_works	select top(@v) t.* from Production.Transaction	0x84C102F23329	0x31F005092	2 809.65	2 324.28	2.7305
adv_works	select * from (select p.ProductID, p.ProductName	0x0FE47590D	0xEBDB3456	1 237.37	845.38	6.3131
adv_works	SELECT Production.GetProductName(ProductID,	0xDF1D6547F4E	0xCB74A266C	1 070.03	582.54	0.0393
adv_works	select @qty = isnull(sum(OrderQty),?) from Pr	0xF9C0C67B4D5	0x41E7652AF	414.51	355.89	0.0000

When you click on **[+]** ("plus"), it shows additional context menu, which enables for detailed analysis of a particular query, which will be discussed in the section **"Performance SQL Details"**

Database	Query text	Query Hash	Plan Hash	Elapsed Time	Cpu Time	Time per 1 exec	Executions
				[Seconds]	[Seconds]	[Seconds]	
adv_works	select top(@v) t.* from Production.Transaction	0x84C102F23329	0x31F005092	2 809.65	2 324.28	2.7305	1 029
adv_works	select * from (select p.ProductID, p.ProductName	0x0FE47590D	0xEBDB3456	1 237.37	845.38	6.3131	196
adv_works	SELECT Production.GetProductName(ProductID,	0xDF1D6547F4E	0xCB74A266C	1 070.03	582.54	0.0393	27 229
adv_works	select @qty = isnull(sum(OrderQty),?) from Pr	0xF9C0C67B4D5	0x41E7652AF	414.51	355.89	0.0000	25 243 612
adv_works	select @name = Name from Production.Product	0x797529C7392D	0x524FD52C9	206.68	177.21	0.0000	25 243 612
adv_works	select @qtyPI = isnull(sum(Quantity),?) from P	0x57F2C80FB8B	0x524FD52C9	206.68	177.21	0.0000	25 243 612

For example, if you select **[Add to query hash list]**, we move a query identifier to the clipboard with a list of queries for later analysis of specific queries

Below the slide of 4 queries added to the analysis in **SQL Details** functionality.

Instance Load
Wait
Latches
SQL Analyze
SQL Details
Load Trends
Compare Trends
Top SQL
SQL SD
Top Day
Slow SQLs
Perf Counters

SQL Statements

Click on query hash to analyze Query Performance Details

Query Hashes list

- 0x0F1D554705C5B93
- 0x0FE47590073F13D6
- 0x54C102F2329C0C98
- 0x2F5EE731FCEDF74A

Statements filter: Top 20 statements by CPU

SNAPSHOT OF SQL STATEMENTS EXECUTED WITHIN 15 MINUTES AT 2018-04-24 12:32:49

Search query by any value in below snapshot table

Database	Query text	Query Hash	Plan Hash	Elapsed Time (Seconds)	Cpu time (Seconds)	Time per 1 exec (Seconds)	Executions	Disk reads (MB)	Buffer gets (MB)	Buffer writes (MB)	Rows processed	Elapsed Time Load (%)	Cpu Time Load (%)
adv_works	select top (@v) t.* from Production.TransactionHistory	0x54C102F2329C0C98	0x31F605092	2 808.65	2 324.28	2 7305	1 029	0 MB	1 202 890 MB	0 0 MB	13 741 269	37.45	40
adv_works	select * from [select p.ProductID, o.ProductName	0x0FE47590073F13D6	0xEBCB3490	1 237.37	845.38	6 3131	195	2 433 MB	1 024 045 MB	0 0 MB	25 288 704	16.49	18
adv_works	SELECT Production.OleProductName/Product	0x0FD0C547F4E1	0xC874A2096	1 070.03	592.54	0 0393	27 229	0 MB	687 959 MB	0 0 MB	26 966 885	14.25	12
adv_works	select @qty = ianl(jum(OrdOrderQty,?) from Ph	0xF8C0C67B4DCE	0x41E7652AF	414.51	355.89	0 0000	25 243 612	0 MB	617 095 MB	0 0 MB	25 243 612	5.53	8
adv_works	select @name = Name from Production.Prod	0x70752DC73620	0x03EB49A98	445.52	299.30	0 0000	28 967 920	0 MB	698 019 MB	0 0 MB	28 967 920	5.64	6
adv_works	select @@PI = ianl(jum(Quantity,?) from F	0x57F2C80FB8BB	0x524FD5C0F	206.68	177.21	0 0000	25 243 612	0 MB	364 450 MB	0 0 MB	25 243 612	2.76	4
adv_works	select top ? * from Production.TransactionHist	0x7E7E3X0ZF458	0x030035C46	66.69	57.02	0 0104	6 726	0 MB	46 136 MB	0 0 MB	6 726	0.93	1
adv_works	SELECT p.Name AS ProductName, NonDisc	0x0C4D9050065A	0x0843A3963	37.03	28.36	0 2572	144	0 MB	157 517 MB	0 0 MB	0	0.49	1

STATEMENT TEXT FOR QUERY HASH: 0x54C102F2329C0C98

```
select top (@v) t.* from Production.TransactionHistory s where t.ProductID = @p ORDER BY s.TransactionDate option(optimize for (@v=?))
```

EXPLAIN PLAN FOR PLAN HASH: 0x31F605092B25E86A

Show plan objects for 0x31F605092B25E86A

- Database: adv_works
- Missing indexes
- /Missing index impact: 86.72% / use [adv_works]; create index [missing_index_TransactionHistory_03042018] on [Production].[TransactionHistory] ([ProductID]) include ((TransactionID),[Reference
- SELECT (Cost = 5.71621 , Rows = 0 , CPU = 0 , IO = 0)

Below the table with a list of queries is presented the full text of the query with the execution plan. Clicking on particular query will refresh these fields.

```

STATEMENT TEXT FOR QUERY HASH: 0X84C102F2332DC383

select top(50) v.* from Production.TransactionHistory v where v.ProductID = @p ORDER BY v.TransactionDate option(optimize for (@v=7))

EXPLAIN PLAN FOR PLAN HASH: 0X31F605092B25E86A

Show plan objects for 0x31F605092B25E86A
-Database: adw_works
-Missing indexes
  -/Missing index impact: 86.7274% use [adw_works]; create index [missing_index_TransactionHistory_05042018] on [Production].[TransactionHistory] ( ([ProductID]) ) include ([TransactionID],[ReferenceNumber])
-SELECT ( Cost = 3,07821, Rows = 0, CPU = 0, IO = 0 )
-Top ( Cost = 3,07821, Rows = 1000, CPU = 0,0001, IO = 0 )
  -Inner Join/Nested Loops ( Cost = 3,07821, Rows = 1000, CPU = 0,28926, IO = 0 )
    -Index Scan ([TransactionHistory].[IX_TransactionHistory_Date] [t]) ( Cost = 1,93701, Rows = 1000, CPU = 37,4685, IO = 48,9676 )
      -Clustered Index Seek ([TransactionHistory].[PK_TransactionHistory_TransactionID] [t]) ( Cost = 3,27739, Rows = 1, CPU = 0,001581, IO = 0,003125 )
-Plan Compilation Time: 1 ms
-Sampled values used for parameters at plan compilation time
  -@v: 1000
  -@p: 1

```

The following information is available in the explain plan:

- Name of the database in which the query is performed
- Optionally, information about the absence of an index, which is determined by the database optimizer based on index statistics,
- Algorithm of the explain plan
- List of parameters (example parameter values) used when compiling the first explain plan.

In the area of the explain plan, there is a link that allows you to perform the following operations:

- download the explain plan into an XML file
- Generating a plan guide script
- Generating a plan guide script with the query text and a filled list of parameters
- Change explain plan view to graphical

STATEMENT TEXT FOR QUERY HASH: 0XAF4660EAFCD24D07

```
SELECT [claims].* FROM [claims] WHERE [claims].[safo_invoice_id] = @0 AND ([claims].[safo_id] IS NOT NULL)
```

EXPLAIN PLAN FOR PLAN HASH: 0X040B21AB9C729F18

Show plan objects for 0x040B21AB9C729F18

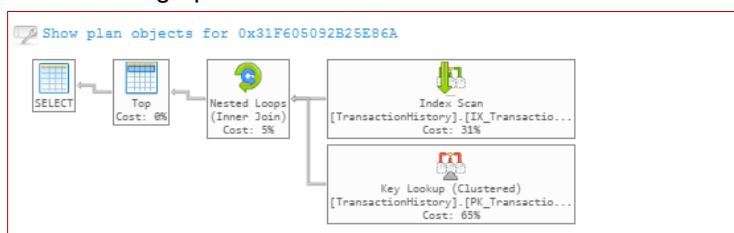
- Explain plan options
- Save to XML
- Generate plan guide script
- Show statement script with filled parameters
- Change view to graphical

Rows - 2.64165 , CPU - 0.0402102 , IO - 0)

Claims_3213E83F76FA0294] (Cost - 31.291 , Rows - 2.64165 , CPU - 0.00924682 , IO - 31.2816)

compilation time

Below the plan of execution in a graphic format:



In the upper part of the explain plan window there is a link [Show plan objects for ...](#) to the functionality that allows its analysis and analysis of objects participating in the query, among others:

- what tables, indexes participated in the execution of the query
- how the engine referred to the given objects
 - searching for data (seek)
 - reading full data (scan index or table)
- whether the query was performed in multithreaded mode
- what mechanism was used to download and connect "data" from objects:
 - Nested Loop
 - Hash / Merge Join connection

Clicking the **[Show Plan Object]** link, presents User with a form of used queries.

SQL TEXT

```
select top(10) t.* from Production.TransactionHistory t where t.ProductID = @p ORDER BY t.TransactionDate option(optimize for (@p=?))
```

EXPLAIN PLAN

```

Database: adv_works
Missing indexes
Missing index input: 56.7274*/* use [adv_works] create index [missing_index_TransactionHistory_05042]
SELECT ( Cost = 3.07811 , Rows = 0 , CPU = 0 , IO = 0 )
Top ( Cost = 3.07811 , Rows = 1000 , CPU = 0.0001 , IO = 0 )
Inner Join-Nested Loops ( Cost = 3.07811 , Rows = 1000 , CPU = 0.0001 , IO = 0 )
Index Scan ((TransactionHistory).[IX_TransactionHistory].[IX_TransactionHistory].[IX_TransactionHistory] [t]) ( Cost = 1.89701 , Rows = 1000 , CPU = 0.0001 , IO = 0 )
Clustered Index Seek ((TransactionHistory).[PK_TransactionHistory].[PK_TransactionHistory].[PK_TransactionHistory] [t]) ( Cost = 3.07811 , Rows = 1000 , CPU = 0.0001 , IO = 0 )
  
```

OBJECTS USED IN EXPLAIN PLAN

Type	Owner	Object Name	Table Name	Database
index	[Products]	[IX_TransactionHistory_Date]	[TransactionHistory]	[adv_works]
index	[Products]	[PK_TransactionHistory_TransactionID]	[TransactionHistory]	[adv_works]
table	[Products]	[TransactionHistory]	[TransactionHistory]	[adv_works]

INDEXES FOR SELECTED OBJECT [PRODUCTION].[TRANSACTIONHISTORY]

Index name	Enabled	Index columns	Included columns	Seeks	Scans	Lookups	Updates
IX_product_quantity	✓	ProductID	Quantity	18 074	0	0	0
IX_TransactionHistory_De	✓	TransactionDate, ProductID		0	25 698	0	0
IX_TransactionHistory_Ph	✓	ProductID		880 385	0	0	0
IX_TransactionHistory_Re	✓	ReferenceOrderID, ReferenceOrderID					
IX_TransactionHistory_Su	✓	StatusID		757 762	0	0	615
IX_TransactionHistory_Tri	✓	TransactionDate					

Object columns

Column	Included	Type	Max Length	Position	Is identity	Is computed	Is sparse	Density	Unique values	Rows sampled
TransactionDate	✓	datetime	8	1	✓	✓	✓	0.00000690	144 955	34 052 112
ProductID	✓	int	4	2	✓	✓	✓	0.00000775	129 024	34 052 112

In **[Show Plan Objects]** we have information about the query text and the explain plan. Below we see areas such as:

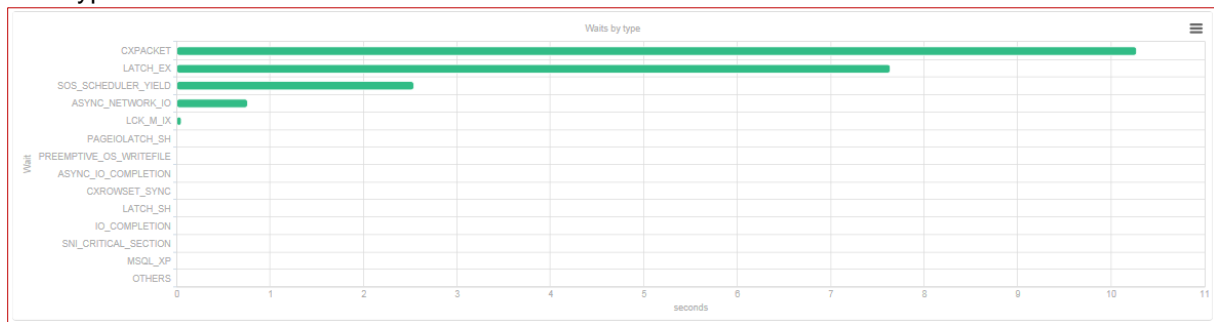
- **Objects Used in Explain Plan** – a list of all objects used by the query in the explain plan
- **Indexes for selected object** – list of indexes for selected table - row selected in the "Objects Used in Explain Plan" consists of 3 bookmarks:
 - a. **Object Columns** – a list of the individual columns of the selected object, along with information such as: column name, data type, column id, density (the lower density, the higher selectivity of the column)
 - b. **Info** – basic information about selected object (DDL info)
 - c. **Properties** – additional properties of selected object

Info tab and Properties Tab are visible when checkbox "Load object properties (slower)" is selected.

When analyzing the explain plan, we pay attention to:

- Limiting the choice of data, or of the data with the where clause and table joins
- Whether the request is with parameters or literals
- The operation the SQL instance engine chose to retrieve/download data
- Whether the table has appropriate indexes
- Way of reading the data - Nested Loops vs. Hash Join

Another tab (next to **[SQL Statements]**) is **[Waits]**. Waits presented here are shown in graphical and table form. The graph shows the duration for each second of the selected snapshot (of 15 minutes) for each type of wait that occurred at the time of the instance.

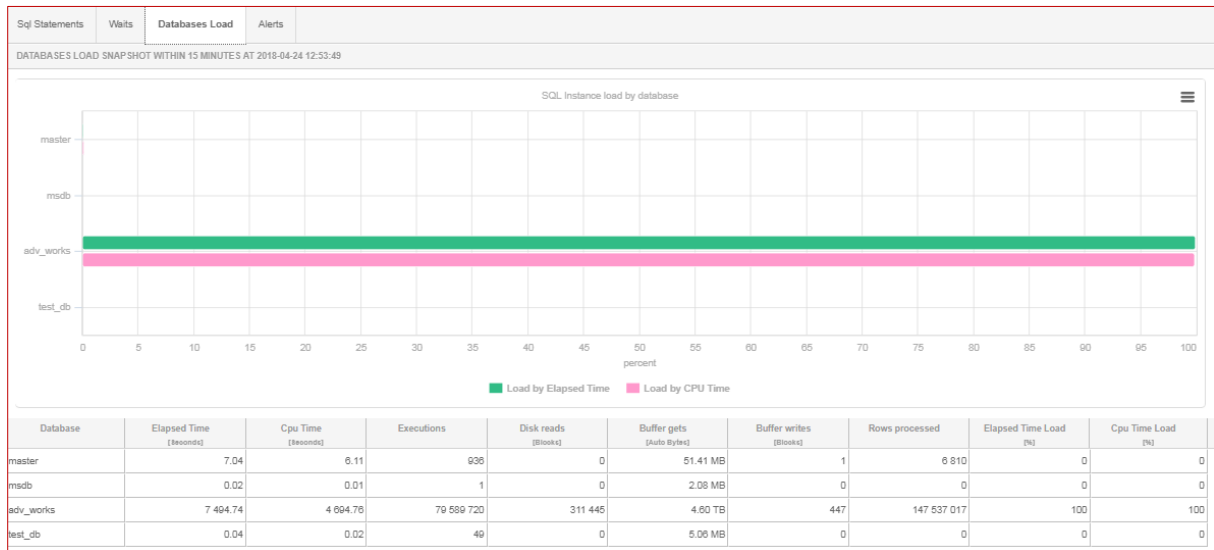


A table is located below the graph with following columns:

- Name - the name of wait
- Wait time - per 1 second (sec.) - Duration of wait type in seconds
- Total wait time for snapshot (sec.) - the total duration of wait type in the snapshot (15 minutes)

Search by any value in below waits table			
Name	Wait time per 1 second (seconds)	Total wait time for snapshot (seconds)	
LCK_M_S	8.2580	7.432.484	
ASYNC_NETWORK_IO	1.0330	929.636	
LCK_M_U	0.3430	308.728	
LATCH_EX	0.3380	304.241	
HADR_SYNC_COMMIT	0.0910	81.773	

The Databases Load Tab presenting the load from the point of view of the databases to which the queries were generated.



Next tab is **Alerts**. A list of alerts that occurred during this snapshot.

Sql Statements Waits Databases Load Alerts

SNAPSHOT OF ALERTS EXECUTED WITHIN 15 MINUTES AT 2018-04-30 20:45:17

Logdate	Alert name	Message
2018-04-30 20:45:15	Elapsed Time per 1 e	Alert Type: Sql Query, Alert level: CRITICAL, The measured statistic value is 12.1 times higher than average , Statement query hash: 0x64C102F23329DC98, Statistics: Elapsed Time per 1 exec, Last value: 3.60 s, History value: 0.198
2018-04-30 20:45:15	Elapsed Time	Alert Type: Sql Query, Alert level: CRITICAL, The measured statistic value is 138 % higher than average , Statement query hash: 0x64C102F23329DC98, Statistics: Elapsed Time, Last value: 1725 s, History value: 361.8 s , Faster plan
2018-04-30 20:45:15	Wait Time	Alert Type: Load Trends, Alert level: CRITICAL, The measured statistic value is 115 % higher than average , Last value: 795.5 s, Reference history value: 369.0 s
2018-04-30 20:45:15	Elapsed Time	Alert Type: Load Trends, Alert level: WARNING, The measured statistic value is 99 % higher than average , Last value: 3341 s, Reference history value: 1681 s
2018-04-30 20:45:15	Cpu Time	Alert Type: Load Trends, Alert level: WARNING, The measured statistic value is 44 % higher than average , Last value: 2511 s, Reference history value: 1742 s

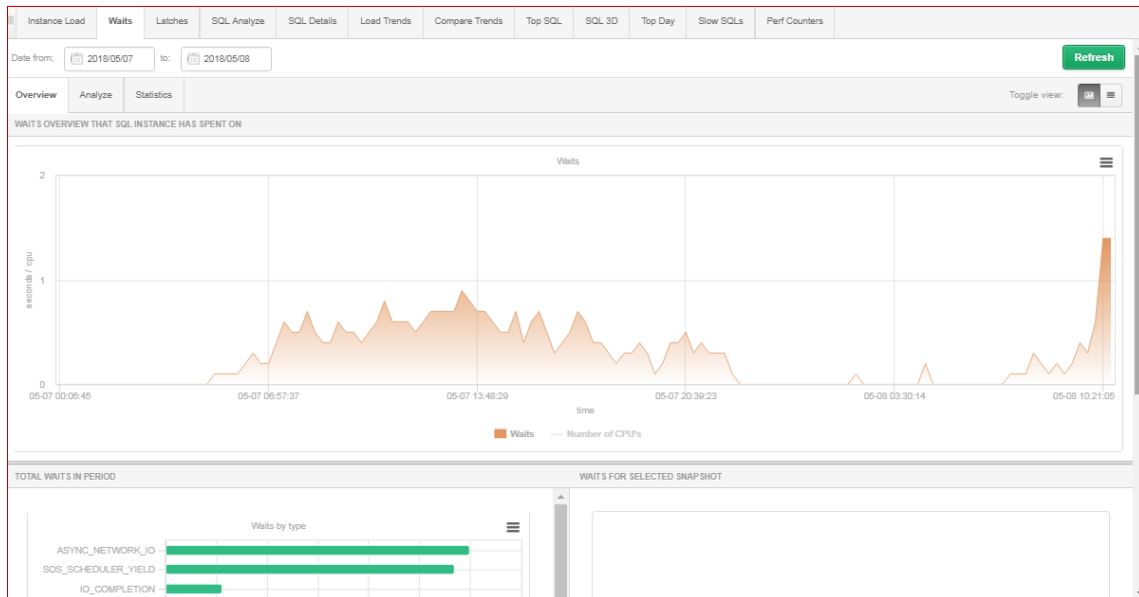
6.2.1.2 Waits Tab

Waits tab shows the duration of waits, which occurred at a time for all sessions on SQL Instance. Depending on the selection of **[Toggle View]** option, data can be shown graphically or in table form.

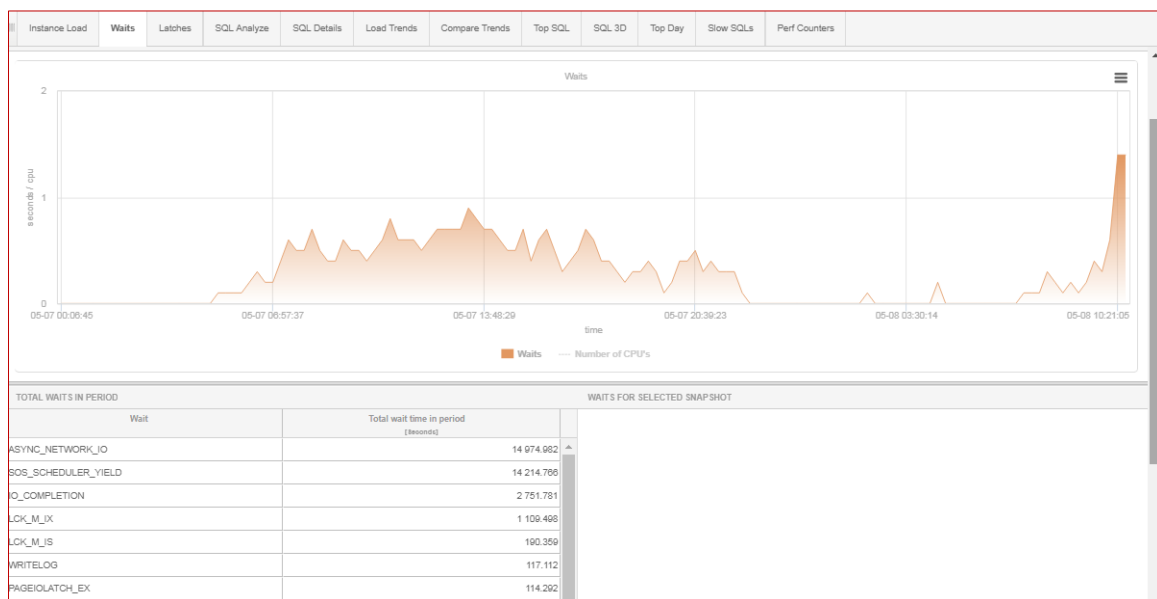
Waits screen in a similar way to **[Instance Load]** screen, consists of the following fields:

- filtration fields - fields of dates by which we define the period in which we see database' waits
- graph presenting the level of waits for specific time (on the left) and for snaps (on the right).
- detailed information about waits in a given moment of time

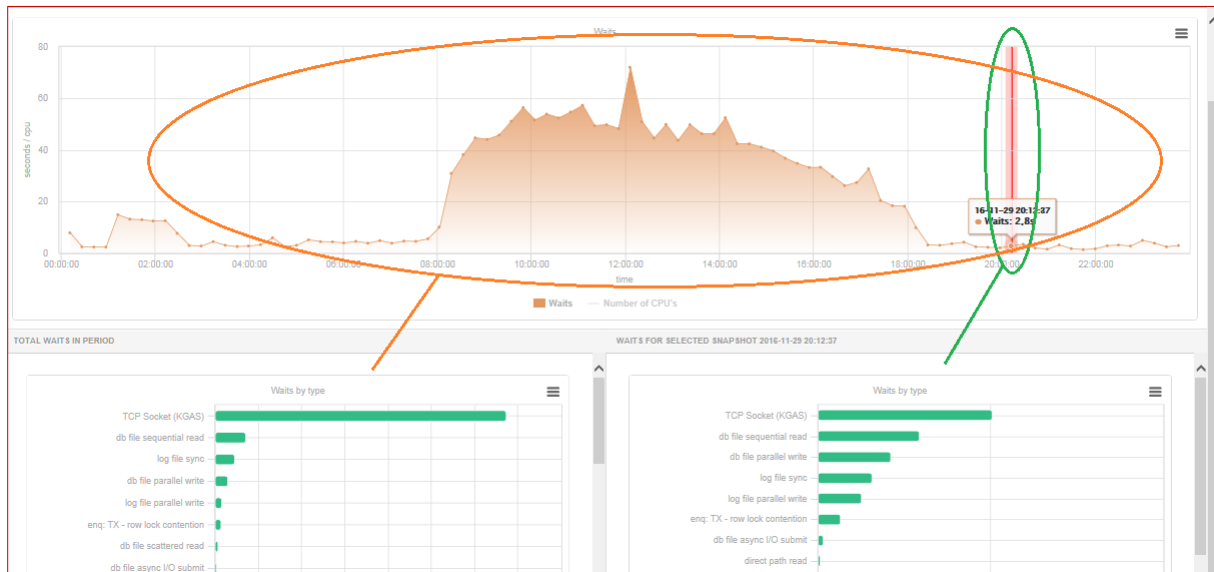
The Y axis of the graph illustrates (in seconds for a given second) time of all waits that occurred during the period shown on the X-axis. X-axis of the graph shows the period in which waits occur.



After switching the view with **[Toggle view]** button we get we get detailed information about the duration of the type of wait.



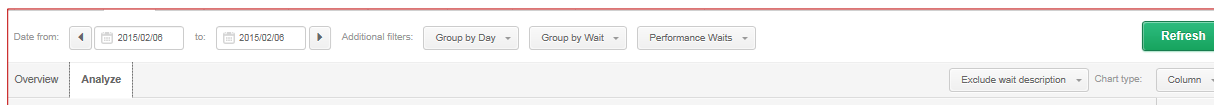
Similar like in the screen **[Instance Load]**, **[Waits]** chart is "clickable". Clicking on the part of the graph (its point) will show us waits summary, appropriate for a snapshot in time.



From the above chart we can see:

- what the SQL instance doing during the day (the default) or a selected period limited by dates in the filter
- what SQL Instance doing during the last snap

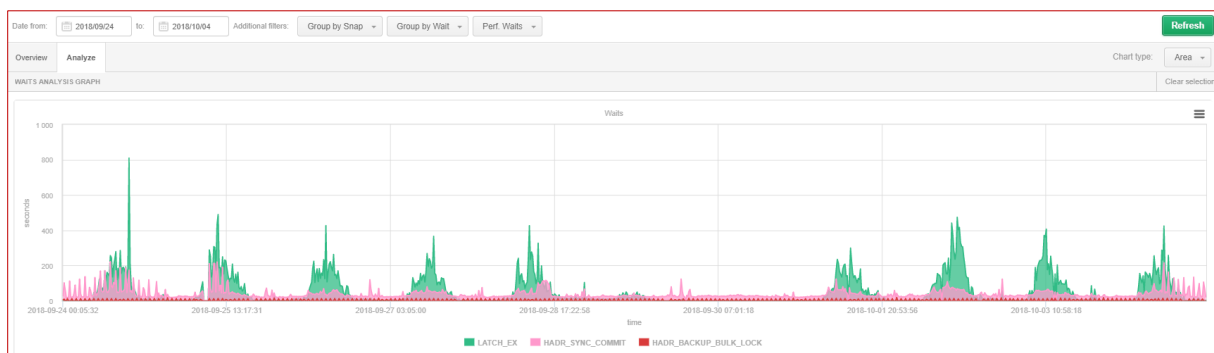
The system also allows to analyze individual Waits - the frequency, length and time of occurrence. To do this, click on the subtab **[Analyze]**:



On this page User can analyze performance waits or all of waits grouped by:

- Perf. Waits or All Waits,
- day, hour, snapshot.
- Group by Wait Class

An example results of analysis we present below:

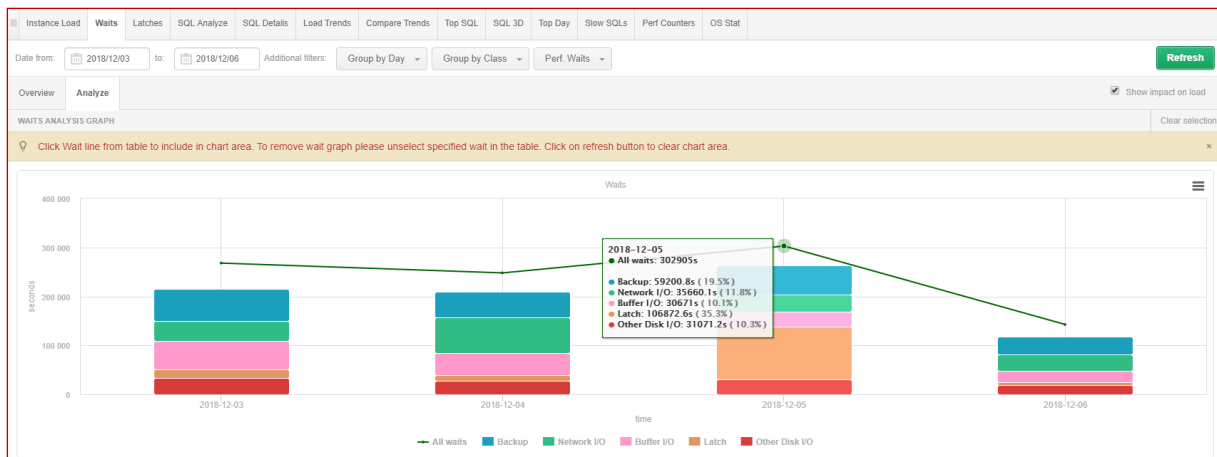


Data is transferred on the chart area by clicking on the selected wait's row or dragging it to the empty field below. A view showing the class of top waits divided into days:

time			
		LATCH_EX	HADR_SYNC_COMMIT
		LCK_M_S	ASYNC_NETWORK_IO
Waits Statistics			
Search wait by any value from below table column			
wait	Description	Total wait time in period (seconds)	Load (%)
ASYNC_NETWORK_IO	Occurs on network writes when the task is blocked behind the network. ->Verify that the client is processing data from the server.	490 832.162	78.8
LATCH_EX	Occurs when waiting for an EX (exclusive) latch. ->This does not include buffer latches or transaction mark latches. ->A listing of LATCH_ waits is available in the SQL Server Enterprise Manager. ->This wait is also reflected by the Transaction D.	55 598.959	8.9
HADR_SYNC_COMMIT	Waiting for transaction commit processing for the synchronized secondary databases to harden the log. ->This wait is also reflected by the Transaction D.	41 048.917	6.6
LCK_M_S	Occurs when a task is waiting to acquire a Shared lock.	15 808.681	2.5
HADR_BACKUP_BULK_LOCK	The AlwaysOn primary database received a backup request from a secondary database and is waiting for the background thread to finish processing the request.	3 384.332	0.5
ASYNC_IO_COMPLETION	Used to indicate a worker is waiting on an asynchronous I/O operation to complete not associated with database pages.	3 046.367	0.5

Group waits by class

Depending on the period you have selected, you can also view the graph of the share of a given valid class in the total number of waiters. The functionality works for grouping after performance wait and for all waitresses.

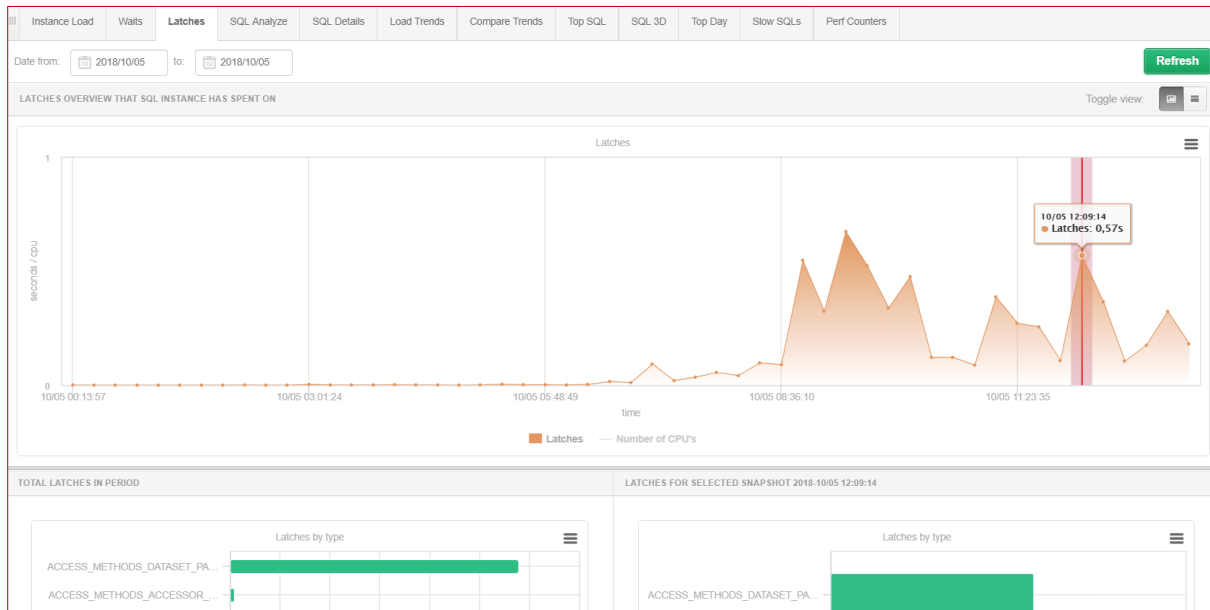


Information is also available in table form under the graph. The data in the table are presented for the entire selected period.

Waits Statistics			
Search wait by any value from below table column			
Class	Total wait time in period (seconds)	Load (%)	
Backup	213 184.439	22.2	
Network I/O	181 885.408	18.9	
Buffer I/O	158 329.403	16.5	
Latch	141 108.972	14.7	
Other Disk I/O	111 287.741	11.6	
Lock	69 209.978	7.2	
Cpu	58 415.465	6.1	
Tran Log I/O	24 044.539	2.5	
Buffer Latch	2 846.919	0.3	

6.2.1.3 Latches Tab

The Latches tab shows the duration of all LATCHES in seconds, which occurred at a given time for all user' sessions of SQL instance.



Latch screen maintains almost the same functionality as the Waits screen (click-through charts, all latches presentation and latches at a specific point in time).

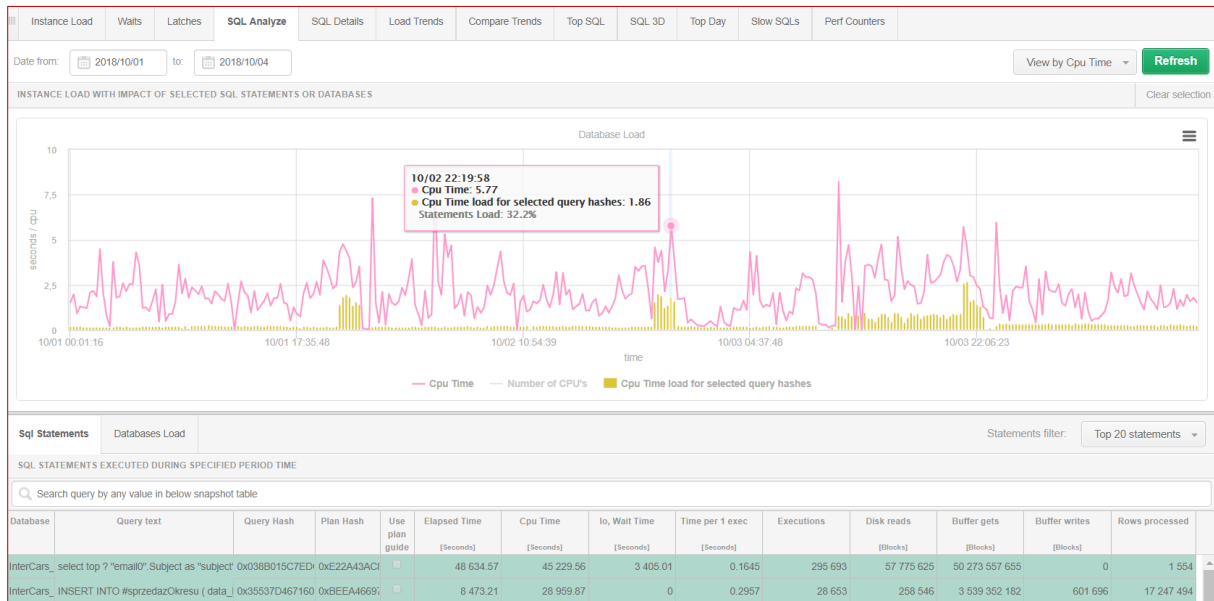
The Y axis graph shows the time in seconds of all LATCHES that occurred at the time shown on the X-axis. The X-axis of the graph shows the time in which LATCHES occur. The chart is active and refreshes every 15 minutes by clicking the [Refresh] button.

6.2.1.4 SQL Analyze Tab

SQL Analyze functionality presents an additional view of SQL Instance load. As with the **instance Load** chart, graph here also shows the utilization of the base.

The screen consists of the following areas:

- filtration fields:
 - date and time fields by which the time is determine in which users can to familiarize themselves with the instance load
 - way of the presentation of the load - by CPU Time or Elapsed Time
- the graph shows the CPU load or Elapsed Time (depending on the choice in the filter)
- cumulative statistics:
 - with Group by plan option unchecked – broken down into queries that generated a specific load in a given period
 - with Group by plan option checked – broken down into performance plans that generated a specific load in a given period



Graphs Y-axis shows the number of seconds for each second of duration of the query in SQL instance database.

The X-axis represents the time at which the query caused the utilization of SQL Instance server. Differences that can show up between the load shown in the Instance Load graph, and utilization statistics of the SQL Instance server from the operating system side, arise due to including in the chart all kinds of waits, which is not shown in the operating system. The graph shows a full picture or performances, not just time.

After clicking on certain number of queries, Users can see their share of Instance load and when it took place with an accuracy to 15 minutes:

The table in the SQL Statements tab shows statistics for each query:

- Query text – content of the query
- Hash Value – the query ID
- Plan Hash – execution plan ID
- Elapsed Time (sec) – summary of time in seconds of query duration for all queries executions taking place in the selected time. If the filter is marked "Group by Plan," then the Time of duration (sec) indicates the duration of all searches for the PLAN HASH VALUE for a selected period.
- CPU time (sec) – summary of time in seconds of CPU utilization for all executions of the query taking place in the selected time interval. If the filter is marked "Group by Plan," time of duration (sec) indicates the duration of all searches for the HASH VALUE PLAN for a selected period.
- I/O Wait Time [Seconds] – total time in seconds of waiting for I/O for all query executions taking place in the selected time interval. If the "Group by Plan" filter is selected, then Time of duration (sec) shows the duration of all queries for a given PLAN HASH in the selected time interval
- Executions - The number of query executions in the selected time
- Disk Reads - The number of disks reads for a query in the selected time
- Buffet Gets - The number of buffers utilized for a query in the selected time
- Rows processed - Number of rows returned by the query in the selected time

IMPORTANT: SQL Analyze screen maintains similar functionality to the Instance Load:

- Clicking on a query row (in addition to showing the load on the graph) will display the full text of the query and its execution plan
- Next to the query identifier the [+] [Plus] button is located, which adds a query to the clipboard with a list of queries

If the Group by plan option is checked - the [Plus] button adds the hash plan identifier to the SQL Plan tab

Database	Query text	Query Hash	Plan Hash	Elapsed Time [seconds]	Cpu Time [seconds]	Io, Wait Time [seconds]	Time per 1 exec [seconds]	Executions	Disk reads [Blocks]	Buffer gets [Blocks]	Buffer writes [Blocks]	Rows processed
adv_works	select * from (select p.ProductID, p.ProductName, p.Name, p.SafetyStockLevel, Production.GetSafetyStockQuantity(p.ProductID) as TotalQuantity from Production.Product p) p order by p.TotalQuantity desc	0x0FE4759073F13D6	0xEBDB345610166111	52.688.56	10.146.45	7.0362	8.930	14.191.088	5.972.053.598	0	1.152.184.320	
adv_works	SELECT Production.GetProductName(Production.ProductID) as ProductName from Production.Product	0x0FE4759073F13D6	0xEBDB345610166111	34.969.05	9.000.75	0.0396	1.111.062	6	3.593.174.512	0	1.183.281.030	
adv_works	select top (@v) t.* from Production.Transaction	0x64C102F23326	0xF02EB8803	25.038.17	2.733.73	2.4975	11.120	13.305	1.663.703.499	0	148.496.480	
adv_works	select @qty = isnull(sum(Quantity),?) from Production.Transaction	0xF9C0C67E4D5	0x0305CE48	22.213.79	2.491.35	0.0000	1.152.270.632	1.312	3.603.055.078	0	1.152.270.632	
adv_works	select @name = Name from Production.Product	0x797529C73920	0xD3EB49A8	21.433.09	17.879.60	3.554.09	0.0000	1.183.281.030	0	3.583.174.950	0	1.183.281.030
adv_works	select @qtyPl = isnull(sum(Quantity),?) from Production.Transaction	0x57F2C80FBBB	0x524FD52C	11.956.40	10.688.67	1.287.73	0.0000	1.152.270.631	4	2.304.648.434	0	1.152.270.631
adv_works	select top (@v) t.* from Production.Transaction	0x64C102F23326	0xF02EB8803	13.870.48	8.800.91	5.069.57	0.1663	83.400	15.894	4.738.502.405	0	1.113.723.600
adv_works	select top ? * from Production.TransactionHistory	0x677E3020F458	0x0305CE48	7.487.09	6.475.66	1.011.43	0.0148	506.487	2.124	444.695.588	0	506.659

STATEMENT TEXT FOR LAST SELECTED QUERY HASH: 0x0FE4759073F13D6

select * from (select p.ProductID, p.ProductNumber, p.Name, p.SafetyStockLevel, Production.GetSafetyStockQuantity(p.ProductID) as TotalQuantity from Production.Product p) p order by p.TotalQuantity desc

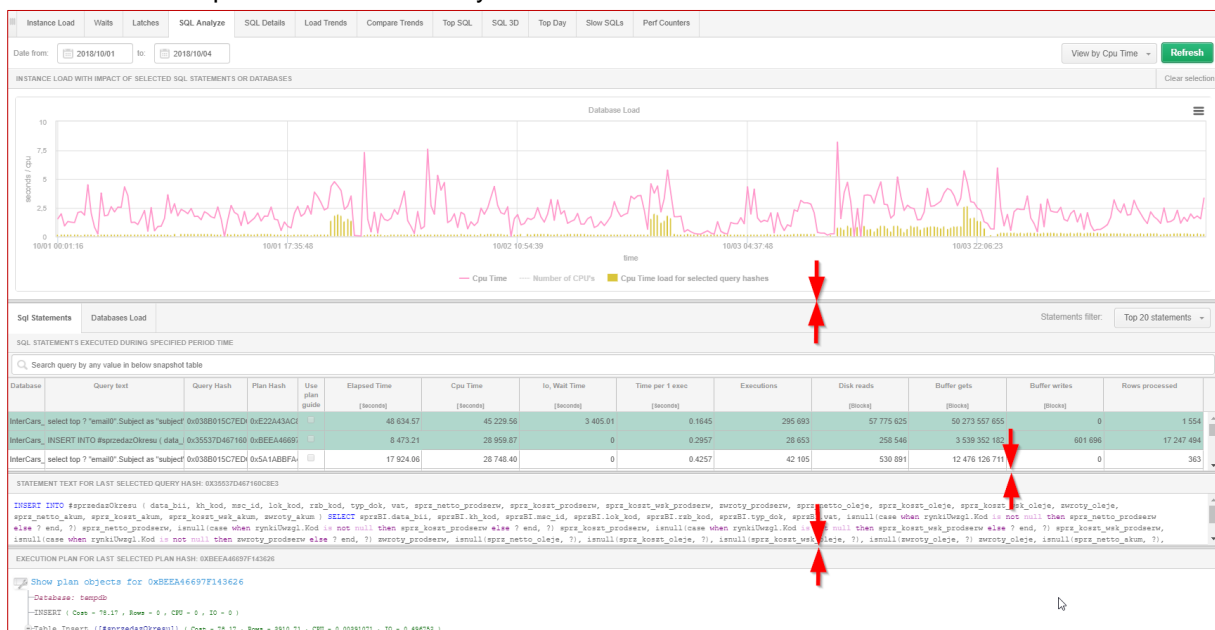
EXECUTION PLAN FOR LAST SELECTED PLAN HASH: 0xEBDB345610166111

Show plan objects for 0xEBDB345610166111

Database: adv_works
--SELECT (Cost = 12,8667 , Rows = 0 , CPU = 0 , IO = 0)
--Sort (Cost = 12,8667 , Rows = 129024 , CPU = 10,0944 , IO = 0,0112613)
--Compute Scalar (Cost = 2,80107 , Rows = 129024 , CPU = 0,0129024 , IO = 0)
--Table Scan ([Product] [p]) (Cost = 2,78817 , Rows = 129024 , CPU = 0,142005 , IO = 2,64617)

It is worth noting that for individual components of the screen you can change the height - this applies to min. charts, data tables, query text controls, execution plan.

Below is an example from the SQL Analyze screen:



6.2.1.5 SQL Details Tab

SQL Details tab shows detailed information about the query such as:

- frequency,
- execution time,
- whether the request has changed execution plan,
- the number of returned records,
- the number of executions,
- role of query in load of SQL Instance,

This information provides the opportunity to decide whether it makes sense to optimize given query

The screenshot shows the 'SQL Details' tab in DBPLUS. At the top, there are navigation tabs: Instance Load, Waits, Latches, SQL Analyze, SQL Details (selected), Load Trends, Compare Trends, Top SQL, SQL 3D, Top Day, Slow SQLs, and Perf Counters. Below these, a search bar contains the query hash '0x64C102F23329DC98'. Filters for 'From' (2018/05/02 00:00) and 'to' (2018/05/08 23:59) are set. A 'Group by plan hash' checkbox is checked. Buttons for 'Refresh' and 'Find SQL' are present. The 'STATEMENT TEXT' section shows a SQL query: `select top (@v) t.* from Production.TransactionHistory t where t.ProductID = @p ORDER BY t.TransactionDate option(optimize for (@v=?))`. Below this, the 'SQL STATISTICS' section shows a table with columns: Date, Plan hash, Elapsed Time, Cpu Time, Rows processed, Executions, Disk Reads, Disk Reads, Buffers Get, Buffers Write, Buffer Quality, and Elapsed Time per 1 Exec. The table contains five rows of data for different dates and plan hashes. At the bottom, the 'Explain plan' section shows a tree view of the execution plan, including 'Missing indexes', 'SELECT', 'Top', 'Sort', and 'Inner Join-Nested Loops'.

[SQL Details] tab is divided into several areas:

Clipboard with a list of query IDs (expanded and collapsed by clicking the green button in the lower right corner of the screen) - the queries to the clipboard are added from screens:

- Performance ->Instance Load
- Performance ->SQL Analyze
- Performance ->Top SQL
- Performance ->SQL 3D
- Performance ->Top Day
- Performance ->Slow SQL

The screenshot shows a clipboard window titled 'Query Hashes list'. It contains a list of query hashes: 0xDF1D6547F4EC5B93, 0x0FE47590673F13D6, 0x2F5EE731FCEDF74A, and 0x64C102F23329DC98 (highlighted with a green checkmark). A green button with a clipboard icon is in the top right corner. A yellow tooltip says 'Click on query hash to analyze Query Performance Details'. At the bottom, there are buttons for 'Clear list', a trash icon, a copy icon, and a green button with a clipboard icon.

IMPORTANT: List of queries is remembered under the SQL Instance for specified user. That list can be saved to file or opened again.

Filters area and the way to display statistics for:

- specified Query Hash / Hash Value or SQL ID of the query
- selected date range
- a grouping of statistics by day, snapshot, etc.
- navigation buttons which allow to refresh the screen, to search another query or to show the statistics of queries in report.

The screenshot shows the filters area at the bottom of the DBPLUS interface. It includes a search bar with the query hash '0x64C102F23329DC98', 'From' and 'to' date/time filters (2018/05/02 00:00 to 2018/05/08 23:59), a 'Group by plan hash' checkbox, a 'Group by Day' dropdown, an 'Online values' checkbox, and 'Refresh' and 'Find SQL' buttons.

IMPORTANT: Selecting [checkbox Online Values] – some filters are hidden by default and clicking the [Refresh] button will present statistic of specified query according to information which are available in sys.dm_exec_query_stats system view.

Instance Load Waits Latches SQL Analyze SQL Details Load Trends Compare Trends Top SQL SQL 3D Top Day Slow SQLs Perf Counters

0x0CFC36787A5B452 ☒ Online values Refresh Find SQL

STATEMENT TEXT

```
select distinct lck_kod from [IT].[CRM].[v_sp_det_rec] s
where lck_kod in ('118666','497763') and
s.rnk_pcsidco = @data_cd
and s.lck_maxlcknt = @data_do
order by lck_kod
```

SQL STATISTICS ☐ Show values per 1 executions

Plan hash	Elapsed Time [seconds]	Cpu Time [seconds]	Rows processed	Executions	Disk Reads [pages]	Disk Reads [MB]	Buffers Get [pages]	Buffers Write [pages]	Buffer Quality [%]	Version count	Gen. Num	Elapsed Time per 1 Exec [seconds]
0x0CFC36787A5B452	596.1	473.1	45	44	64	0.50 MB	77 885 641	2	100.0	17	17	11.5620

Explain plan Graph

Show plan objects for 0x0CFC36787A5B452

```
--Database: IT
--Missing indexes
/*Missing index impact: 92.6817% use [IT]; create index [missing_index_SPREEDAR_FAKTY_LOV_OI_PROD_10042018] on [CRM].[SPREEDAR_FAKTY_LOV_OI_PROD] ( [lck_kod], [lck_minlcknt] ) include ([lck_kod], [lck_maxlcknt]);
/*Missing index impact: 92.6817% use [IT]; create index [missing_index_SPREEDAR_FAKTY_LOV_OI_PROD_10042018] on [CRM].[SPREEDAR_FAKTY_LOV_OI_PROD] ( [lck_kod], [lck_minlcknt] ) include ([lck_kod], [lck_maxlcknt]);
--SELECT ( Cost = 1870.34 , Rows = 0 , CPU = 0 , IO = 0 )
```

➤ Area with query text– with a scroll ability – convenient for longer query content

STATEMENT TEXT

```
select top (@v) t.* from Production.TransactionHistory t where t.ProductID = @p ORDER BY t.TransactionDate option(optimize for (@v=?))
```

➤ Detailed execution statistics in form of the table

SQL STATISTICS ☐ Show values per 1 executions

Date	Plan hash	Elapsed Time [seconds]	Cpu Time [seconds]	Rows processed	Executions	Disk Reads [Blocks]	Disk Reads [GB]	Buffers Get [Blocks]	Buffers Write [Auto Bytes]	Buffer Quality [%]	Elapsed Time per 1 Exec [seconds]
2018-05-02	0x31F605092B25	48 325.2	43 857.5	265 744 600	19 900	13 606	0.104 GB	2 977 325 827	0 B	100.00	2.428402
2018-05-04	0x31F605092B25	99 643.9	88 805.1	529 739 826	39 666	12 632	0.096 GB	5 934 331 201	0 B	100.00	2.512074
2018-05-08	0x31F605092B25	30 058.9	27 162.9	161 269 612	12 078	13 305	0.102 GB	1 807 024 611	0 B	100.00	2.488728
2018-05-03	0xF02EB8B03876	6 460.1	4 324.0	593 945 858	44 466	4 488	0.034 GB	2 526 405 953	0 B	100.00	0.145283
2018-05-07	0xF02EB8B03876	13 870.5	8 800.9	1 113 723 600	83 400	15 894	0.121 GB	4 738 502 405	0 B	100.00	0.166313

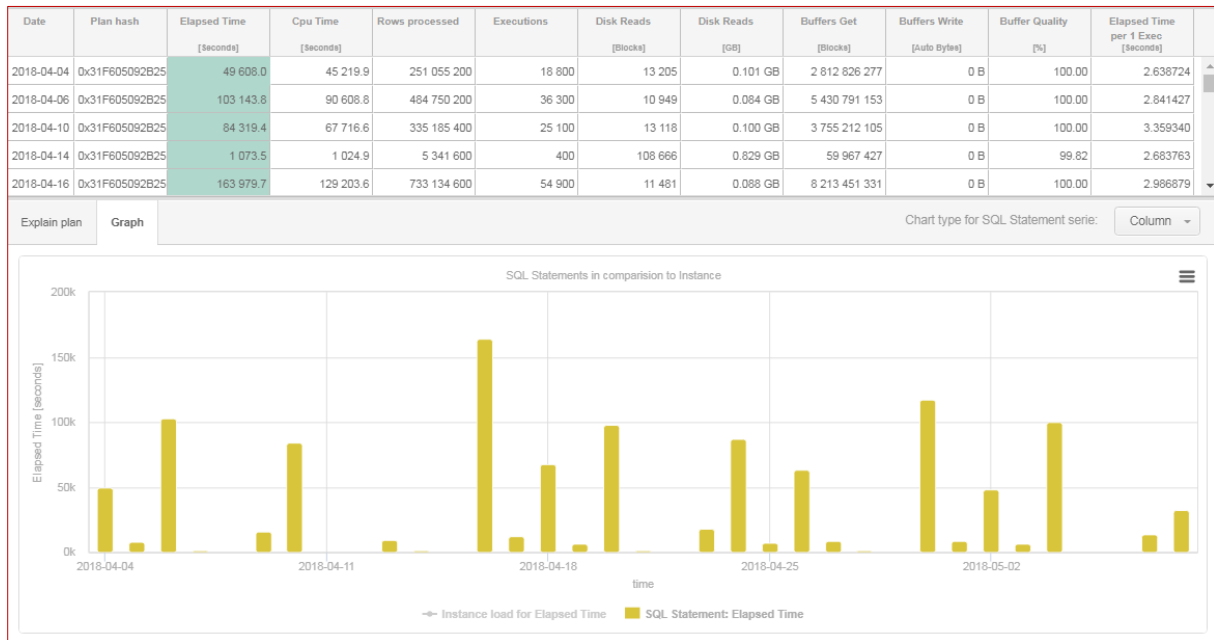
➤ Execution plan (with the [Explain Plan] tab selected)

Explain plan Graph Compare Plans 0xF02EB8B03876CBD9

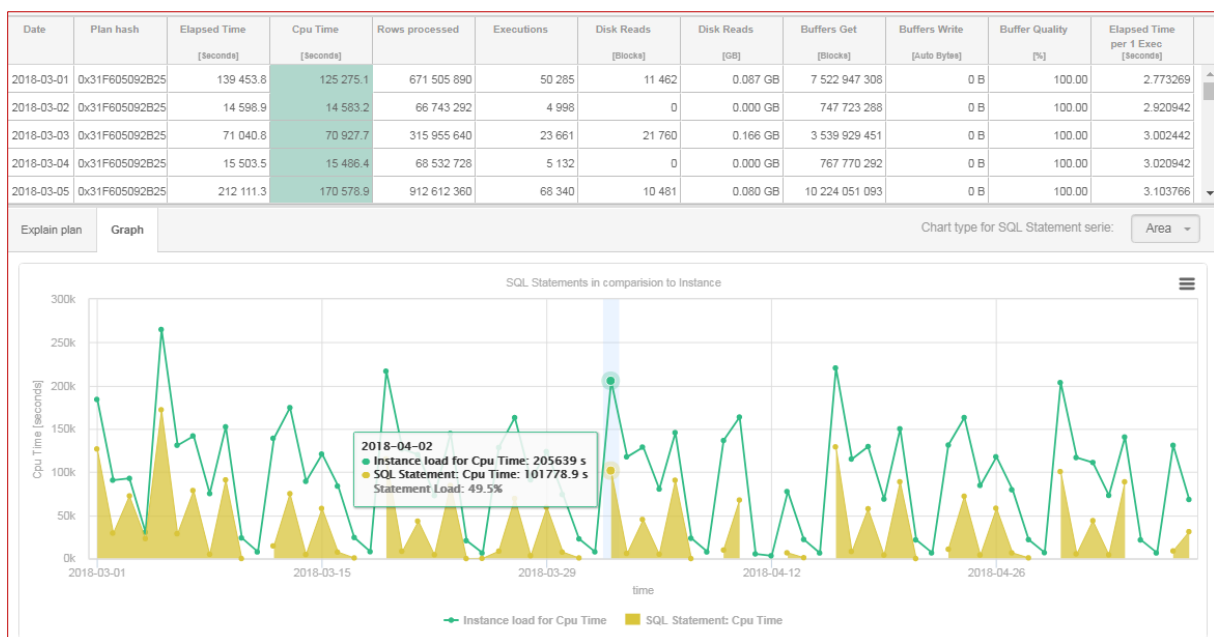
Show plan objects for 0xF02EB8B03876CBD9

```
--Database: adv_works
--Missing indexes
/*Missing index impact: 97.4518% use [adv_works]; create index [missing_index_TransactionHistory_05082018] on [Production].[TransactionHistory] ( [Prod
--SELECT ( Cost = 204,369 , Rows = 0 , CPU = 0 , IO = 0 )
--Top ( Cost = 204,369 , Rows = 1000 , CPU = 0.0001 , IO = 0 )
--Sort ( Cost = 204,369 , Rows = 1000 , CPU = 5.09024 , IO = 0.0112613 )
--Inner Join-Nested Loops ( Cost = 199,328 , Rows = 69244 , CPU = 0.28526 , IO = 0 )
--Index Seek ([TransactionHistory].[IX_TransactionHistory_ProductID] [t]) ( Cost = 0.165074 , Rows = 69244 , CPU = 0.0752254 , IO = 0.0894689 )
--Clustered Index Seek ([TransactionHistory].[PK_TransactionHistory_TransactionID] [t]) ( Cost = 199,878 , Rows = 1 , CPU = 0.0001581 , IO = 0.008125 )
--Plan Compilation Time: 11 ms
--Sampled values used for parameters at plan compilation time
--@v: 1000
--@p: 1
```

➤ Graphical presentation (With the Graph Tab selected) of any indicators/column from the statistic table



By clicking the SQL Statement Load tab, we can see the load generated by the given query (line / yellow area) against the background of the total instance load:



On the SQL Details screen, the User had the option of change the table presentation by check additional checkboxes. In the latest version of changes, grid is possible after select the appropriate item in the Grid view list:

- General statistics (default view),
- Statistics per 1 exec (view with columns converted into a single execution),
- Additional time Details (view contains information about interrupted queries).

Instance Load	Waits	Latches	SQL Analyze	SQL Details	Load Trends	Compare Trends	Top SQL	SQL 3D	Top Day	Slow SQLs	Perf Counters	OS Stat
0xA7C62AEF5C46008D From: 2020/04/01 00:00 to: 2020/04/01 23:59 <input checked="" type="checkbox"/> Connect queries with the same text <input checked="" type="checkbox"/> Group by plan Group by Snap <input type="checkbox"/> Online values Refresh Find SQL												
STATEMENT TEXT View Session History Print Preview Q Format SQL												
<pre>SELECT k.*, ROUND(isnull(przychod_netto,0),2) przychod_netto into #karta_obrot FROM #karty k LEFT JOIN (SELECT s.kh_kod, s.kraj_kod kraj_kod, s.wal_kod, SUM(s.sprz_netto) przychod_netto FROM it.crm.SPRZEDA2_FAKTY_01 s WHERE s.mec_naz = @mec_naz AND s.rab_kod IN (SELECT rab_kod FROM it.crm.BIIC_RIB WHERE grp_rab_kod IN (7,7)) --(7, 7, 7, 7, 7, 7) AND s.G1_KOD NOT IN (7,7) GROUP BY s.kh_kod, s.kraj_kod, s.wal_kod) o ON o.kh_kod=k.accountnumber and k.kraj_kod = o.kraj_kod and k.wal_kod = o.wal_kod</pre>												
SQL STATISTICS Grid view: General statistics Show details Additional time details												
Date	Plan hash	Elapsed Time [Seconds]	Cpu Time [Seconds]	Rows processed	Executions	Disk Reads [Blocks]	Disk Reads [MB]	Buffers Get [Blocks]	Buffers Write [Blocks]	Buffer Quality [%]	Gen. Num	
2020-04-0...	0xB4FFE13C6...	69.21	112.3	172 177	1	2 878 198	22 486 MB	6 450 629	2 650	69.2	1	69.2096280
2020-04-0...	0xB4FFE13C6...	179.98	140.9	172 211	1	6 443 157	50 337 MB	6 450 623	2 650	50.0	1	179.9700670

After choosing the right view, changing the settings for columns and table is analogous to that described earlier for Load Trends. Changes can be defined for each view separately.

[Enter Query Hash] displays statistics for given query identifier according to set filters.

0x64C102F23329DC98	From: 2018/04/04 00:00 to: 2018/05/08 23:59 <input checked="" type="checkbox"/> Group by plan hash Group by Day <input type="checkbox"/> Online values Refresh Find SQL
STATEMENT TEXT	

IMPORTANT: If you do not know and do not have any query ID and clipboard with a list of queries is empty, you can:

- go to one of the screens (Instance Load, SQL Analyze, Top SQL, Top Day), where you can search suboptimal/long-lasting query
- click on the **[Find SQL]** to search for a specific query (search by its text)

Statistics show:

- Plan Hash- query plan identifier
- Elapsed Time (sec) - total time in seconds for the duration of the query for the selected grouping period.
- CPU time (sec) - total time in seconds for CPU utilization for the query for the selected grouping period.
- Rows processed [Rows] - Number of rows processed by query for the selected grouping period
- Executions - The number of query executions for the selected grouping period
- Disk Reads [MB/Blocks] - Number of readings from the disk for a given query for the selected grouping period, broken down by the amount of MB and data blocks
- Buffet Get - The number of utilized buffers for a given query for the selected grouping period
- Buffer Write
- Buffer quality [%] – percentage of data that has been downloaded from the memory for a given query
- Elapsed time per 1 exec - duration of a single query execution for the selected period grouping

Filter **[Group by period]** - shows statistics for a given query grouped according to the choice:

- **No group by period** - no grouping i.e. selection date ranges from 1 to 20 days of the month will show summary statistics for the selected period
- **Month** – shows statistic for a given query, broken down by months
- **Day** - shows statistics for a given query, broken down into periods of one day
- **Hour** - shows statistics for a given query, broken down into periods of one hour
- **Snap** - shows statistics for a given query, broken down by snapshots - periods of 15 minutes

Online values option allows Users to display current information about queries stored in the SQL Instance buffer.

Statistics on queries in other tabs appear in 15 minutes and after activating the **[Show Online Values]** option, you can see them right away if only the query starts at that moment.

To check whether the query is performed or not at the moment, after supplying the Query Hash identifier, activate the Online Values checkbox and click the [Refresh] button to observe the value in the columns Execution, Elapsed Time, CPU Time. If the values change, it means that the query is still executing. If the values are fixed, the query has stopped running.


Explain Plan Tab

Shows the query explain plan. If there is more than one for the query then you can click on the Compare plans checkbox, which will display two explain plans - it makes it easier to find differences between them, which in turn are highlighted in yellow:

The control with the explain plan has the following available:

- Link with additional options for text and explain plan
- Link Show Plan Objects, which allows you to analyze the explain plan.

6.2.1.5.1 Explain plan options

In the area of the explain plan, there is a link  that allows you to perform the following operations:

- download the explain plan into an XML file
- Generating a plan guide script
- Generating a plan guide script with the query text and a filled list of parameters
- Change explain plan view to graphical

The first item from the pop-up menu allows you to download the execution plan in xml format - a plan in this form can be displayed in Microsoft Sql Studio.

Another feature is the Generate plan guide scripts - an option that is useful in situations where the database optimizer changes the execution plan for the worse. Very often, with the change of the plan, the time of query execution increases and the slow-running query can be the source of other

performance problems (e.g. prolonging transactions, increase blockages, increase reads from disk devices).

Create PLAN GUIDE for the query, we permanently set a specific execution plan with which the query should run.

After selecting the Generate plan guide scripts, a form with selection options is shown, on which depends the further form of the script.

In the form, the user has the following information available, including:

- Query text
- Statement type, we can choose:
 - From the application level
 - From the level of the base object (e.g. procedure, function, trigger)
 - From the T-SQL block
- The name of the database
- List of parameters or the name of the object from which the query is run
- Number of query identifiers
 - if there are more than 1 it can mean that the query is with a literal or runs in several databases

Before running plan guide script generated, the user can:

- Specify the name of the execution plan
- Change the name of the database (e.g. the option may be useful when identical queries are run in several databases)
- Choose tips, i.e. hints for the script with plan guide. The option allows to specify:
 - Custom tips for indexes, join operators, order of operations, etc.
 - Directions according to the plan on which the form was launched
 - Tips with plan parameterization (if the query is a literal)

PLAN GUIDE GENERATOR

Online statement text: `SELECT * FROM "Navision NN"."dbo"."NNNNNN NN$AWHM Document Header" WHERE (("Source Document ID"=@P1)) AND (("Document Type"<=>@P2) AND ("Document Type"=@P3)) AND (("Location Code"=@P4)) AND "Warehouse Document Type"=@P5 AND "Warehouse Document No_"<=>@P6 ORDER BY "Warehouse Document Type" DESC,"Warehouse Document No_" DESC,"Document ID"`

Statement Type: SQL (run from application)

Number of query handles in sys.dm_exec_query_stat view: 1 if there is more than 1 handle, it could be LITERAL query or query is ran in several db's

Database: ERP

Parameters definition: @P1 varchar(20),@P2 int,@P3 int,@P4 varchar(10),@P5 int,@P6 varchar(20)

Object name:

Plan guide name: DBPLUS_0x52E706C0113715EA

Plan guide HINTS: Use hints from selected plan hash

Generate plan guide Cancel

Option **Plan guide HINTS: Use hints from selected plan hash**, means that the plan-based script will use the selected execution plan.

Option **Plan guide HINTS: Use own hints**, we have the opportunity to provide our own suggestions for the plan - below is an example:

Plan guide HINTS

Use own hints

Plan guide hints

TABLE HINT("Navision NN"."dbo"."NNNNNN NNSAWHM Document Header", INDEX(I_225DATEIDX)),
FORCE ORDER, LOOP JOIN

Option **Use Parametrization Forced hint**, we use in a situation where we deal with literals. For example, for the selected Query Hash ID, we have many SQL handles in the system view with query statistics.

An example of the form of queries, where it is worth to use parameterizations, are:

```
select * from employees where nr=1
```

```
select * from employees where nr=2
```

```
select * from employees where nr=3
```

```
select * from employees where nr=4
```

Below is a screenshot with an example script for the implementation of the execution plan:

```

/* Please execute following plan guide script on instance [SQL12-12\NAV_NN] with sysadmin/dbowner rights*/

use master;
declare @query_hash varbinary(8);
declare @query_text nvarchar(max);
declare @query_text_template nvarchar(max);
declare @plan_name nvarchar(max);
declare @plan_name_old nvarchar(max);

declare @plan_type nvarchar(20);
declare @plan_module nvarchar(128);
declare @plan_params nvarchar(max);
declare @plan_hints nvarchar(max);

set @query_hash = 0x52E706C0113715EA;
set @plan_name = N'DBPLUS_0x52E706C0113715EA';
set @plan_type = N'SQL';
set @plan_module = null;
set @plan_params = N'@P1 varchar(20),@P2 int,@P3 int,@P4 varchar(10),@P5 int,@P6 varchar(20)';
set @plan_name_old = null;
set @plan_hints = N'OPTION(TABLE HINT("Navision NN"."dbo"."NNNNNN NNSAWHM Document Header", INDEX(I_225DATEI...))

select top 1 @query_text =
    (SUBSTRING(qt.text, qs.statement_start_offset / 2 + 1,
        (CASE WHEN qs.statement_end_offset = -1
            THEN LEN(CONVERT(nvarchar(max), qt.text)) * 2
            ELSE qs.statement_end_offset end -
            qs.statement_start_offset
        ) / 2 + 1
    ))
from sys.dm_exec_query_stats qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
where qs.query_hash = @query_hash;

if (@query_text collate database_default is null)
begin

```

In the next step, the script should be copied to Microsoft SQL Studio, and executed on a user with administrative privileges.

Important notice:

The option with generation script for the implementation of the execution plan is available since SQL2008 and does not work in the EXPRESS, AZURE editions.

In some cases, you may not be able to run a script to run the execution plan. This applies to i.a. the following situations:

The SQL database optimizer does NOT support all types of SQL commands (e.g. for the command: `update employees set name = @ X where nr_pesel = @ y`, set plan is impossible, but for the command: `update p set p.name=@X from employees p where p.nr_pesel=@y`, it is possible to set a plan)

The SQL database optimizer does NOT support the case when the query is run in a T-SQL block - the plan can be set when the query is run from an application or a database object.

You need to be very carefully when applying execution plans, i.a. choose the right plan or the right tips for the plan.

After submitting the execution plan, you can check its existence in the sys.plan_guides system view or in **Plan Guides** tab.

Information about the plan guide will also be visible in the query plan view.

EXECUTION PLAN FOR LAST SELECTED PLAN HASH: 0XB2E92E93BE20C57F

Show plan objects for 0xB2E92E93BE20C57F

Database: Navision LV

Missing indexes

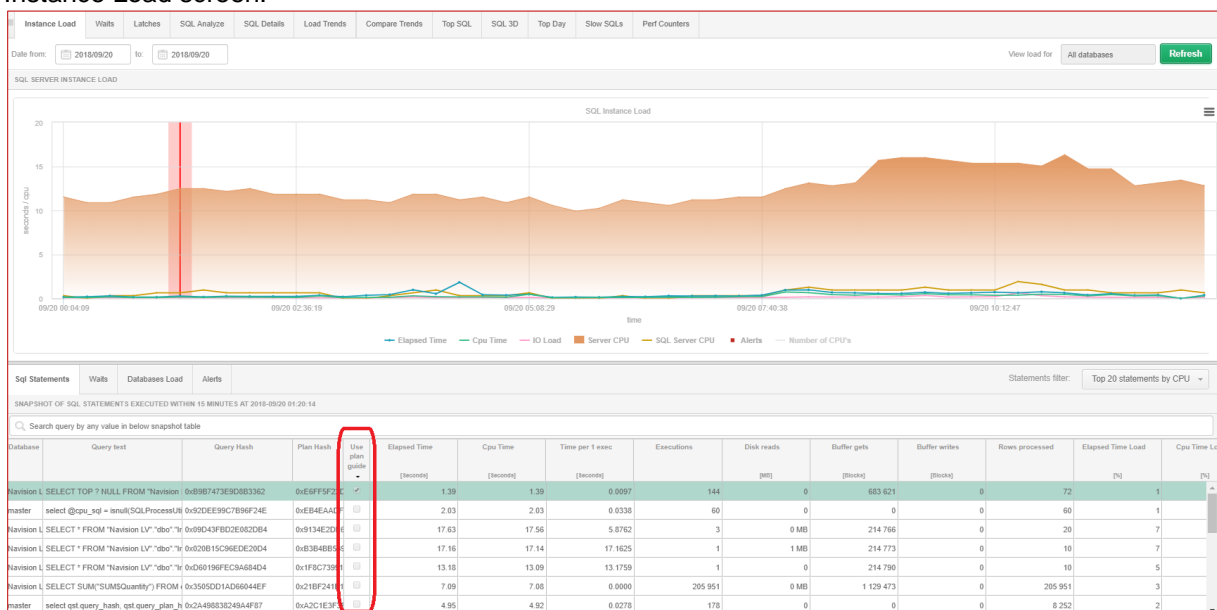
/*Missing index impact: 73.9546*/ use [Navision LV]: create index missing_index_Inter Cars LV\$Reservation Entry Qx_09202018] on

SELECT (Cost - 0.0148431, Rows - 0, CPU - 0, IO - 0) - PlanGuide: DBPLUS_0xFA1B50229DD66C33

Compute Scalar (Cost - 0.0148431, Rows - 1, CPU - 0.0000001, IO - 0)

Top (Cost - 0.014843, Rows - 1, CPU - 0.0000001, IO - 0)

Instance Load screen:



Slow SQL's screen:

Instance Load | Waits | Latches | SQL Analyze | SQL Details | Load Trends | Compare Trends | Top SQL | SQL 3D | Top Day | Slow SQLs | Perf Counters

Date from: 2018/09/20 to: 2018/09/20 Min elapsed execution time: 1 seconds Refresh

SQL STATEMENTS EXECUTED DURING SPECIFIED PERIOD TIME

Search statistic by query text or hash value

Query text	Query Hash	Plan Hash	Use plan guide	Elapsed Time [seconds]	Cpu Time [seconds]	Time per 1 exec. [seconds]	Executions	Disk reads [MB]	Buffer gets [blocks]	Buffer writers [blocks]	Rows processed
SELECT * FROM "Navision LV" "dbo" "Inter Cars LV\$Reservation Entry Qx_09202018"	0xA803D42268	0x6F68682C9183	✗	4.40	0.01	0.0254	173	0	407	0	1
SELECT TOP 7 NULL FROM "Navision LV" "dbo" "Inter Cars LV\$Reservation Entry Qx_09202018"	0x9B87473E9D863362	0x6FF9F220C57F	✓	15.21	15.05	0.0088	1 729	12 MB	7 800 880	0	9
SELECT TOP 7 NULL FROM "Navision LV" "dbo" "Inter Cars LV\$Reservation Entry Qx_09202018"	0x9B87473E9D863362	0x6FF9F220C57F	✓	12.51	12.50	0.0082	1 517	6 MB	6 778 416	0	8
SELECT TOP 7 * FROM "Navision LV" "dbo" "Inter Cars LV\$Reservation Entry Qx_09202018"	0x3106F18C2291	0x42F602F4C171	✗	1.02	1.02	0.0027	373	0 MB	31 578	0	3 1
SELECT * FROM "Navision LV" "dbo" "Inter Cars LV\$Reservation Entry Qx_09202018"	0x788454FDC2D	0x6B753F4C1711	✗	1.02	1.01	0.3416	3	1 MB	220 646	0	1
SELECT * FROM "Navision LV" "dbo" "Inter Cars LV\$Reservation Entry Qx_09202018"	0x74A58ABA9B	0xA1DEEB16A1	✗	1.03	1.02	0.0034	302	0 MB	145 934	0	1

The next function in the execution plan pop-up menu is [Show statement scripts with filled parameters](#). This option is useful for testing and checking queries (statistics or execution plan) directly in the monitored SQL instance.

After clicking Show statement scripts with filled parameters, a new T-SQL block appears in the new window, containing:

- Reference to the database
- Parameter declarations and setting their value
- Query text

```
USE [adv_works];

--parameters declaration
DECLARE @v int;
DECLARE @p int;

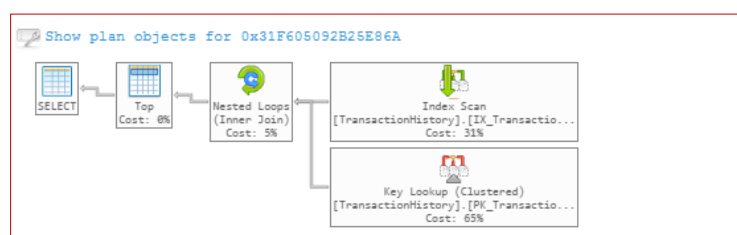
--parameters sample values defined in execution plan
SET @v = 1000;
SET @p = 1;

--statement
select top(@v) t.* from Production.TransactionHistory t where t.ProductID = @p ORDER BY t.TransactionDate option(optimize for (@v=1000))
```

Notice:

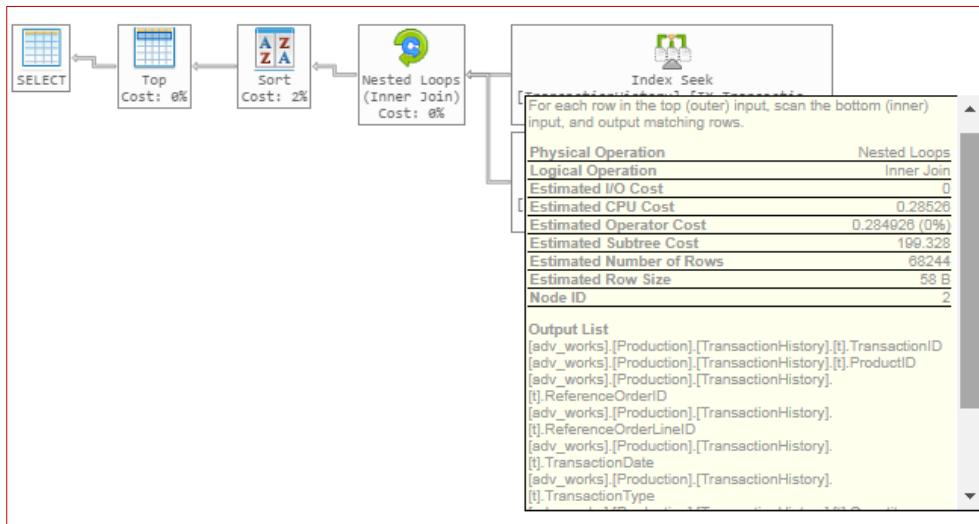
- Information about parameters and values is retrieved from the execution plan - the values that the database engine used when compiling the execution plan.
- In the Sessions, Sql Details screen (with the Online Values filter enabled) real parameter values will be displayed - values used during the last build of the execution plan
- For queries run from the T-SQL procedures / functions / block, problems may arise in determining the type of parameter - this results from the limitation of SQL and truncation of the query text in the plan tree (nodes with the StatementText attribute)

The last function from the pop-up menu allows you to change the format of the execution plan - below the example with the plan of execution in the graphic format:



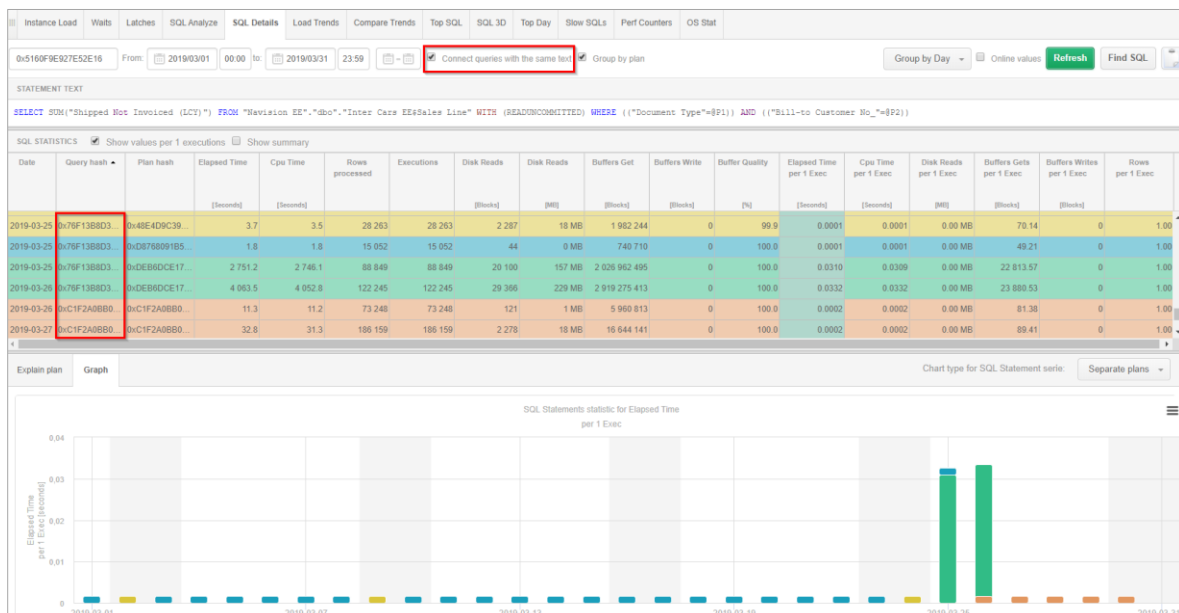
After hovering the mouse over individual nodes of the execution plan, more detailed operations are shown in the tooltip:

- description of the operation
- list of predicates for search for SEEK operations
- list of columns being returned



IMPORTANT: after changing the format in any other system window, the same format of the execution plan will be used.

In the latest version of the application in the SQL Details tab we have added the ability to combine queries with the same query content. This functionality is useful when the Plan Guide for a query is implemented. In this case, the query receives a new identifier and it is not easy to verify and assess whether the implementation of PlanGuide brought the expected result. In the new version, by selecting the Connect queries with the same text option in the tab, the user searches for queries that have the same query text. Thanks to this, the chart can combine several queries at the same time and assess whether the change related to the implementation of Plan Guide has brought the expected stability and improved query performance.



6.2.1.5.2 Show Plan Objects functionality

The **Show Plan Objects** functionality appears in screens where the query text and execution plan are available. After clicking on the link with the same name, a window will appear as below:

- **Objects Used in Explain Plan** - a list of all objects used by the query in given execution plan
- **Indexes for selected object** - list of indexes for selected table - row selected in the "Objects Used in Explain Plan"
- The area comprised of three tabs:
 - a. **Object Columns** - a list of individual columns of the selected object, along with information such as: column name, data type, id columns, density (the lower density - the higher selectivity of the column)
 - b. **Info** - basic information about selected object
 - c. **Properties** - additional properties of selected object

- Limiting the choice of data, or of the data with the where clause and tables joins
- Whether the request is with parameters or literals
- Actions that SQL Engine chose to data download
- Whether appropriate indexes are in the table
- Way of reading the data - Nested Loops vs. Hash Join

In the current version, the Show Plan Objects view showed the content of the query in unformatted form.

SQL TEXT (QUERY HASH: 68A52B7AD30E45A6)

Parse SQL Query

EXPLAIN PLAN (PLAN NAME: 6X2DBAICF07BC9C4F)

Close Plan Objects

```

select distinct top 7 "role0",RoleId as "roleId", "role0".Name as "Name", convert(bigint, "role0".VersionNumber) as
"versionNumber" from Role as "role0" join SystemUserRoles as "systemUserRoles" on ("role0".RoleId = "systemUserRoles".RoleId
and ("systemUserRoles".SystemUserId = @SystemUserId)) where ("role0".Name = @Name) order by "role0".RoleId asc

```

```

--Database: InterCar_MSCRM
--SELECT (Cost = 0.000000, Rows = 0, CPU = 0, IO = 0)
--Top (Cost = 0.000000, Rows = 0.000000, CPU = 0.0000000000, IO = 0)
--  --CostEstimate (Cost = 0.000000, Rows = 0.000000, CPU = 0.0000000000, IO = 0.011810)
--    --CostEstimate (Cost = 0.000000, Rows = 0.000000, CPU = 0.0000000000, IO = 0.011810)
--      --Compute Scalar (Cost = 0.011809, Rows = 0.000000, CPU = 0.0000000000, IO = 0)
--        --Inner Join--Nested Loop (Cost = 0.011809, Rows = 0.000000, CPU = 0.0000000000, IO = 0)
--          --Index Seek ( (SystemUserRoles). [UQ_SystemUserRoles] [systemUserRoles]) (Cost = 0.000000, Rows = 0.000000, CPU
--            --Clustered Index Seek ( (RoleBase). [Index_PrimaryKey_Role] [1]) (Cost = 0.000000, Rows = 0, CPU = 0.000000, IO
--
--Plan Completion Time: 12 ms

```

OBJECTS USED IN EXPLAIN PLAN

INDEXES FOR SELECTED OBJECT ([DBO].[SYSTEMUSERROLES])

Type	Owner	Object Name	Table Name	Database	Index name	Enabled	Index columns	Included columns	Seeks	Scans	Lookups	Updates
index	[dbo]	[UQ_SystemUserRoles]	[SystemUserRoles]	[InterCar_MSCRM]	cnh_PrimaryKey_SystemU	<input checked="" type="checkbox"/>	SystemUser/Role		985	8 805	1 123	213
index	[dbo]	[cnch_PrimaryKey_Role]	[RoleBase]	[InterCar_MSCRM]	fnch_Sync_VersionNumber	<input checked="" type="checkbox"/>	VersionNumber		0	0	0	213
table	[dbo]	[SystemUserRoles]	[SystemUserRoles]	[InterCar_MSCRM]	ndr_for_cascaderelationsh	<input checked="" type="checkbox"/>	RoleId		247			0
table	[dbo]	[RoleBase]	[RoleBase]	[InterCar_MSCRM]	UQ_SystemUserRoles	<input checked="" type="checkbox"/>	SystemUser/RoleId		10 161 765	1 189	0	213

- manual,
- automatic.

In the manual mode, after enter Show Plan Objects and press the [Parse SQL Query] button, the query is formatted and parsed. Formatting the query changes the presentation in the SQL TEXT window to a form that facilitates query analysis.

The parsing function in the current version gives the opportunity to highlight columns that belong to the given object participating in the query. In the following case, the [SystemUserRoles], table was selected on the query, and all columns associated with the table.

Depending on the object that the User indicates, the objects are marked in different colors:

- Table (green),
- Indexes (yellow).

The highlighting is performed in both the SQL TEXT and EXPLAIN_PLAN fields.

OBJECTS USED IN EXPLAIN PLAN

Type	Owner	Object Name	Table Name	Database
index	[dbo]	[UQ_SystemUserRoles]	[SystemUserRoles]	[InterCars_MSCRM]
index	[dbo]	[cndr_PrimaryKey_Role]	[RoleBase]	[InterCars_MSCRM]
table	[dbo]	[SystemUserRoles]	[SystemUserRoles]	[InterCars_MSCRM]
table	[dbo]	[RoleBase]	[RoleBase]	[InterCars_MSCRM]

INDEXES FOR SELECTED OBJECT [SYS].[SYSTEMUSERROLES]

Index name	Enabled	Index columns	Included columns	Seeks	Scans	Locks	Updates
cndr_PrimaryKey_SystemU...	✓	SystemUserRoleId		985	8 805	1 123	213
idx_Sync_VersionNumber	✓	VersionNumber		0	0	0	213
idx_for_cascadeRelation...	✓	RoleId		247	0	0	213
UQ_SystemUserRoles	✓	SystemUserRole, RoleId		10 144 238	1 189	0	213

As part of the mechanism, it is possible to select the range of highlighted objects on the query. To change the configuration, click the [cog] button on the Show plan Objects page.

Parse SQL Query **EXPLAIN PLAN**

(P_DOK_MA.ZNA,1,0,P_DOK_MA.FILE)) PRM,
AR)) WPRM FROM DOK_MA, P_DOK_MA WHERE
TWEEN :B3 AND :B2 AND
OD

↳ INSERT STATEMENT
↳ LOAD TABLE CONVENTION
↳ SORT (GROUP BY)
↳ NESTED LOOP
↳ NESTED
↳ TA

As a result of the click, a window will open where the User can choose:

- SQL Parser – On demand/Automatic – parsing mode.

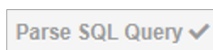
Depending on the mode, when the User enter the Show Plan Objects screen, the query will be automatically formatted and parsed (Automatic mode).

- Highlight columns - depending on the selection, the columns in the query will be highlighted
- Highlight color – color selection for table / index highlight.

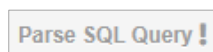
Depending on the selected option (checkbox selected), the User can highlight columns located in any part of the analyzed query.

Each time after parsing the query, the User will receive information about the status of the performed operation. Under the first version of the parser mechanism, not all types of queries were handled. Support for subsequent queries will be added in each subsequent release.

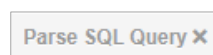
If everything went well, the button on the right will be presented in this form:



In case the query was formatted correctly, however, there was a problem with reading all objects from the query:



When the "X" character is presented after parsing, it means that the query could not be properly formatted as well as the parsing done. Support for such queries will be provided in subsequent version updates.



6.2.1.5.3 Find queries in SQL Details

In a situation where no Query ID is known, and a specific query needs to be found e.g.:

- Queries that refer to a specified table
- Queries that changed execution plan
- New Queries
- Queries that use specific SQL instance objects e.g. an index

- Queries using objects stabilizing execution plans

Clicking on **[Find SQL]** button – displays the search query window.

Searching for queries containing specific text - Statement by text

Queries found can be "moved" to the clipboard of the SQL Details screen by clicking the **[Plus]** button next to the query identifier. For each query information is presented on:

- Query ID
- Last execution date
- Elapsed Time
- CPU Time
- Number of executions
- Number of reading blocks:
 - From disk devices
 - From memory
- Number of contained records
- Text of the query

Entering several expressions in the search field, the result will be returned in two separate grids:

- FIND RESULT FOR **EXACT** QUERY TEXT MATCHING WITH
- FIND RESULT FOR **SIMILAR** QUERY TEXT MATCHING WITH

For example, the result presented for searching for queries after entering "select max". In the upper table queries that exactly agree with the searched content "SELECT MAX (SNAP_ID) ..." are returned. In the bottom table, queries for "select% max" were found
 "SELECT NVL(MAX(P.LP), 0) + 1 FROM P_R_SAM P...".

Statement by text

Plan Flip-Flop Statements

New statements

Statements using objects

Date from:
Date to:
Max. returned statements:

Search

FIND RESULTS FOR EXACT QUERY TEXT MATCHING WITH SELECT TOP ?

Query Hash	Last execution date	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Buffer writes [Blocks]	Rows processed	Query text
0x346F207C0F4329A2	2018-09-18	10.37	10.12	10	397 MB	9 849 872	15	493	INSERT INTO #wrk_tab_01 (rekord_id, n
0x926088A61AC81837	2018-09-18	478.52	439.11	1	197 MB	6 094 450	7 478	86 998	INSERT INTO AZURECRM365.crm_365
0x6863A0BBC4A4DF16	2018-09-18	221.91	79.24	11 606	1 MB	2 175 894	73 924	11 600	SELECT (select top ? isnull(qp.value,-?)
0x3D750EF32C53071B	2018-09-18	280.68	259.37	5 340	0	1 216	0	5 264	select @cpu_sql = isnull(SQLProcessUtil

FIND RESULTS FOR SIMILAR QUERY TEXT MATCHING WITH SELECT%TOP%?

Query Hash	Last execution date	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Buffer writes [Blocks]	Rows processed	Query text
0xF81B4FB563975405	2018-09-18	218.28	54.18	10	1 812 MB	707 413	0	3 449 656	SELECT acc.AccountNumber kh_kod, ei
0xDBEB8D1C55BB6A21	2018-09-18	3.67	3.52	103	0 MB	2 709 717	402	103	select COUNT(*) as [#TotalRecordCount
0xA7E5508B9F702C74	2018-09-18	1 720.08	1 539.54	928	2 180 MB	857 078 250	22	124 878	SELECT DISTINCT k.data, k.rok, k.mies
0x3C34FB8ECE63EE87	2018-09-18	3.16	3.16	1	0	18 312	0	0	select DISTINCT top ? "account0".Accot

Searching the queries that change the plan

With the Plan Flip-Flop Statements tab selected, a search for queries that have changed the plan of execution in a given period of time. Using the values of i.a. Elapsed Time, CPU Time, an ability to search for those queries whose share in the load is significant.

For queries changing the execution plan, additional information is grouped according to the following areas:

- Statistics with a summary for all performance plans on which the query worked,
- Slowest plan statistics summary,
- Fastest plan statistics summary,
- Comparison between Slowest and Fastest
- Possible time reductions for queries statistic.

Below is an example of the search results for those questions that will change the execution plan within two weeks:

View of the areas [Total statistics, Slowest plan statistics]

Statement by text

Plan Flip-Flop Stateme...

New statements

Search

CLICK ON [ADD TO SQL DETAILS] BUTTON (ICON WITH +) TO ADD QUERY IDENTIFIER TO QUERY HASHES TOOLBAR LIST

Query Hash	Query text	Total statistics				Slowest plan statistics				
		Elapsed Time	Cpu Time	Executions	Number of plans	Plan Hash	Elapsed Time	Cpu Time	Executions	Elapsed Time Per 1 exec
		[Seconds]	[Seconds]				[Seconds]	[Seconds]		[Seconds]
0x64C102F23329DC98	select top(@v)	486 925.70	394 926.91	426 821	2	0x31F605092B25	442 606.41	367 606.00	156 096	2.8355
0xA86C6E5BE207D6E8	select max(Errc	70.20	24.68	43	2	0x397376A5E330	52.39	19.52	21	2.4946
0x25B65C61193863C4	select * from Pr	11 726.99	10 221.35	1 176 774	3	0xD445611DDBA	420.77	138.82	1 773	0.2373
0xE95D16F7F24BD1F3	SELECT DB_ID	68.70	60.64	6 695	2	0x2370E781E95E	25.13	22.26	1 339	0.0188
0x24BFF45573B477FD	select convert(i	98.46	88.74	1 343	2	0x89C31130AB10	26.72	24.42	343	0.0779
0x89EB3EE49C2797CF	select ? as rec	16.09	15.89	20 742	2	0x43B435618BC8	7.77	7.68	6 612	0.0012

View of the areas [Fastest plan statistics, Slowest vs. Fastest, Estimation statistics]

Plans	Elapsed Time Per 1 exec [Seconds]	Plan Hash	Fastest plan statistics				Slowest vs Fastest		Estimation statistics	
			Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Elapsed Time Per 1 exec [Seconds]	Times faster	Elapsed Time Per 1 exec difference [Seconds]	Elapsed Time to reduce [Seconds]	Cpu Time to reduce [Seconds]
5 096	2.8355	0xF02EB8B03876	44 319.29	27 320.91	270 725	0.1637	17	2.6718	417 052.5628	351 853.1681
21	2.4946	0xFE2C0C637B8	17.82	5.16	22	0.8098	3	1.6848	35.3808	14.5963
1 773	0.2373	0x90B998ECB7C	8 388.18	7 310.25	1 169 949	0.0072	33	0.2301	3 289.8811	2 868.4574
1 339	0.0188	0x43E66D931657	43.57	38.38	5 356	0.0081	2	0.0106	14.2387	12.6670
343	0.0779	0x2B459523C16C	71.73	64.33	1 000	0.0717	1	0.0062	2.1186	2.3535
6 612	0.0012	0xC99C4CF8765	8.32	8.21	14 130	0.0006	2	0.0006	3.8777	3.8421

An important area of the **Flip-Flop Statements** plan screen is the statistics estimation. The columns **Elapsed Time to reduce** and **CPU Time to reduce**, is a calculation about the possible reduction of time for the case when the query would work to be disabled on the fastest execution plan.

Helpful tip:

Sorting one of these columns will allow to find those questions whose optimization will bring the greatest improvement in performance.

Searching new queries – New Statements

It is also possible to search for new queries that started to run in a given time period, for which the total execution time is greater than the specified value.

In this case, queries that were performed on 30.08.2018 will be searched and not performed on 29.08.2018 for which the total duration for all queries was greater than 100 seconds.

Statement by text
Plan Flip-Flop Statements
New statements
Statements using objects

Statement executed in period
Date from: 2018/10/04 00:00 Date to: 2018/10/04 23:59 Min. elapsed time (sec): 100
And statement not executed in the period range
Date from: 2018/10/03 00:00 Date to: 2018/10/03 23:59

Search

FIND RESULTS

Query Hash	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Buffer writes [Blocks]	Rows processed	Query text
0xB58F2CFC4029B1D0	932.37	911.63	3	3 156 MB	359 922 726	0	28	WITH acc as (select top (@top_num) accountnumber from (
0x93479BAC0B2E6D3B	809.76	521.87	1	3 411 MB	278 781 666	1 618 888	20 078 709	delete CRM.[SPRZ_FAKTY_DETAL] where rok_miesiac in (s
0xD95BB4DF4291EFB9	670.59	767.46	1	1 383 MB	324 171 686	1 622 421	20 147 052	INSERT INTO [CRM].[SPRZ_FAKTY_DETAL] ([kh_kod], [pla
0x5A322C3AB33CB575	472.51	453.28	20	18 MB	206 298 730	623	91 994	WITH "ic_zs_platnik0Security" as (select ic_kontoid as "ic_ko
0x445B10BB181FA5A	354.36	308.57	29	37 MB	145 239 830	0	839	WITH "account0Security" as (select AccountId as "Accountid
0x3D20233042CDB935	249.40	213.30	1	19 MB	7 286 471	31 143	603 662	INSERT INTO IT.CRM.SPRZEDAZ_FAKTY_LOK_G1_PROD
0x10A1F204C9DA534E	236.47	227.73	14	104 MB	150 098 444	0	70 014	WITH "ic_zs_platnik0Security" as (select ic_kontoid as "ic_ko
0x2CE7411E70E63C79	130.96	111.13	5	188 MB	98 856 874	495	23 570	INSERT INTO #konta (accountid ,platnikid ,accountnumber ,
0xCA399F8269438668	123.88	121.09	1	0 MB	15 154 140	0	1	SELECT s.rok, s.miesiac, p.prod_naz, s.prod_id, s.wal_kod, s

Statements using objects

It is also possible to search for a query after entering the name of the object. In the case below, queries using DBPLUS_SNAPS objects were searched in a given period of time.

Statement by text
Plan Flip-Flop Statements
New statements
Statements using objects

DBPLUS_SNAPS
Date from: 2018/09/18 00:00 Date to: 2018/09/18 23:59 Max. returned statements: 100

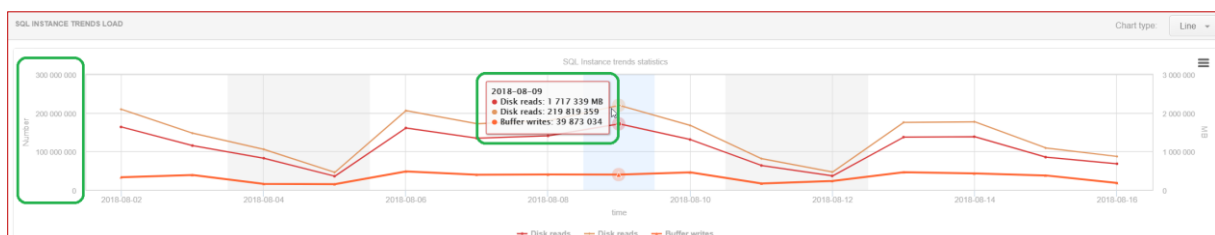
Search

FIND RESULTS

Hash Value	Elapsed Time [Seconds]	Cpu Time [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Rows processed	Query text
3710599702	51.52	21.57	132	11 MB	3 710 599 702	132	SELECT MAX (SNAP_ID) FROM DBPLUS_SNAPS WHERE SNAP_ID < :b1 AND NUM
965693090	14.53	5.72	1 650	7 MB	965 693 090	1 650	SELECT MIN (SNAP_ID) FROM DBPLUS_SNAPS WHERE LOGDATE >= :b1
276493203	6.72	2.51	66	26 MB	276 493 203	66	SELECT NVL(MAX (LOGDATE) , '2000-01-01 00:00:00') , MAX (SNAP_ID) FROM DBF
3715327	5.41	2.20	16	7 MB	3 715 327	16	SELECT MAX (SNAP_ID) FROM DBPLUS_SNAPS WHERE LOGDATE <= SYSDATE -

6.2.1.6 Load Trends Tab

Load Trends tab allows for detailed information on trends in SQL Instance.



The page consists of three components:

- Filter with the date range and grouping option
- Graph presenting certain indicators over time
- The table of statistics

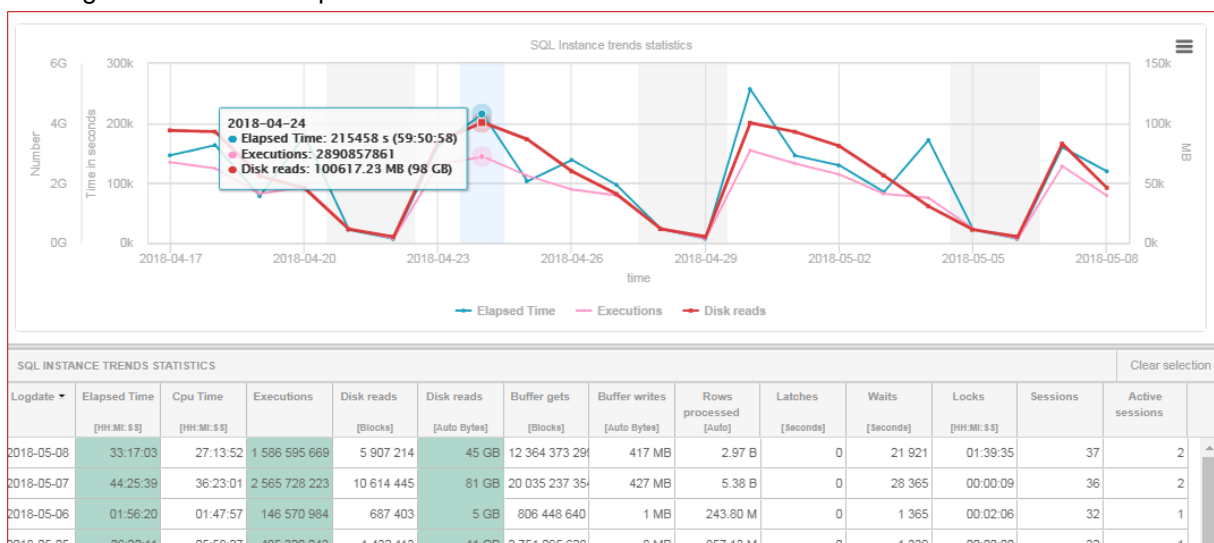
Information displayed on the graph can be shown in groups of:

- **No group by period** – with no grouping; selection of date range
- **Month** – statistics broken for months
- **Day** – statistics broken by day
- **Hour** – statistics broken by one hour
- **Snap** – statistics broken by 15 minutes

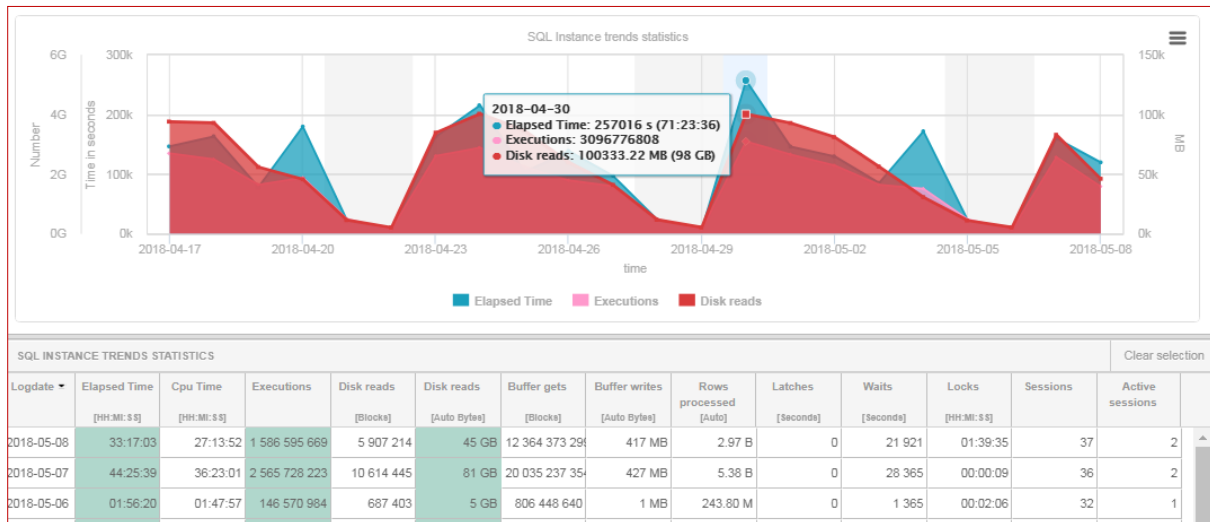
Load Trends Statistics include the following information:

- Logdate - represents the point in time for which the statistics are presented (i.e.: day, hour, minutes, for the entire period)
- Elapsed Time [Seconds] - total length of time in seconds of all queries for the selected grouping period
- CPU Time [Seconds] – total time of utilization CPU for query for the selected grouping period,
- Deadlock wait time,
- Lock timeout/cancel – time of interrupted query waiting on lock,
- Elapsed Canceled – query time interrupted by the User.
- Executions - number of performances of all searches for the selected grouping period
- Disk Reads [Blocks] – number of block readings for the selected grouping period
- Disk Reads [MB] - The number of disks reads for all queries for the selected grouping period
- Buffer Gets [Blocks] - number of utilized buffers for all queries for the selected grouping period
- Rows processed [Rows] - number of rows processed by all queries for the selected grouping period
- Latches [Seconds]- total time in seconds the duration of all latches that occurred for the selected grouping period
- Waits [Seconds]- total time in seconds spent on all waits including latches that occurred for the selected grouping period
- Locks [Seconds]- total time in seconds of the duration of all locks that have occurred for the grouping period
- Sessions – average number of logged users
- Active sessions – average number of active sessions

Clicking selected columns presents their behavior as function of time:

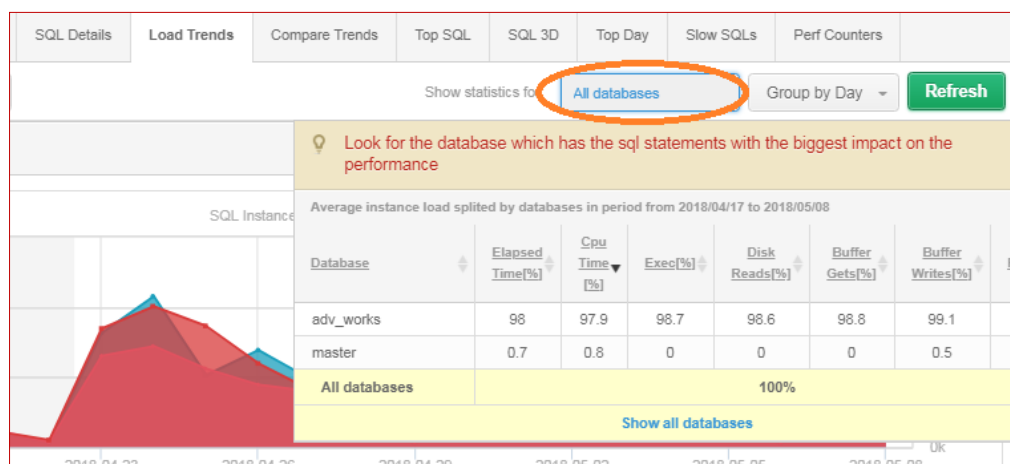


Changing the graph type to 'Area' results in Graph changes to the example below:



In the filter area there is a control that allows displaying statistics for a specific SQL Instance - by default, statistics are displayed for all databases in the SQL instance.

As a result of clicking on "All databases", statistics with % share of individual performance parameters are shown



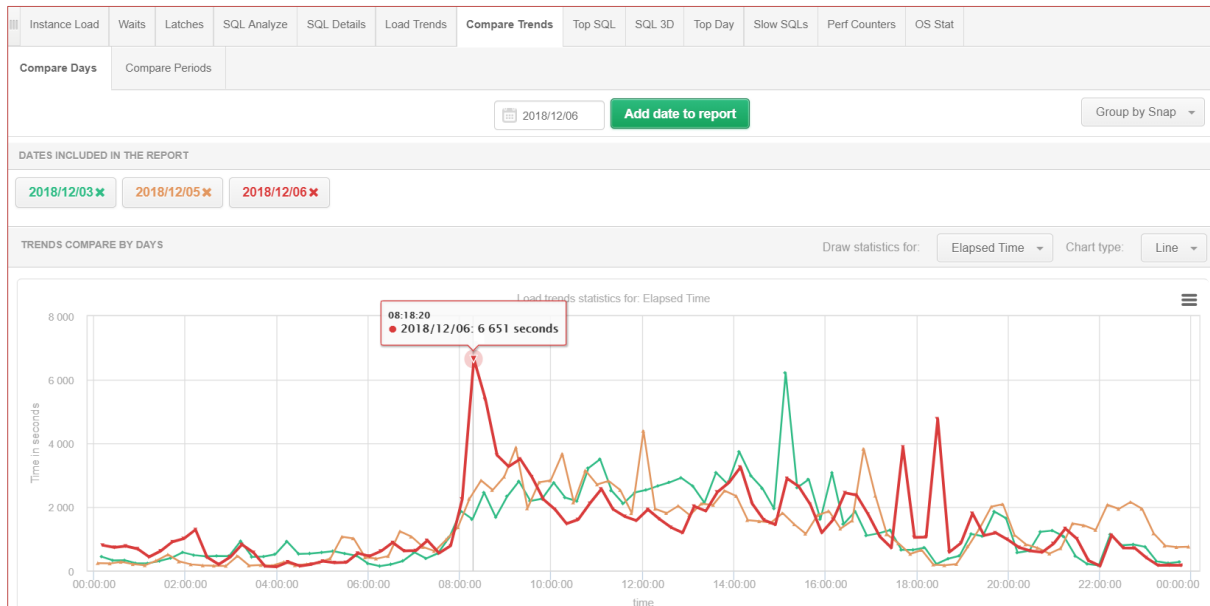
6.2.1.7 Compare Trends Tab

Compare Trends tab allows for statistics comparison. These can be compared by either days or periods.

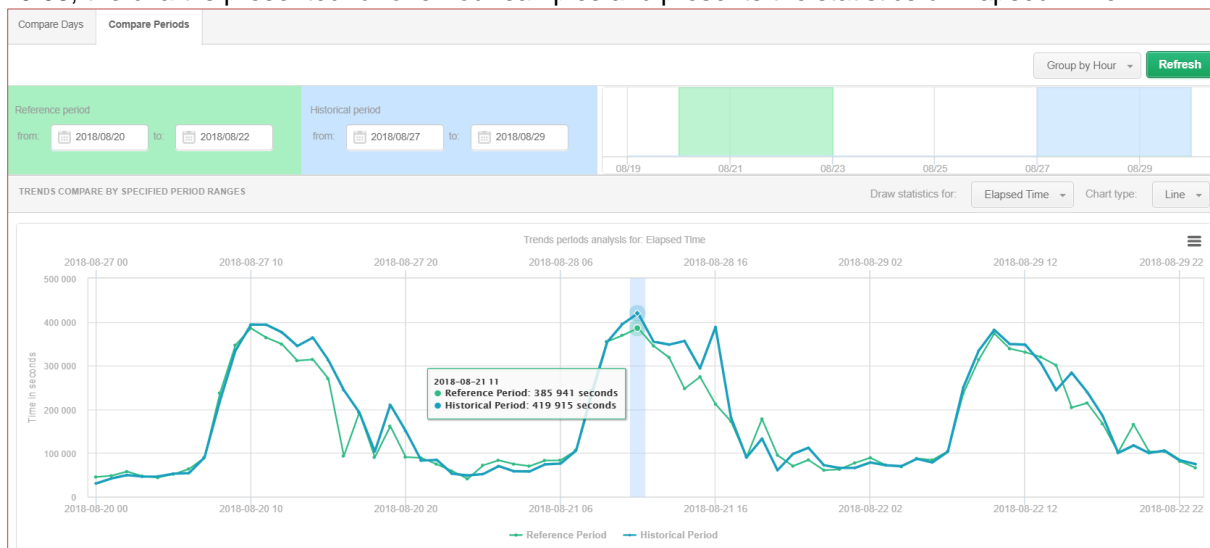
Information are presented in three areas:

- Selection of comparing method – Compare Days or Compare Periods
- Date, range of date and statistic selection
- The graph presenting specific indicators over time

Comparing databases by day [**Compare Days**], any day can be picked from the calendar and added to the report.



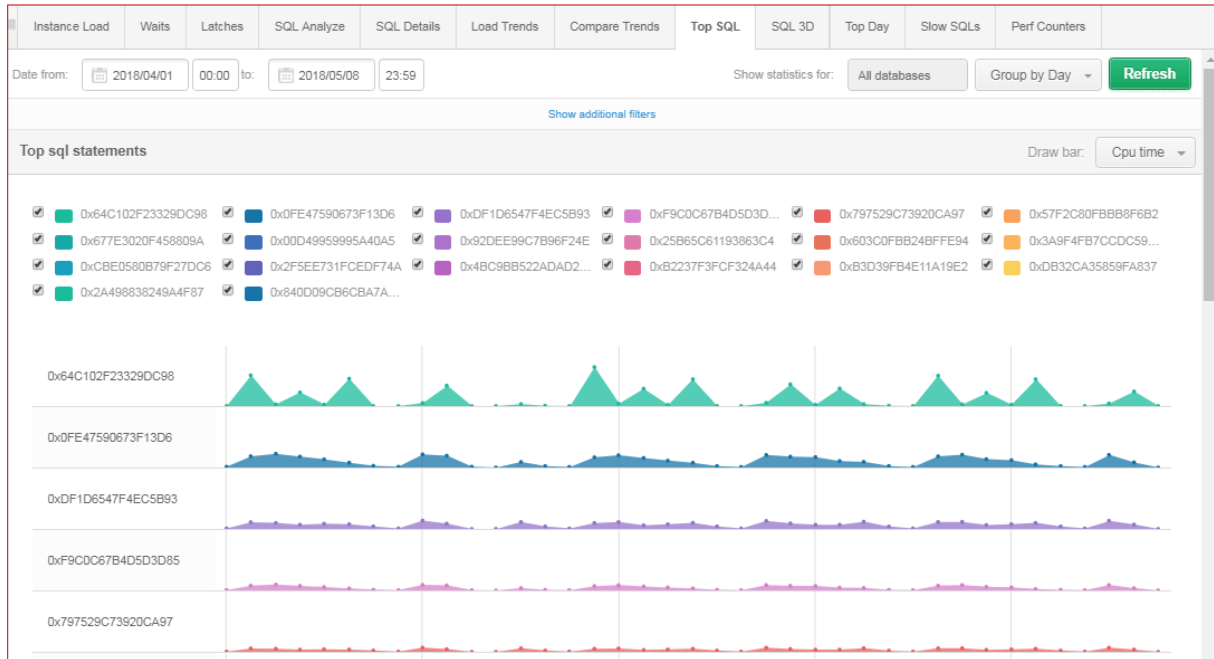
Comparing databases by period [Compare Periods] filtering options available are as for comparison after day. The chart below shows a comparison between two periods from 20.08 to 22.08 and 27.08 to 29.08, the chart is presented for one-hour samples and presents the statistics of Elapsed Time.



6.2.1.8 Top SQL Tab

The data presented on the Top SQL tab presents the most demanding queries depending on whether the user is interested in the query execution time, the number of read data, the number of processed blocks from memory (Buffer Gets), or the number of readings from disk (Disk Reads), etc.

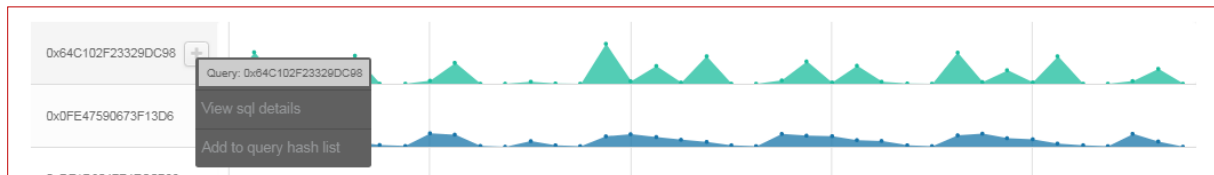
Queries are presented in the form of graphs in descending order according to the duration of the query in the selected time period for Elapsed time or other selected indicator.



From the [Top SQL] screen, any query can be easily added to the [SQL Details] by clicking the [Plus] button next to the query identifier and clicking the options:

- SQL View details - to move to the SQL Details screen and analyze specific query
- Add to query hash list - to add the query to the clipboard with a list of questions for further analysis

If the query is grouped by the query plan (selected checkbox [Group by plan]), clicking the [Plus] button, adds the query plan identifier which will be available in the [SQL Plan] tab).



Deleting individual charts from the [Top SQL] view can be done using the checkboxes in the legend.



The right side presents a choice to show top queries by selected filter:

- Elapsed time
- CPU Time
- Disk Reads
- Buffer Gets
- Buffer writes
- Rows Processed
- Executions

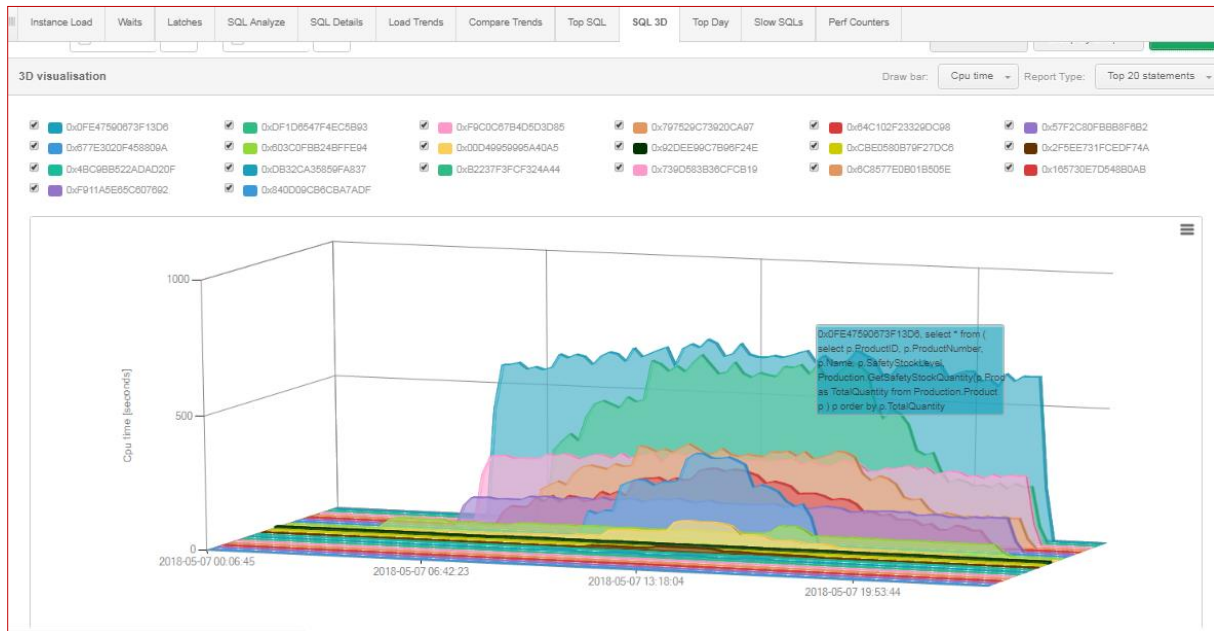
After clicking on the Show Additional filters link additional possibilities are presented including:

- Change the report type to:
 - Top 20 statements
 - Top 20 cursor statements
 - Top 20 procedures
- Change the size of the charts for the presented queries
- Adding the name of the presented statistic to the Y axis in the chart.

6.2.1.9 SQL 3D Tab

Data presented on the SQL 3D tab presents the most demanding queries: execution time, number of read data, number of blocks processed from memory (Buffer Gets), number of reads from disk (Disk Reads), etc.

The data presented on this page is analogous to those presented in Top SQL. They differ in the way of presentation. In this case, an opportunity to look at the queries in one view is presented. It is easier to indicate which query at the time has the most impact on the indicator.



On the website an option to display queries for a given date range is available. It is possible to present data in samples for day, time and snap.

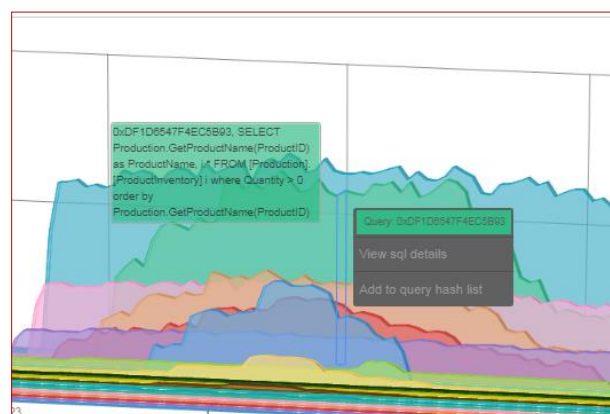
Similarly, as for the Top SQL, the graph can present data for indicators:

- Elapsed time
- CPU Time
- Disk Reads
- Buffer Gets
- Buffer writes
- Rows Processed
- Executions

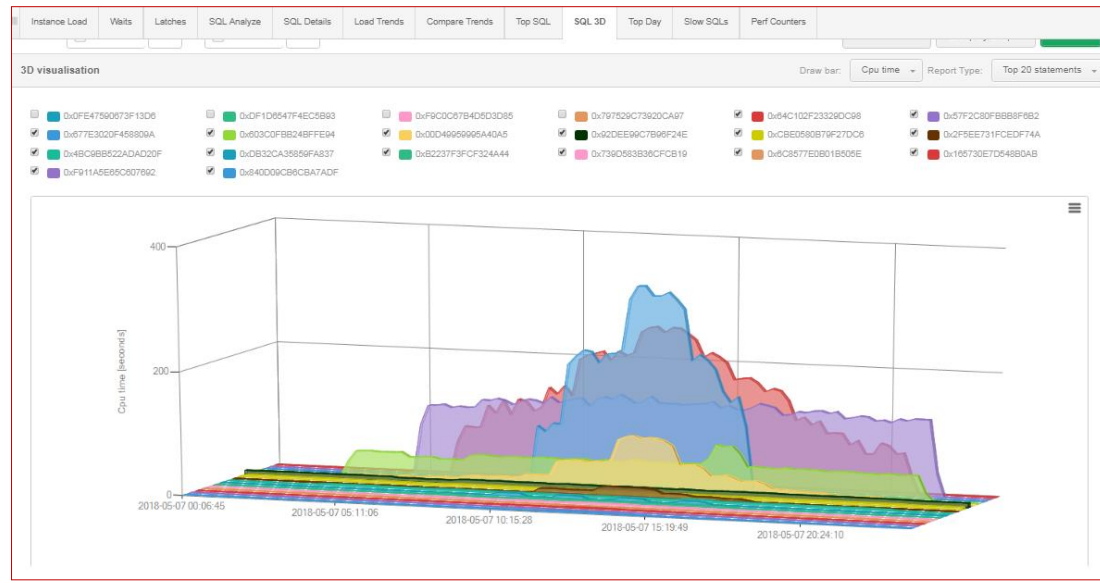
Selecting additional filters will allow for each question to be grouped by the plan as well as chart for top procedures or queries to the log.

After indicating the appropriate query on the graph, user can add them for further analysis by adding to the clipboard or going straight to the details (SQL Details tab).

Attention: In the case of selected group by plan [Group By Plan] clicking details, user is taken to the details of the given query plan (SQL Plan tab).

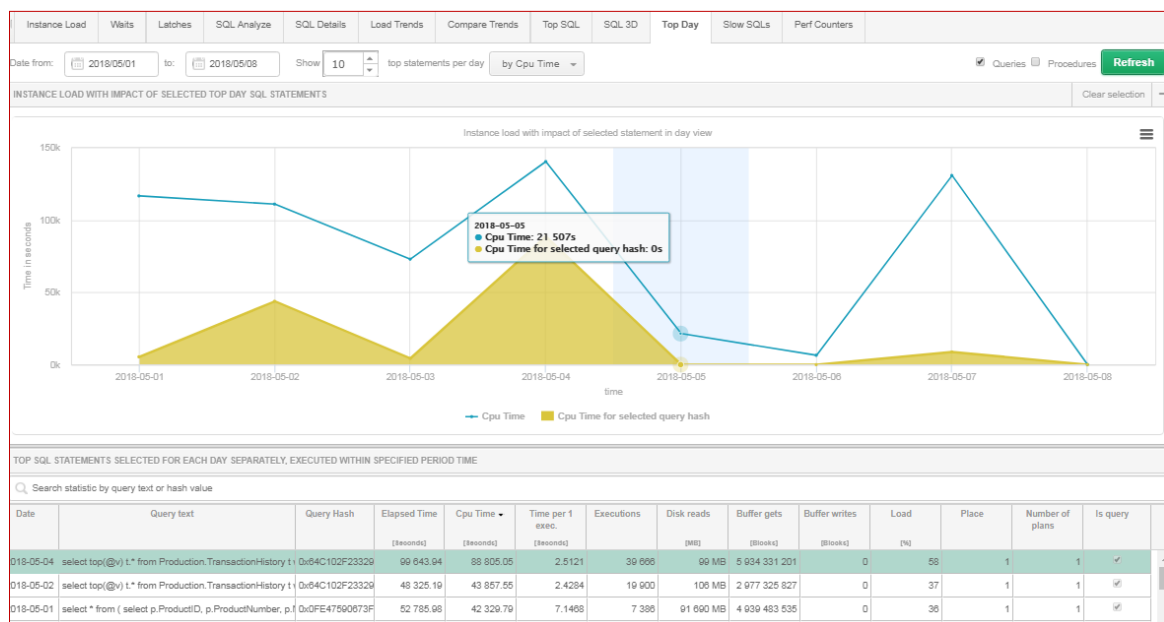


Each query can be freely unchecked by clicking on the checkbox for a given query. This will remove the query from the graph.



6.2.1.10 Top Day Tab

[Top Day] window allows to view top queries for CPU Time or Elapsed Time and track their behavior changes.



On the above slide, presented top queries in the last 2 weeks and the share of the first query impact against the CPU Load of entire SQL Instance.

Conclusion: optimizing the selected queries instance load can be reduced by 80%!

Table with top queries contains

- Date – the date the request was made,
- Query hash – the content of the query,
- Elapsed Time [Seconds] – the total execution time of the SQL query on a given day,

- CPU Time [Seconds]– total processor usage time,
- Time per 1 exec [Seconds] – the time of a single query execution
- Executions – number of executions on a given day for a given query,
- Disk reads [MB] – the amount of read data from the disk,
- Buffer gets [Blocks] – the number of utilized buffers for all queries,
- Load [%] – percentage of sql instance load,
- Place – the place where the given query affects the SQL Instance on a given day
- No. of plans – number of execution plans for a given query,
- Is Query – whether a given record is a query or a procedure.

Below the table a **Statement Text** is located– text of the selected query. By checking the query in the table, user can drag query to chart **[Instance Load]** and observe changes of its influence on the overall load of the database.

6.2.1.11 Slow SQLs Tab

On the tab system presents queries depending on the duration time. Queries which exceeded 1000 seconds (default 200 seconds) are presented (all queries for a given Hash Value).

Instance Load	Waits	Latches	SQL Analyze	SQL Details	Load Trends	Compare Trends	Top SQL	SQL 3D	Top Day	Slow SQLs	Perf Counters		
Date from: <div><div></div>2018/04/01</div> to: <div><div></div>2018/04/30</div>			Min elapsed execution time <div><div>1000</div></div> seconds										<div>Refresh</div>
SQL STATEMENTS EXECUTED DURING SPECIFIED PERIOD TIME													
<div><div></div> Search statistic by query text or hash value</div>													
Query text	Query Hash	Plan Hash	Elapsed Time [Seconds]	Cpu Time [Seconds]	Time per 1 exec. [Seconds]	Executions	Disk reads [MB]	Buffer gets [Blocks]	Buffer writes [Blocks]	Rows processed			
select top(@v) t.* from Production.TransactionHistory t	0x64C102F23329	0x31F605092	959 354.45	816 160.09	2.8172	340 532	2 619 MB	50 946 821 720	0	4 547 464 328			
select * from (select p.ProductID, p.ProductNumber, p.	0x0FE47590673F	0xE8DB3456	790 389.16	631 187.05	6.6672	118 549	1 471 876 MB	79 281 201 341	41	15 295 666 176			
SELECT Production.GetProductName(ProductID) as Pr	0xDF1D6547F4E	0xCB74A269C	524 836.72	409 994.04	0.0383	13 697 456	1 MB	44 297 572 748	0	14 587 790 640			
select @qty = isnull(sum(OrderQty),?) from Production.	0xF9C0C67B4D5	0x41E7652AF	299 064.53	265 859.45	0.0000	15 295 706 297	88 MB	47 827 435 112	0	15 295 706 297			
STATEMENT TEXT FOR QUERY HASH: 0X64C102F23329DC98													
select top(@v) t.* from Production.TransactionHistory t where t.ProductID = @p ORDER BY t.TransactionDate option(optimize for (@v=?))													

Below the table there is the content of the query and the execution plan for the selected statistics.

Notice: remember about the possibility of a detailed analysis of a specific query by clicking the [Plus] button on the query.

6.2.1.12 Perf Counters Tab

The Perf Counters window allows you to analyze Performance Counters, i.e. counters containing information about various operations performed by SQL instances.

Counters are defined according to the class below:

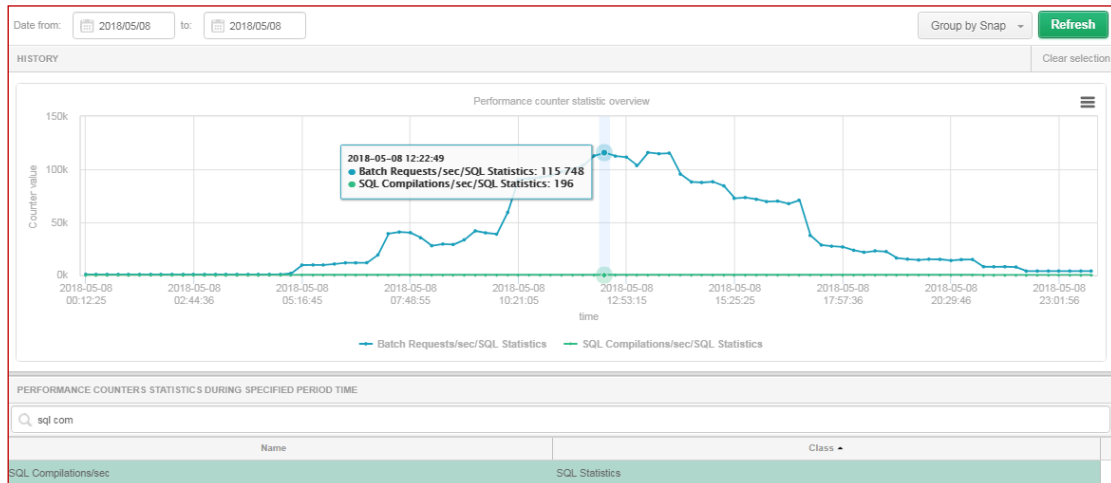
- | | |
|--------------------------|----------------------|
| • Access Methods | • Database Replica |
| • Availability Replica | • Databases |
| • Broker Activation | • File Table |
| • Broker Statistics | • General Statistics |
| • Broker TO Statistics | • Latches |
| • Broker/DBM Transport | • Locks |
| • Buffer Manager | • Memory Manager |
| • Catalog Metadata | • Plan Cache |
| • CLR | • Query Execution |
| • Cursor Manager by Type | • SQL Errors |
| • Cursor Manager Total | • SQL Statistics |

- Transactions

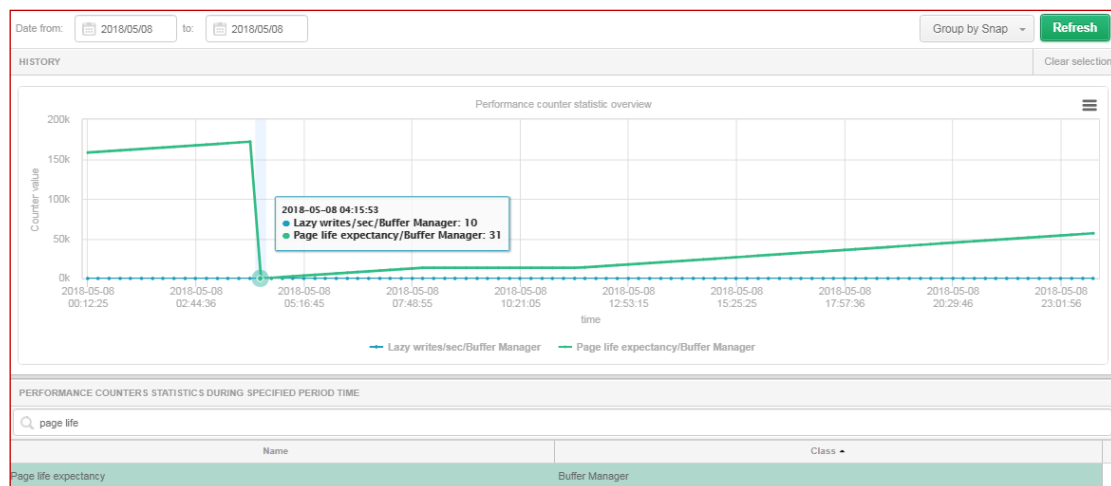
Below the chart is a list of indicators that can be searched. After clicking on the indicator, its statistics are added on the chart

Selected examples below:

Analysis of the number of Batch requests in relation to the SQL compilation of query texts / execution plans



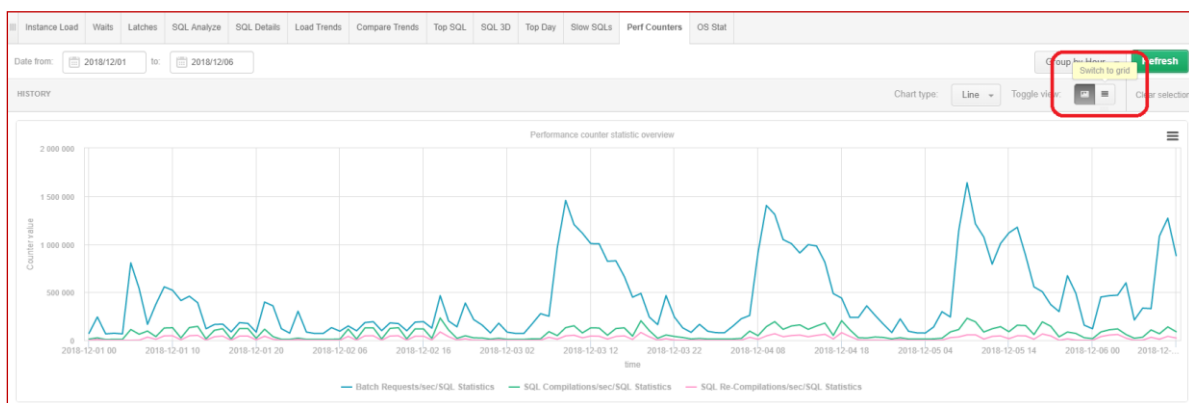
Analysis of the number of Lazy writes in relation to the Page life expectancy



Analysis of the number of Page reads in relation to the Page writes



Export is performed for statistics previously selected from the table. Export is possible by changing the chart preview to the tabular form **[Switch to grid]**.

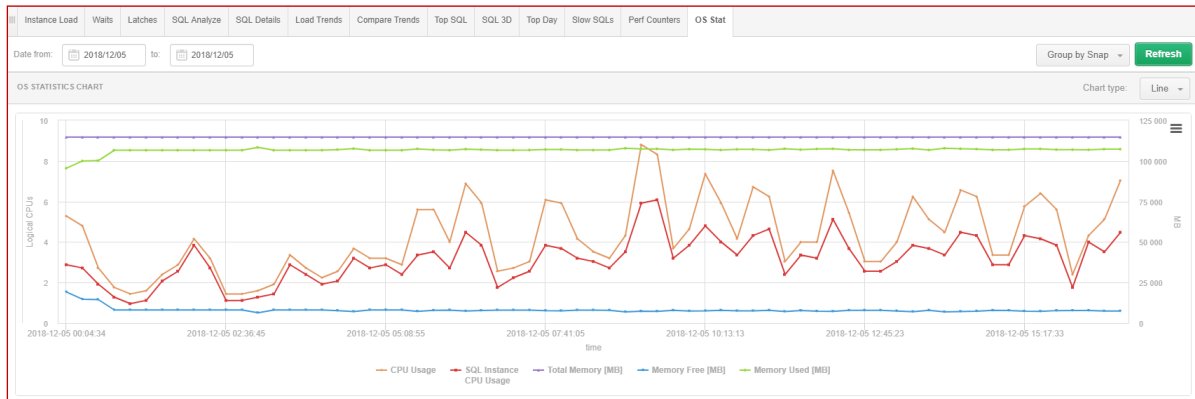


We perform exports for statistics previously selected from the list by right clicking and selecting one of the two available options: "Export grid" or "Export grid with formatted data".

Date from:	2018/12/01	to:	2018/12/06	Group by:	Hour	Refresh
HISTORY						
PERFORMANCE COUNTERS STATISTICS DURING SPECIFIED PERIOD TIME						
com						
Name	Class					
Log Compressed Bytes Rcvd/sec	Database Mirroring					
Log Compressed Bytes Sent/sec	Database Mirroring					
Log Compression Cache hits/sec	Database Replica					
Log Compression Cache misses/sec	Database Replica					
Log Compressions/sec	Database Replica					
Log Decompressions/sec	Database Replica					
Page compression attempts/sec	Access Methods					
Pages compressed/sec	Access Methods					
SQL Compilations/sec	SQL Statistics					
SQL Re-Compilations/sec	SQL Statistics					
SELECTED COUNTERS DETAILS WITHIN SPECIFIED PERIOD						
Logdate	Batch Requests/sec/SQL Statistics	SQL Compilations/sec/SQL Statistics	SQL Re-Compilations/sec/SQL Statistics	Log Compressed Bytes Sent/sec/Database Mirroring	Log Compressions/sec/Database Replica	Log Decompressions/sec/Database Replica
2018-12-01 00	71 947	11 533	149	3 369 161 775	0	0
2018-12-01 01	242 011	25 191	8 675	713 239 856	0	0
2018-12-01 02	66 516	6 923		440 746 692	0	0
2018-12-01 03	73 074	9 089		6 138 484 890	0	0
2018-12-01 04	66 618	8 995		8 429 355 975	0	0
2018-12-01 05	805 520	111 743		1 792 198 025	0	0
2018-12-01 06	537 650	62 844	1 332	2 160 688 099	0	0
2018-12-01 07	165 956	95 959	33 120	765 258 730	0	0

6.2.1.13 OS Stat Tab

The tab presents information about statistics collected at the operating system level.



The OSS Stat tab contains the following information as:

- Logical CPUs – number of available processors,
- SQL Instance Logical CPUs – number of available processors on SQL Instance,
- CPU Idle [Seconds] – the number of processor inactivity seconds, relative to all processors
- CPU Usage [Seconds] - number of seconds in which the processor was busy executing the user or kernel code, including all processors on the server
- SQL Instance CPU Usage – [Seconds] - number of seconds in which the processor was busy executing the user or kernel code, including all processors on the SQL Instance,
- Total Memory [MB] - total amount of physical memory.
- Memory Free [MB] – total amount of free physical memory.
- Memory Used [MB] - total amount of used physical memory.

OS STATISTICS	Logical CPUs	SQL Instance Logical CPUs	CPU Idle	CPU Usage	SQL Instance CPU Usage	Total Memory [MB]	Memory Free [MB]	Memory Used [MB]	Clear selection
2018-12-05 00:04:34	16	16	10.72	5.28	2.88	114 687 MB	19 158 MB	95 529 MB	
2018-12-05 00:19:47	16	16	11.20	4.80	2.72	114 687 MB	14 723 MB	99 964 MB	
2018-12-05 00:35:00	16	16	13.28	2.72	1.92	114 687 MB	14 499 MB	100 188 MB	
2018-12-05 00:50:13	16	16	14.24	1.76	1.28	114 687 MB	8 096 MB	106 591 MB	
2018-12-05 01:05:26	16	16	14.56	1.44	0.96	114 687 MB	8 092 MB	106 595 MB	
2018-12-05 01:20:39	16	16	14.40	1.60	1.12	114 687 MB	8 091 MB	106 596 MB	
2018-12-05 01:35:52	16	16	13.60	2.40	2.08	114 687 MB	8 092 MB	106 595 MB	
2018-12-05 01:51:05	16	16	13.12	2.88	2.56	114 687 MB	8 093 MB	106 594 MB	

6.2.2 Plan Guides Menu

The Plan Guides menu has been added, available from Instance Analysis for each instance. Information on the Plan Guides established in a given instance is available on the screen. Current information as well as historical data are available.

Back to dashboard

Performance

Plan Guides

Anomaly monitor

I/O Stats

Space monitor

Memory

Sessions

Backups

Locks

Parameters

Logs

Reports

Version: 2018.4.2

Plan Guides Overview

Plan Guides History

Plan guides for All databases Filter by Query Hash

Include dropped plan guides

Refresh

CURRENT PLAN GUIDES LIST

If plan guide doesn't contain query hash information it could mean that query is executed very fast or plan guide is not used.

Search by any value in below plan guide list

Database	Name	Create date	Last modify	Is Disable	Statement text	Query Hash	Scope	Scope object name	Scope object type	Parameters	Hints
Navision	DBPLUS_Ox40B73F	2018-08-02 12:07:11	2018-08-02 12:07:11	<input checked="" type="checkbox"/>	SELECT * FROM "N 0x1D7FE64668F1 SQL		SQL			@P1 int,@P2 int,@F OPTION(TABLE HIN	
Navision	DBPLUS_Ox28C51A	2018-04-25 12:28:13	2018-04-25 12:28:13	<input type="checkbox"/>	SELECT TOP 1 NUL	0x098C05A6360C	SQL			@P1 int,@P2 int,@F OPTION(USE PLAN	
Navision	DBPLUS_Ox6995F8	2016-08-09 08:38:08	2016-08-09 08:38:08	<input type="checkbox"/>	SELECT TOP 1 * FR		SQL			@P1 varchar(20),@F OPTION (TABLE HIR	
Navision	DBPLUS_Ox291762	2015-09-18 10:10:50	2015-09-18 10:10:50	<input type="checkbox"/>	SELECT TOP 1 NUL		SQL			@P1 varchar(20),@F OPTION (TABLE HIR	
Navision	DBPLUS_Ox63DA71	2015-05-14 14:35:47	2015-05-14 14:35:47	<input type="checkbox"/>	SELECT TOP 1 NUL		SQL			@P1 varchar(20),@F OPTION (TABLE HIR	
Navision	DBPLUS_Ox67F504	2015-02-16 15:02:01	2015-02-16 15:02:01	<input type="checkbox"/>	SELECT TOP 1 NUL		SQL			@P1 varchar(20),@F OPTION (TABLE HIR	
Navision	DBPLUS_Ox65BCAE	2015-02-16 14:00:45	2015-02-16 14:00:45	<input type="checkbox"/>	SELECT * FROM "U		SQL			@P1 varchar(30) OPTION (TABLE HIR	
Navision	DBPLUS_Ox176387	2015-02-04 15:06:52	2015-02-04 15:06:52	<input type="checkbox"/>	SELECT TOP 1 NUL		SQL			@P1 varchar(50),@F OPTION (TABLE HIR	

DETAILS FOR SELECTED PLAN GUIDE

SQL Text & Hints

Changes history

STATEMENT TEXT

```
SELECT * FROM "Navision -.dbo"."Inter Cars UASAMM Document Reader" WITH (READUNCOMMITTED) WHERE (("Document Type"< >@P1)) AND (("Document Status"< >@P2)) AND (("Location Code"< >@P3)) AND (("Warehouse Document No."< >@P4)) AND (("Warehouse Document Type"< >@P5)) AND "Document Type"> @P6 ORDER BY "Document Type","Document Status","Document ID" OPTION (OPTIMIZE FOR UNKNOWN)
```


The Plan Guide Overview tab contains the following information as:

- Database name
- Name – Plan Guide name,
- Create date
- Last modify – date of last modification
- Is Disable – information about Plan Guide status,
- Statement text
- Query Hash – query ID assigned with Plan Guide
- Scope – [OBJECT/SQL/TEMPLATE]
- Scope object name
- Scope object type – (e.g. procedure, functions)
- Parameters – list of parameters
- Hints – hints related with Plan Guide

Note! Not all Guide Plan will be assigned Query Hash. This will refer specifically to those Guide Plans that have been created in the past and for which queries are not currently performed.

After clicking on the row in the table, below (the SQL Text & Hints tab), the content of the query will be presented as well as the used hints within the plan. The Changes History tab presents information about what changes were made to a given Plan Guide (e.g. Insert / Change / Drop)

In addition, the Plan Guide History tab stores information about all Plan Guide in the SQL Instance. To search for a Historical Plan Guide, select the appropriate date range.

6.2.3 Menu Anomaly Monitor

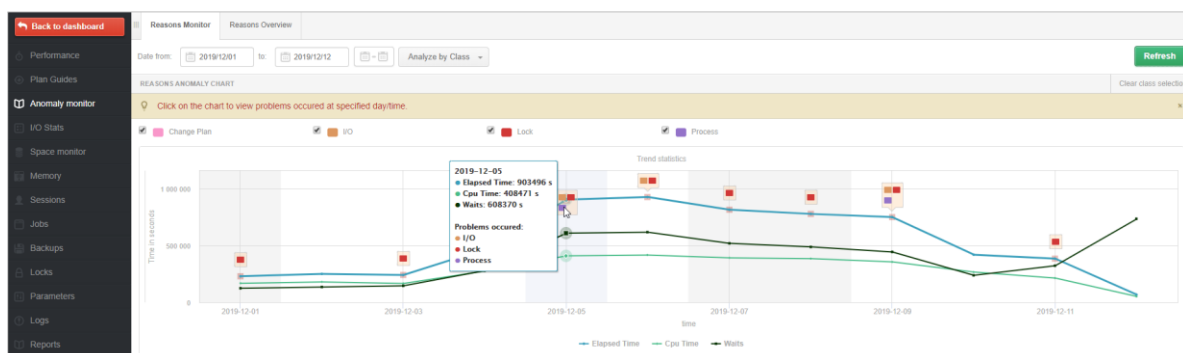
The Anomaly Monitor allows to viewing anomalies (alerts) generated on the database. The browser is available from the Instance Analysis> Anomaly Monitor SQL instance details

6.2.3.1 Alerts viewer in the SQL instance

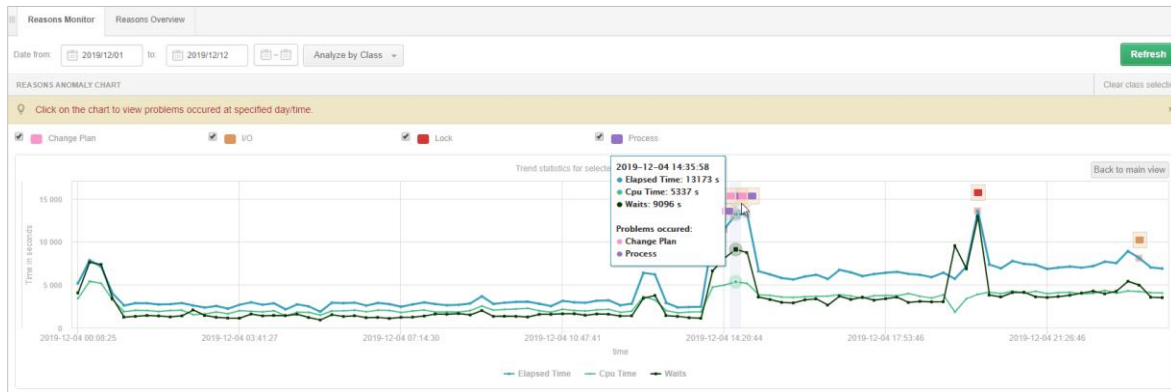
On the page user can choose between two tabs; Reasons Analysis and Reasons Overview.

6.2.3.1.1 Reasons Analysis

The Anomaly Monitor module has been improved in the new version of the application. The method of alert detection and presentation has been modified. After enter the screen, a graph from the last 14 days is presented where a performance problem occurred. The date range can be freely modified. By default, the screen presents problems grouped by class (Analyze by Class), it is also possible to change the presentation and group them by reason of problems (Analyze Reason).



Problems on the chart are marked by colored icons (a different color for each class / reason). For further analysis, select the indicated day on which the problems occurred. After select a specific day (point on the graph) a detailed graph for a given day will be presented with an indication of the point at which performance problems occurred. Each point on the graph represents a given snap (15 minutes). By select a point on the chart, the user will receive information on statistics that have been exceeded at the moment as well as information on the cause of the problem.



In the new version, the Anomaly monitor module has been extended with problem detection, which additionally analyzes database performance at a given time and presents the result of this analysis in the form of a problem. This module is embedded in the application code and is not user configurable. The current alert mechanism works all the time independently of the detection mechanism.

PROBLEMS REPORTED IN SPECIFIED TIME 2019-12-04 19:55:28	
Increase of query processing time caused by locks	
Class	Lock
Reason details & action	Following process was the main blocker session that generated locking. Logdate: 2019-12-04 19:55:32, SessionId: 398, Username: INTER\svrvmssql, Status: running, OS User: svrvmssql, Program: SQLAgent - TSQL JobStep (Job 0xB11C4134C2AF0E40ABCE68F6FDA1C259 : Step 1), Transaction log record count: 1308317, Last Request Runtime: 1704 s, Transaction begin: 2019-12-04 19:33:33, Transaction log size: 8554.4 MB
Additional information	Please go to Locks→Locks history module and analyze blocking cases at specified time.
Lock Time	Alert Type: Load Trends, The measured statistic value is 134 % higher than average, Last value: 8872 s, Reference history value: 3784 s
Elapsed Time	Alert Type: Load Trends, The measured statistic value is 90 % higher than average, Last value: 13564 s, Reference history value: 7140 s

As part of defining causes of the problem in the Alerts settings menu in the "Reasons & Problems definitions" tab for a given cause of the problem, user can specify and add a detailed description of the problem with an indication of the place for detailed analysis.

REASON DEFINITION

Main description

Data writes time problem caused by slow I/O response

Reason Class

I/O

Details description

Slow data writes problem is detected. For detailed verification, go to the I/O Analyze tab in the I/O Stats menu.

Hints for further analysis

Calculation Type

Based on Trends

Enabled

☒

Rules & Formulas

Notifications & Conditions

AND OR

Trends:Elapsed Time

Delete

AND OR

IO:Single Block Write time

Delete

IO:Write time

Delete

NOT:IO:Disk writes

Delete

Add rule

Add group

Rules preview: Trends:Elapsed Time AND (IO:Single Block Write time OR IO:Write time) AND NOT:IO:Disk writes AND (Trends:Wait Event Time - [buffer busy waits] OR Trends:Wait Event Time - [free buffer%])

6.2.3.1.2 Reasons Overview

As part of this tab, the application allows you to view problems in one set. We can choose the same filters as for the Reasons Analysis tab and additionally the option of marking / deselecting grouping after the Cause.

Reasons Analysis | Reasons Overview

Date from: 2018/12/03 to: 2018/12/17 Show reason type: Trends Online Using Query Hash: Enter query hash ☒ Group by reason **Refresh**

Hide additional filters

Reasons list
Search by name ...
Performance problem for specified SQL Statements
Increase of waits events (cause of Locks) on database
Problem - wait: PAGEIOLATCH_SH
Performance problem for specified SQL Statements
Performance problem for specified SQL Statements
Performance problem for specified SQL Statements

Reasons selected to filter

Alerts list
Search by name ...
IO Disk reads
IO Disk writes
IO MB reads
IO MB writes
IO Read time
IO Single MB Read time

Alerts selected to filter

REASONS & ALERTS OVERVIEW

Logdate	Reason name
2018-12-14 14:29:23	I/O>Data reads time problem caused by slow I/O response Read time Alert Type: I/O Stat, The measured statistic value is 2.6 times higher than allowed maximum , Last value: 32871 s, Reference history value: 9204 s Single MB Read time Alert Type: I/O Stat, The measured statistic value is 64 % higher than allowed maximum , Last value: 0.0425 s, Reference history value: 0.0258 s Elapsed Time Alert Type: Load Trends, The measured statistic value is 5.6 times higher than average , Last value: 482.8 s, Reference history value: 72.8 s
2018-12-14 14:29:23	I/O/increase of query processing time caused by slow I/O response Single MB Write time Alert Type: I/O Stat, The measured statistic value is 3.5 times higher than allowed maximum , Last value: 0.1000 s, Reference history value: 0.0224 s Single MB Read time Alert Type: I/O Stat, The measured statistic value is 64 % higher than allowed maximum , Last value: 0.0425 s, Reference history value: 0.0258 s Cpu Time Alert Type: Load Trends, The measured statistic value is 11 times higher than average , Last value: 437.5 s, Reference history value: 36.3 s

Depending on the checkbox **[Group by reason]**, alert data will be displayed in various lists:

- selected

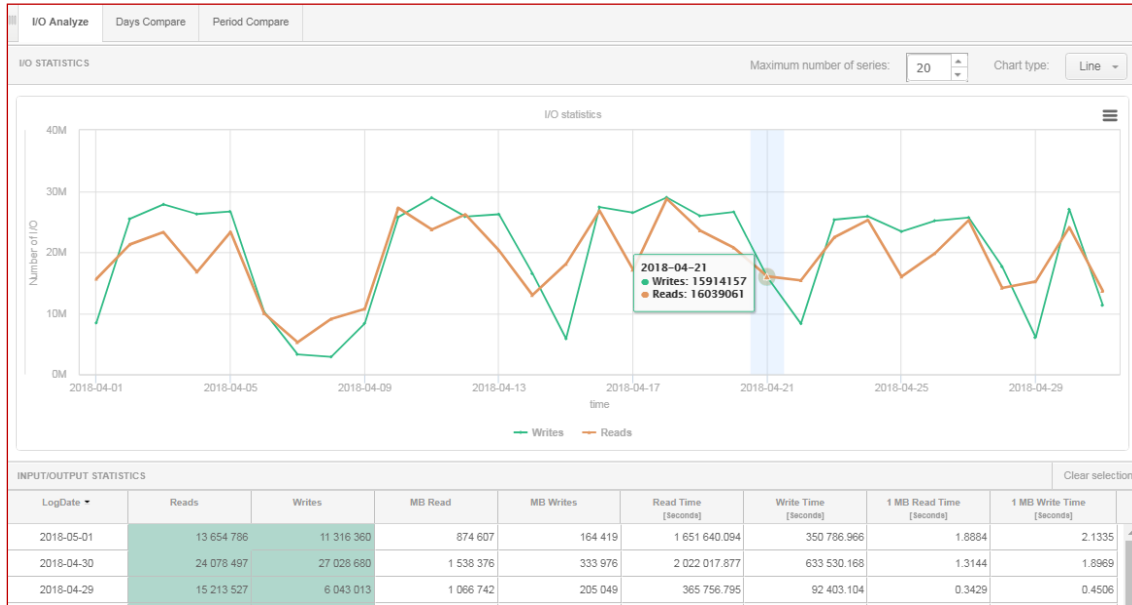
REASONS & ALERTS OVERVIEW	
Logdate	Reason name
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response
	Single Block Write time Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s
	Write time Alert Type: I/O Stat, The measured statistic value is 2.6 times higher than allowed maximum , Last value: 10137 s, Reference history value: 2849 s
	Wait Event Time Alert Type: Load Trends, The measured statistic value is 119 % higher than average , Wait: log file sync, Last value: 60.6 s, Reference history value: 27.6 s
	Elapsed Time Alert Type: Load Trends, The measured statistic value is 66 % higher than average , Last value: 1769 s, Reference history value: 1067 s

- unselected

REASONS & ALERTS OVERVIEW					
Logdate	Reason	Level	Alert name	Hash value	Message
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Single Block Write time		Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Write time		Alert Type: I/O Stat, The measured statistic value is 2.6 times higher than allowed maximum , Last value: 10137 s, Reference history value: 2849 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Wait Event Time		Alert Type: Load Trends, The measured statistic value is 119 % higher than average , Wait: log file sync, Last value: 60.6 s, Reference history value: 27.6 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Warning	Elapsed Time		Alert Type: Load Trends, The measured statistic value is 66 % higher than average , Last value: 1769 s, Reference history value: 1067 s
2018-12-02 06:32:14	I/O/increase of query processing time caused by slow I/O response	Critical	Single Block Write time		Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s

6.2.4 I/O Stats Menu

The screen is accessed from the menu on the left side and is used to analyze the performance of disk components. [I/O Analyze] functionality allows you to see any performance problems on disk devices, among others comparing the performance of writing and reading for specific days, hours, data files, tablespaces as well as collectively for the entire *sql instance*.



The area is divided into the following parts:

- Filter area with date range and additional filters
- A graph for presenting specific indicators

Table showing statistics of:

- Reads - the number of reads
- Writes - the number of data writes by DBWR
- MB Reads - the number of read blocks
- MB Writes – the number of wrote blocks
- Read Time - time to read blocks
- Write Time - time to write blocks
- 1MB Read Time - time to read a one MB
- 1 MB Write Time - time to write a one MB

Filter **[Group by period]** - shows statistics for a given query grouped according to the choice:

- **No group by period** - no grouping i.e. selection date ranges from 1 to 20 days of the month will show summary statistics for the selected period
- **Day** - shows statistics for a given query, broken down into periods of one day
- **Hour** - shows statistics for a given query, broken down into periods of one hour
- **Snap** - shows statistics for a given query, broken down by snapshots - periods of 15 minutes

On the website a possibility of a comparative analysis for I / O statistics is available. Days Compare page consists of two options to compare data: grouped for hour or by snap. Comparing consists of adding specific days from the calendar and adding them to the graph.

6.2.5 Space Monitor Menu

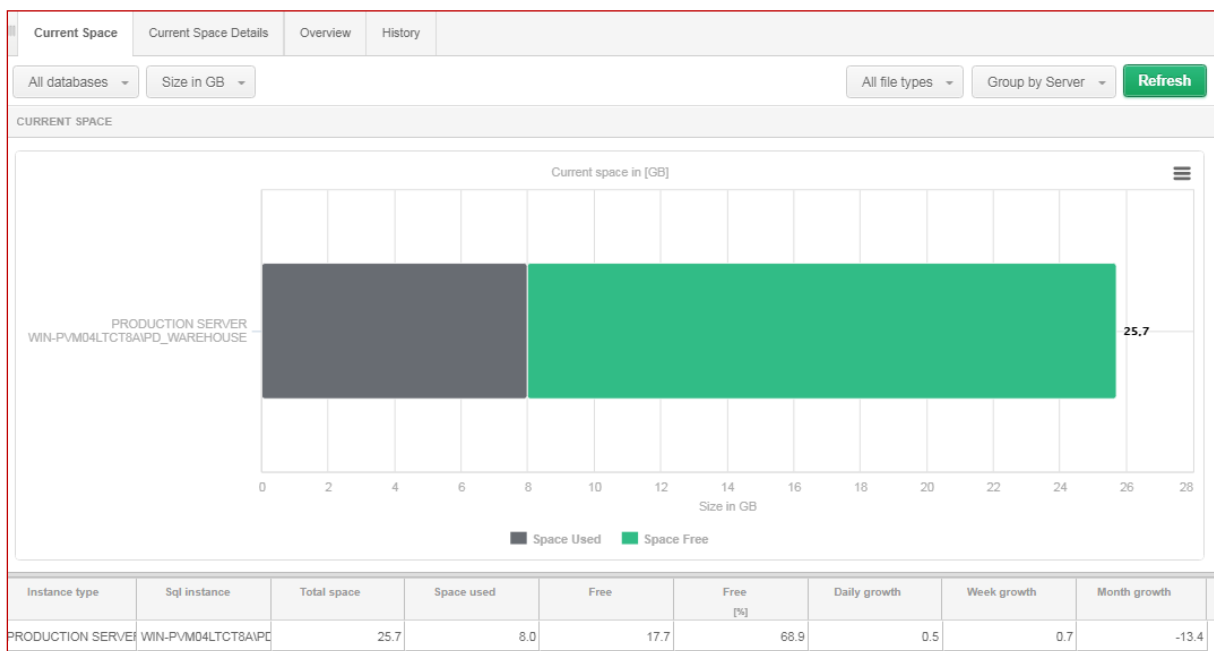
The **Space Monitor** module allows for SQL Instance storage analysis. Three options are provided:

- Display the current size of SQL Instance
- Detailed information on database occupancy (by database objects)
- The history of SQL Instance size change in table and graphical form

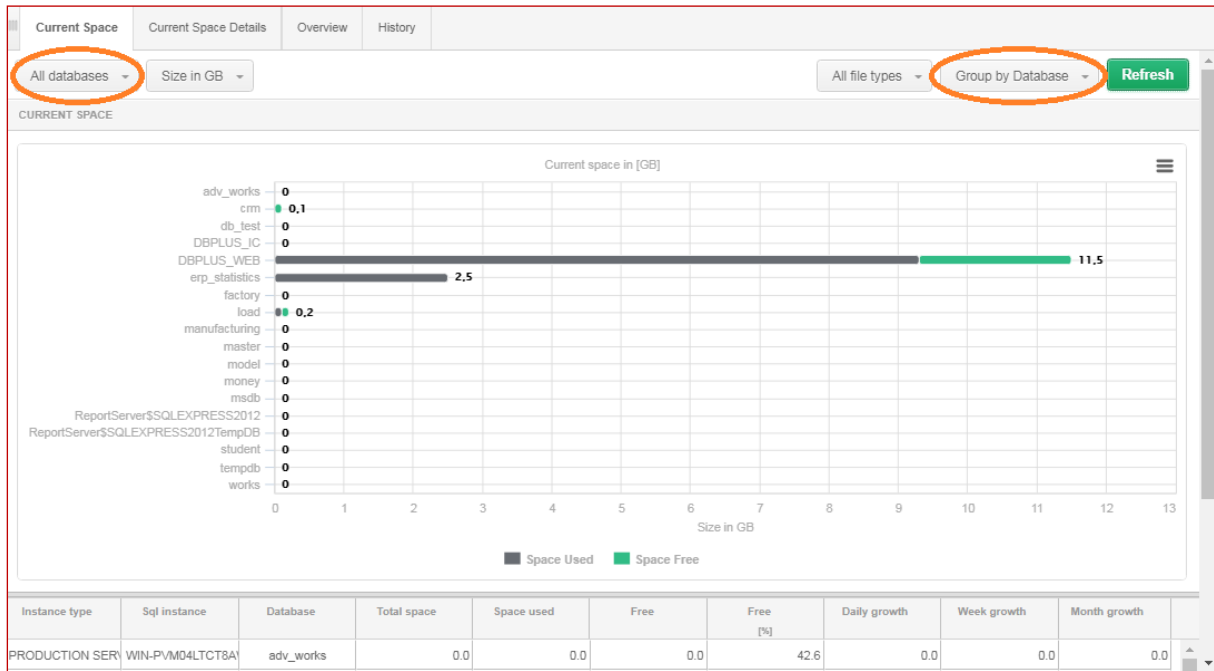
IMPORTANT: [Space Monitor] module is also accessible from the main page (click [Back to dashboard] button) this allows to analyze the space used by SQL Instance.

6.2.5.1 Current Space Tab

The Current Space tab shows the current size of all databases in SQL Instance or specific database and its size over time.



In the filter, you can change the grouping method and set the filter to a specific database:



In the table below the chart, user can see:

- The size of the all databases in SQL instances (or specific database)
- Used space
- Statistics on size increments

6.2.5.2 Current Space Details Tab

In the Current Space Details Tab, the user can check the size of individual database objects. The page consists of 2 grids:

- Object size by type - presents data grouped by object type
- Object size - lists individual objects with their size

Current Space								
Current Space Details								
Overview								
History								
DBPLUS_WEB								
Size in GB								
Refresh								
CURRENT SPACE DETAILS								
Object size by type				Object size				
Segment	Schema	Size [GB]	Rows count	Segment	Schema	Object name	Size [GB]	Rows count
USER_TABLE	dbo	5.471	25 M	USER_TABLE	dbo	dbplus_tab17	1.841	12 191 266
INTERNAL_TABLE	sys	0.000	0	USER_TABLE	dbo	dbplus_tab19	1.554	2 802 733
SYSTEM_TABLE	sys	0.003	10 k	USER_TABLE_INDE	dbo	dbplus_tab17_dbplus	1.079	0
USER_TABLE_INDE	dbo	3.624	0	USER_TABLE_INDE	dbo	dbplus_tab17_dbplus	0.992	0
INTERNAL_TABLE_INDE	sys	0.000	0	USER_TABLE_INDE	dbo	dbplus_tab17_dbplus	0.357	0
SYSTEM_TABLE_INDE	sys	0.001	0	USER_TABLE	dbo	dbplus_tab8sd	0.268	2 066 945
				USER_TABLE	dbo	dbplus_tab4	0.263	485 344
				USER_TABLE	dbo	dbplus_tab2_day	0.244	909 516
				USER_TABLE	dbo	dbplus_errlog	0.233	3

6.2.5.3 Overview Tab

In the Overview Tab, the page presents the size of instances / databases on particular days in the form of a table. The filter area allows you to analyze the space:

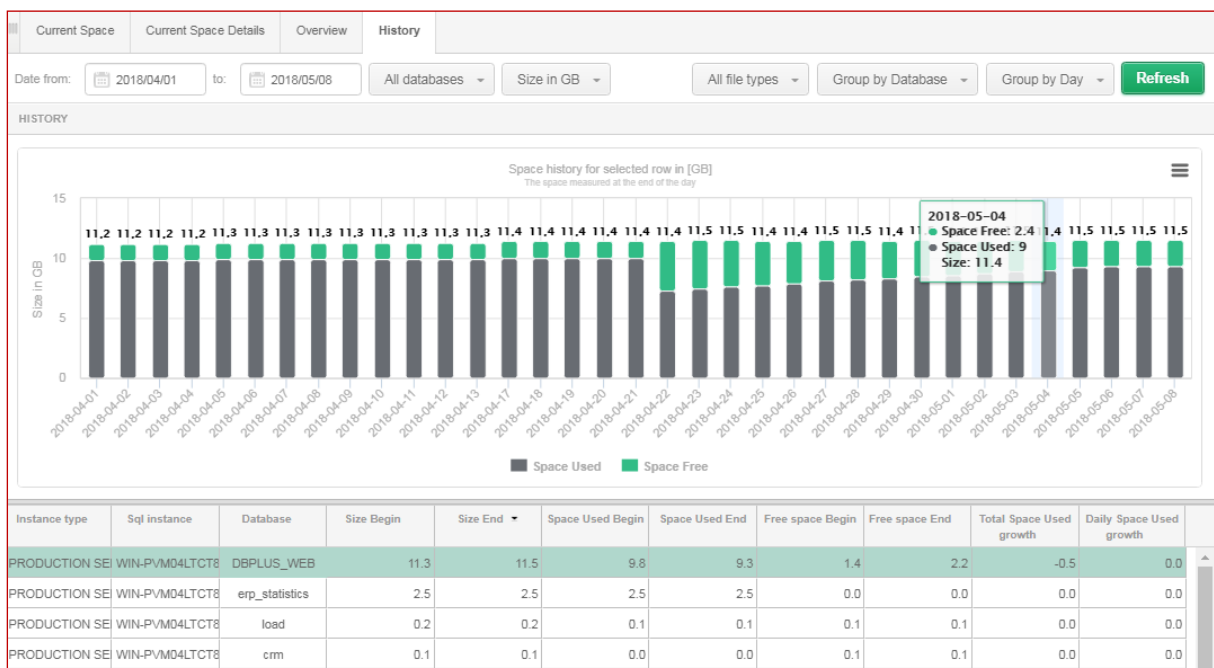
- In the selected time horizon
- For all or a specified database
- For specific file types
- With a specific grouping

Current Space		Current Space Details		Overview		History	
Date from: 2018/05/01		to: 2018/05/08		All databases		Size in GB	
				All file types		Group by Database	
				Group by Day		<div>Refresh</div>	
OVERVIEW							
Date	Instance type	Sql instance	Database	Total space	Space used	Free	Free [%]
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	acs_repair	0.0	0.0	0.0	44.5
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	adv_works	0.1	0.0	0.1	56.2
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	advworks	0.0	0.0	0.0	48.1
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	crm	0.0	0.0	0.0	35.2
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	db_test	11.5	8.6	2.8	24.9
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	DBPLUS_IC	2.5	2.5	0.0	0.6
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	DBPLUS_WEB	0.0	0.0	0.0	51.8
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	erp_statistics	0.0	0.0	0.0	31.8
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	factory	0.0	0.0	0.0	51.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	load	0.0	0.0	0.0	31.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	manufacturing	0.0	0.0	0.0	51.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	master	0.0	0.0	0.0	31.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	model	0.0	0.0	0.0	51.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	money	0.0	0.0	0.0	31.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	msdb	0.0	0.0	0.0	51.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	ReportServer\$SQLEXPRESS2012	0.0	0.0	0.0	31.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	ReportServer\$SQLEXPRESS2012TempDB	0.0	0.0	0.0	31.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	student	0.0	0.0	0.0	31.7
2018-05-01	PRODUCTION SERVER	WIN-PVMD4LTCT8A\SQL	tempdb	0.0	0.0	0.0	31.7

6.2.5.4 History Tab

In the History tab, the page shows the size SQL instance (or specific databases) for each day of the selected date range.

Graphs can be presented for a given period, grouped by day or time.



6.2.6 Memory Menu

The **Memory** module allows to analyze the memory utilization in the database. The tabs provide options to:

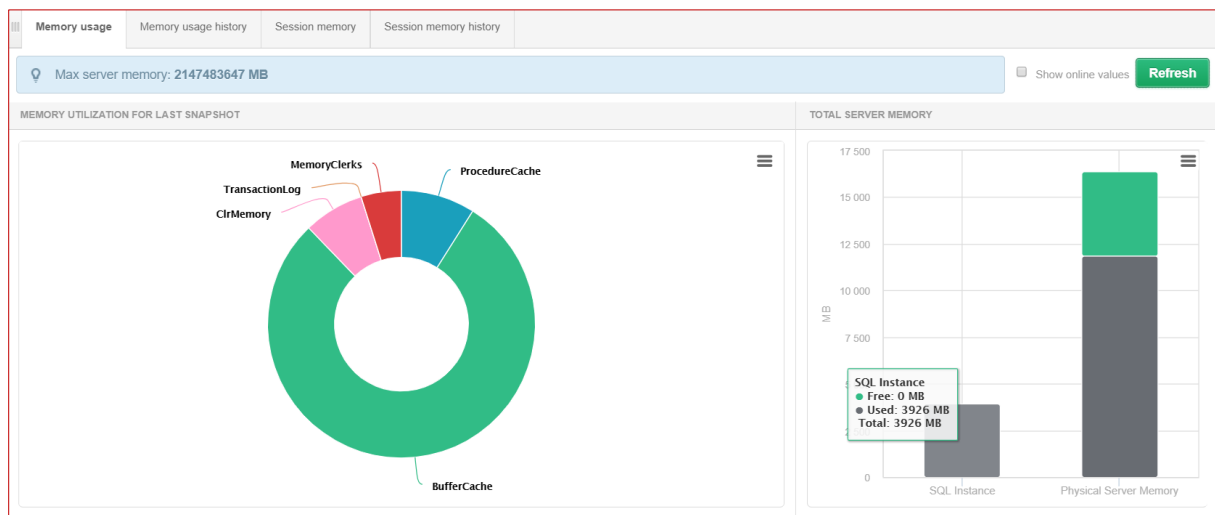
- Display the current memory usage,
- Memory usage history over time,
- The use of memory by user sessions.

6.2.6.1 Memory usage Tab

The **Memory Usage** tab presents the settings of individual SQL instance parameters and utilization of memory by instance broken down into memory areas, i.e.:

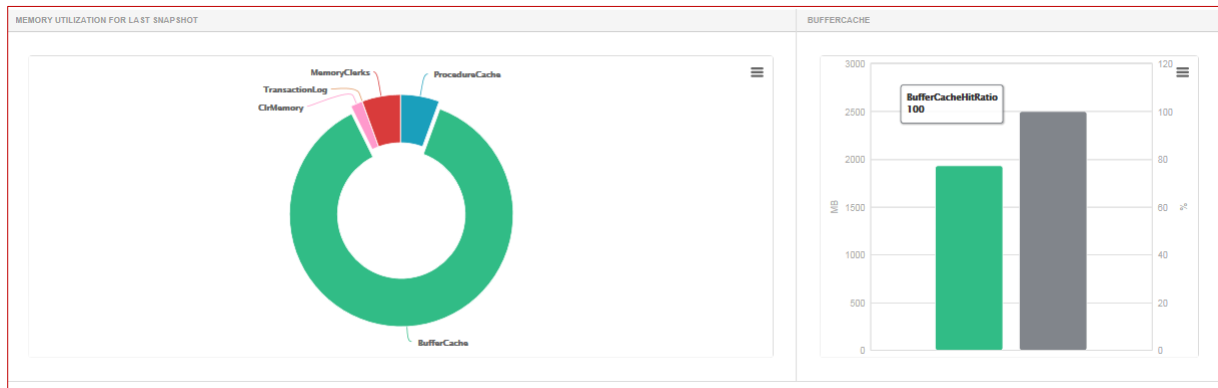
- Buffer Cache (Buffer Pool)
- Procedure Cache
- Others areas (Memory Clerks, Transaction Log, i.e.)

Parameter settings are shown as components of the wheel. The sizes of individual parameters are expressed in MB. Clicking on the interesting fragment of the wheel showing the given parameter will show the statistics of this parameter using the graph on the right side of the Memory wheel. The statistics graph has a different characteristic for each parameter.



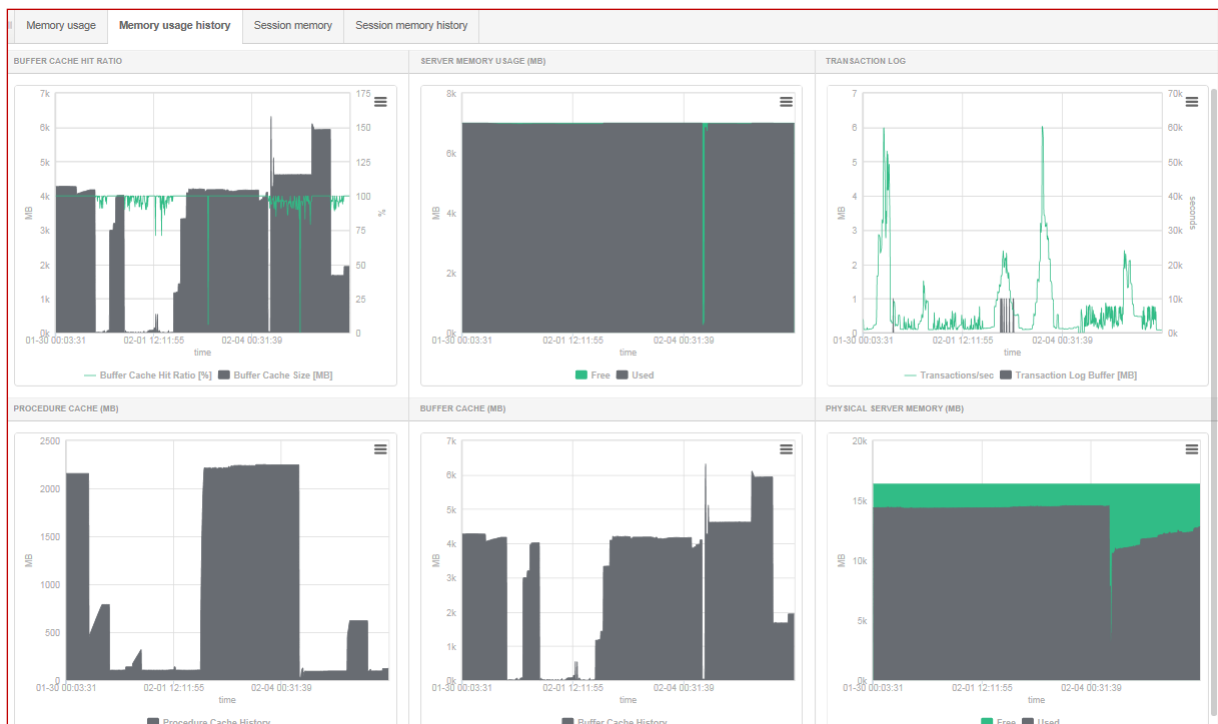
In addition to the size of memory utilization by SQL instances, user also can see the use of memory on the server - the right column of the chart.

After clicking on the Buffer Cache area, the right part of the chart is refreshed accordingly:



6.2.6.2 Memory usage history Tab

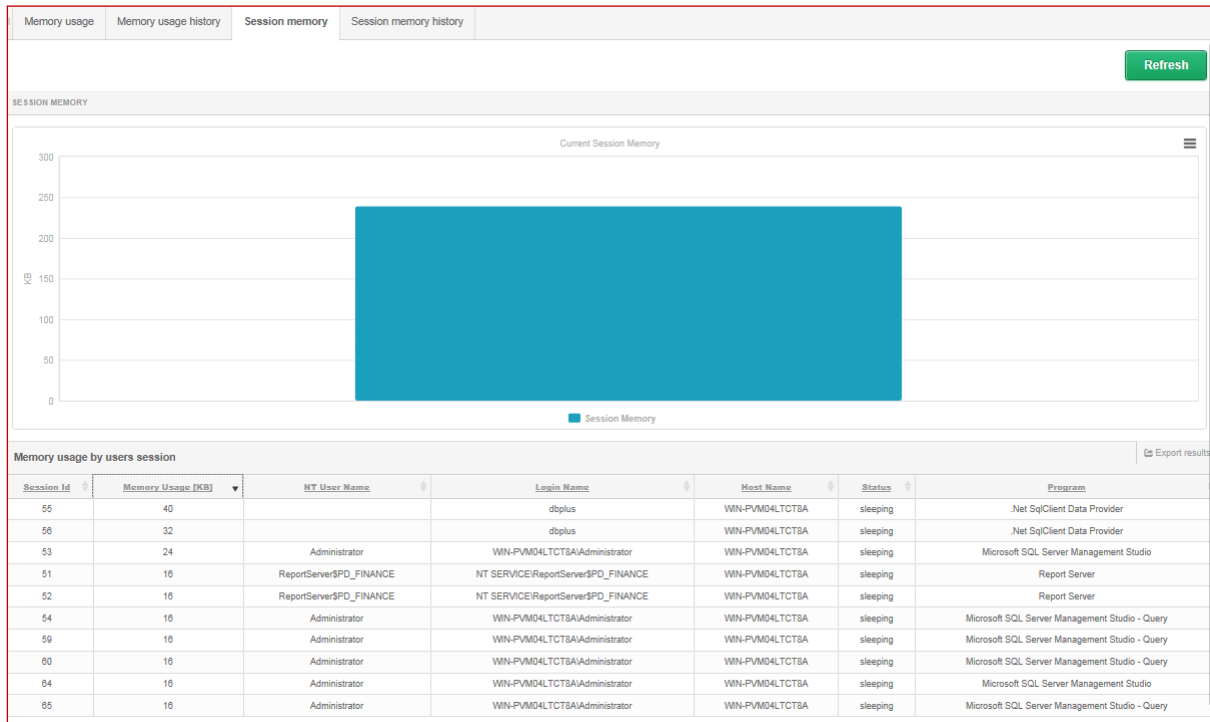
The Memory Usage History tab shows the history of memory utilization over time. In the filter area, you can specify the way of grouping data, according to which graphs for particular areas of memory will be displayed:



6.2.6.3 Session Memory Tab

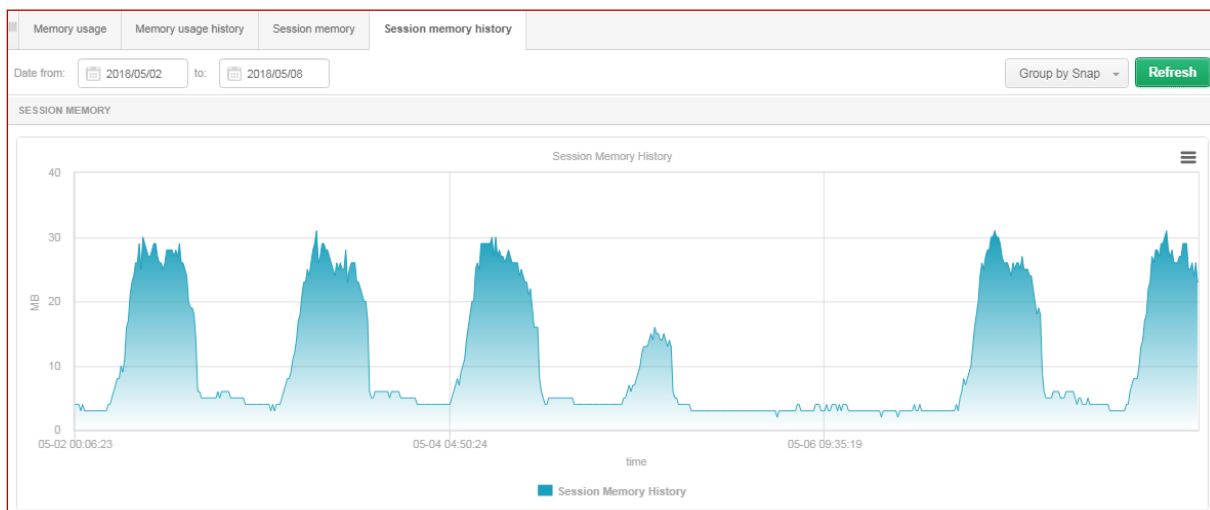
The Session Memory Tab is divided into two parts:

- Bar graph showing the amount of current memory that sessions occupy
- Tables showing the current utilization of memory by sessions



6.2.6.4 Session memory history Tab

As with the Memory usage history tab, the user can check the amount of memory during the utilization of user sessions.



6.2.7 Sessions Menu

Sessions functionality presents information about sessions in the SQL Instances. From the level of upper tabs, access is provided to:

- **Sessions** – sessions in the database displayed according to the criteria in the filters,
- **Temp usage sessions** – a screen that allows for session analysis for the use of temporary space (e.g. a session that performs a query that sorts a large amount of data),
- **Log usage session** – functionality that allows for session analysis for the use of log (e.g. sessions holding a large portion of data in an uncommitted transaction)
- **Session chart history**
- **Active sessions/Log usage sessions history** – Field allowing for following searches:
 - What queries the program / user runs?
 - Which users the specified query hash is run

6.2.7.1 Sessions Tab

The Session tab shows session information on the SQL instance.

After selecting the appropriate filters, session information will be presented:

- **Logon Time** – time of user logging into SQL Instances
- **Session id** – user session ID
- **Query Hash** – identifier of the currently executed command, (it means that the command is currently being executed with accuracy provided by SQL instance).
- **Login name** – database user name,
- **Status** – status of the session: running, sleeping
- **Windows username** – user name in the operating system on which the SQL instance database was logged in,
- **Hostname** – the name of the machine the SQL Instance was logged
- **Program** – the name of the program you logged in to the,
- **Context Info** – the value of context info parameter, set at the session level
- **Blocking session** - the number of the parent session that blocked the current session (when the value is greater than zero)
- **Database** - the database in the context of which the SQL command is executed
- **Wait** – name of the session wait type,

- **Wait time**

Selected rows provide following information below:

- **SQL sub-tab**

It contains the text of the SQL query. The information is displayed only for queries for which Hash value is stored.

The screenshot shows the SQL sub-tab with the following content:

STATEMENT TEXT

```
select r.query_hash, (SUBSTRING(st.text,statement_start_offset/2 +1,
(CASE WHEN statement_end_offset = -1
THEN LEN(CONVERT(nvarchar(max), text)) * 2
ELSE statement_end_offset end -
statement_start_offset
)/2+1
)) query_text, st.text as query_text_batch, st.encrypted, statement_start_offset, statement_end_offset from sys.dm_exec_requests r
cross join sys.dm_exec_sql_text(r.sql_handle) as st
```

EXPLAIN PLAN

Show plan objects for 0x9D14E5DB86D3212F

```
--SELECT (Cost = 0.00240782, Rows = 0, CPU = 0, IO = 0)
--Compute Scalar (Cost = 0.00240782, Rows = 72.0628, CPU = 0.000070628, IO = 0)
--Left Outer Join--Nested Loops (Cost = 0.00240061, Rows = 72.0628, CPU = 0.00103263, IO = 0)
--Filter (Cost = 0.00124897, Rows = 11.1809, CPU = 0.00005544, IO = 0)
--Compute Scalar (Cost = 0.00069437, Rows = 63, CPU = 0.0000063, IO = 0)
--Table-valued function ([SYSSESSIONS]) (Cost = 0.00063137, Rows = 63, CPU = 0.000063137, IO = 0)
--Inner Join--Nested Loops (Cost = 0.00124308, Rows = 22.095, CPU = 0.000043987, IO = 0)
--Stream Aggregate (Cost = 0.00040013, Rows = 1, CPU = 0.000013757, IO = 0)
--Table-valued function ([SYSREQUESTS]) (Cost = 0.000247198, Rows = 22.095, CPU = 0.000022352, IO = 0)
--Filter (Cost = 0.00036778, Rows = 10.1911, CPU = 0.0000106356, IO = 0)
--Table-valued function ([SYSREQUESTS]) (Cost = 0.000471598, Rows = 22.095, CPU = 0.000022352, IO = 0)
```

Plan Compilation Time: 11 ms
Sampled values used for parameters at plan compilation time
@a: N'running'

- **Operation Progress sub-tab**

Presents information about the status of the current operation being performed by the session in the SQL instance.

SQL	Operation progress	Statistics	Waiting tasks										
Cpu Time (seconds)	Memory usage (MB)	Reads (Blocks)	Writes (Blocks)	Buffer reads (Blocks)	Rows count	Total elapsed time (seconds)	Total schedule time (seconds)	Last request start time	Last request end time				
0.031	32.00 KB	0	0	0	0	0.007	0.007	2018-05-07 23:48:51	2018-05-07 23:50:20				

In the **Waiting tasks** tab, individual tasks / threads for the session are shown - below the screen with a multithreaded session

The screenshot shows the Sessions tab with the following content:

Sessions Temp usage sessions Log usage sessions Sessions chart history Active sessions / Log usage sessions history

☒ Active sessions ☒ Users only Min elapsed time: 0 sec. Sid: All databases Username: Refresh

Show additional filters

SESSION LIST (LAST REFRESHED: 14:00:03) Kill session

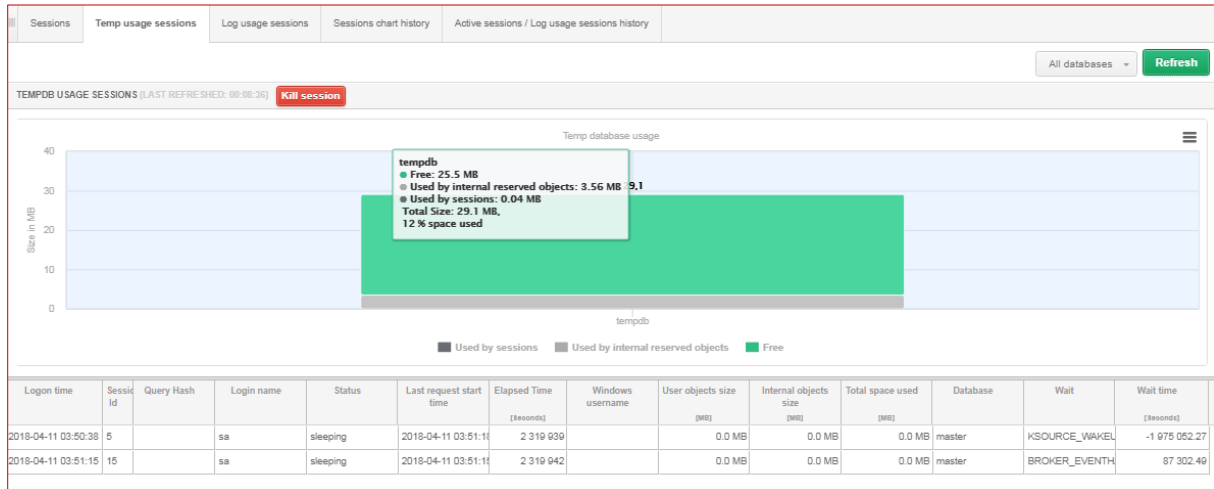
Logon time	SessId	Query Hash	Login name	Status	Last request start time	Elapsed Time [seconds]	Windows username	Host name	Program	Cont Info	Blockin session	Database	Wait	Wait time [seconds]
2018-04-23 13:59:14	55	0x8BD98CF9D17	MAQCH\radoslaw	running	2018-04-23 13:59:14	25	radoslaw	MAQCH	Microsoft SQL	0	dbplus_tester	dbplus_tester	LCK_M_S,CXPACKET	25.49
2018-04-23 14:00:03	62	0xFD543DE9FA0	dbgas	running	2018-04-23 14:00:03	0		MAQCH	DBPLUS Perf	0	DBPLUS	DBPLUS		0.00

Cases for selected multithreaded session:

SQL	Operation progress	Statistics	Waiting tasks											
Exec Context	Wait	Wait Time [seconds]	Blocking Session id	Blocking Exec Context	Resource Description									
0	CXPACKET	47.000	55	1	exchangeEvent id=Pipe90162680 WaitType=e_waitPipeGetRow nodeId=0									
0	CXPACKET	47.000	55	2	exchangeEvent id=Pipe90162680 WaitType=e_waitPipeGetRow nodeId=0									
2	LCK_M_S	47.000	58		keylock hobtid=72057594186301440 dbid=16 id=lock8a846300 mode=X associatedObjectid=72057594186301440									
1	LCK_M_S	47.000	58		keylock hobtid=72057594186301440 dbid=16 id=lock800b1900 mode=X associatedObjectid=72057594186301440									

6.2.7.2 Tempdb usage sessions Tab

In the next **[Tempdb usage sessions]** tab, sessions are presented for the use of temporary space.

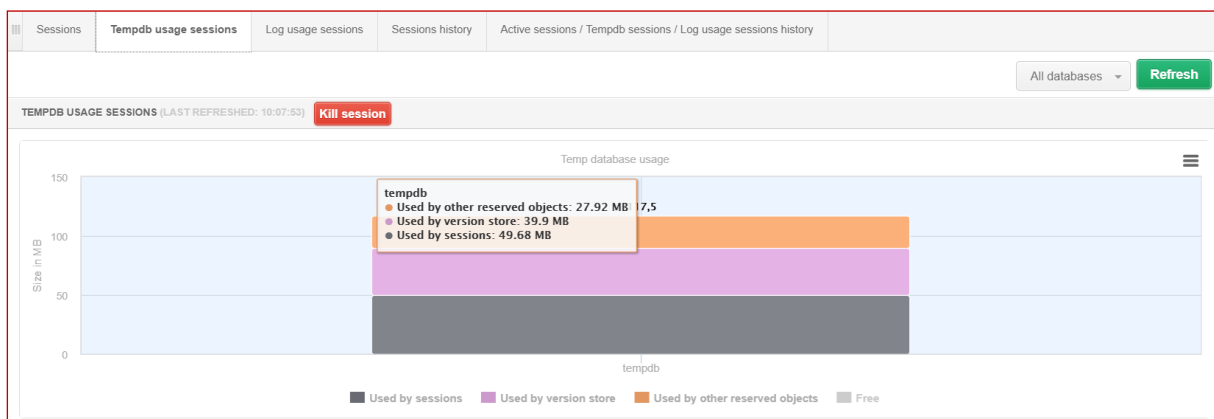


It should be noted that the Tempdb space can be occupied due to:

- Temporary objects - e.g. tables with prefix ##,
- Internal objects created by the database engine with support for hashing, joins, sorting

This information is visible on the chart in the Tempdb usage sessions tab. The chart contains information about:

- the used of memory in the Tempdb database via query version store,
- the used memory by sessions
- the used memory by other objects



The information about queries version store is useful when the "read_committed_snapshot with ON" parameter is enabled on the basis (the parameter is switched on so that "select" type queries do not block the query that make changes).

This setting generates additional entries in the Tempdb database because the change version is kept until the transaction is closed.

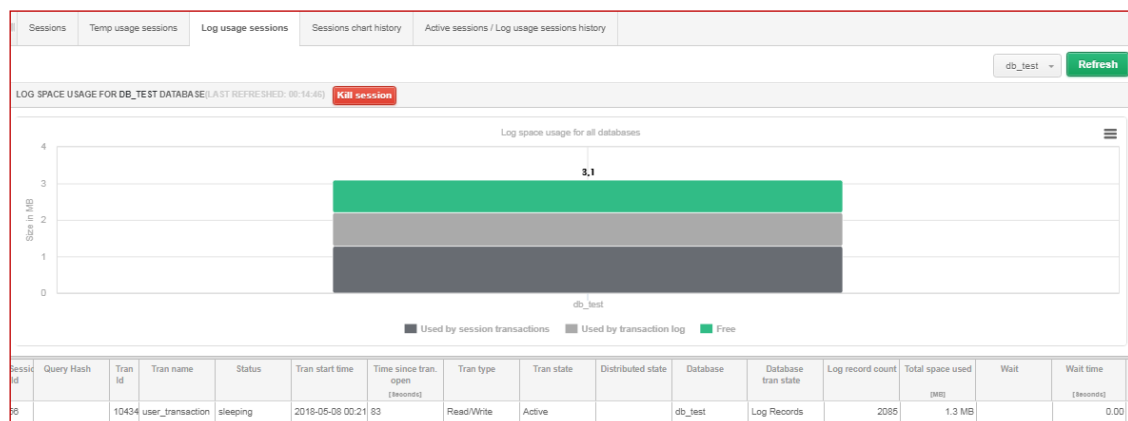
Below the chart there is a list of sessions where we have the same functionality as in the Sessions tab - clicking on the sessions shows in the sub-tabs the content of the query, information about the statistics, the status of the session.

6.2.7.3 Log usage sessions Tab

In the **[Log usage sessions]** tab, sessions presented for the use of the Log transaction space are located. The functionality allows to track those sessions that take up a large Log space (e.g. a large portion of updated data) or long period of time sessions.



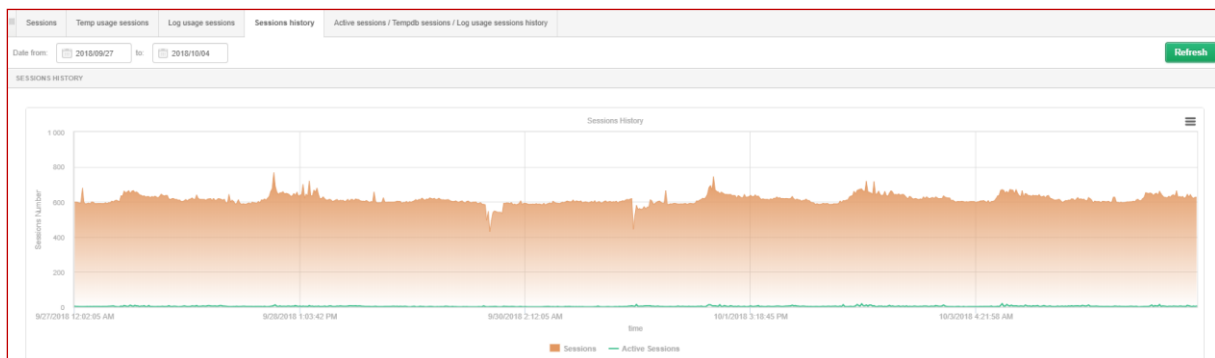
Below is the same example with selection for a database containing changes.



As in the previous tabs (i.e. Sessions, Temp usage sessions), below the chart is a list of sessions.

6.2.7.4 Session history Tab

On this page displays in form of a graph the number of active and inactive sessions in the selected time period.



6.2.7.5 Active Sessions/ Tempdb sessions/Log usage sessions history Tab

The active sessions / Tempdb sessions / log usage sessions history tab shows detailed information about open sessions at a given time:

The data in the table are divided into three groups:

- yellow color presents information about active sessions,
- green color shows information about sessions using Tempdb,
- red color shows information about sessions using Log.

Sessions

Temp usage sessions

Log usage sessions

Sessions history

Active sessions / Tempdb sessions / Log usage sessions history

From:

2018/10/02

00:00

to:

2018/10/02

23:59

Using Query Hash:

Enter query hash

Loginname:

Enter login/username

Sid:

Refresh

Show additional filters

ACTIVE SESSIONS / TEMPDB / LOG USAGE SESSIONS HISTORY

Toggle view:

Logdate	Active Sessions	Sessions using Tempdb	Tempdb Space Used [MB]	Log Usage Sessions	Log Usage Record Count	Log Space Used [MB]
2018/10/02 00:00:29	4	11	227.0 MB	0	0	0
2018/10/02 00:01:00	3	10	226.9 MB	0	0	0
2018/10/02 00:01:31	3	10	226.9 MB	0	0	0
2018/10/02 00:02:02	3	10	226.9 MB	0	0	0
2018/10/02 00:02:32	3	11	231.4 MB	0	0	0
2018/10/02 00:03:03	3	8	227.0 MB	0	0	0
2018/10/02 00:03:34	3	8	227.0 MB	0	0	0
2018/10/02 00:04:05	3	8	227.0 MB	0	0	0
2018/10/02 00:04:35	3	7	226.9 MB	0	0	0

Sessions

Tempdb usage

Log usage

Type	Session id	Program	Nt user name	Host name	Login name	Context Info	Query Hash	Plan Hash	Wait type	Wait time	Blocking session id	Command	Database	Elapsed Time	Cpu Time
										[Seconds]				[Seconds]	[Seconds]
Session	64	Microsoft SQL Se	crm	CRMSQL31	IC/crm				LCK_M_U	429.1	558	BACKUP DATAB	InterCars_MSCR	429.117	0.001
Session	87	SQLAgent - TSQ	crm	CRMSQL31	IC/crm		0xCFAF96834FFEE8	0xD85145F4D85EF	OLED	213.9	0	INSERT	IT	223.557	217.398
Session	558	SQLAgent - TSQ	crm	CRMSQL31	IC/crm				BACKUPIO,BAC	753.9	0	BACKUP DATAB	InterCars_MSCR	756.980	1.612
Session	569	MSCRMw3wp	crm_lisinter	CRMIS32 w3wp	INTER/crm_lisint		0x038B015C7EDC81	0xE22A43AC8E9A		0	0	SELECT	InterCars_MSCR	3.420	3.440

Clicking on the table record presents details for the selected snapshot in the **Sessions, Tempdb usage** or **Log Usage**.

In this case, following information are presented:

- Session Id – user session identifier,
- Program – the name of the system / program the session was launched,
- Host Name – the name of Hostname,
- Login name – user name,
- Nt user name – OS user name,
- Context Info – the value of context info parameter,
- Query Hash – the identifier of the command being executed,
- Plan Hash – the identifier of the execution plan,
- Wait type – specific type of wait,
- Wait time – wait time,
- Blocking session id - the number of the parent session that blocked the current session (when the value is greater than zero),
- Command – type of statement,
- Database – database name,
- Elapsed Time
- CPU Time

Below the screen of active sessions:

Sessions Log usage															
Type	Session Id	Program	Nt user name	Host name	Login name	Context Info	Query Hash	Plan Hash	Wait type	Wait time [seconds]	Blocking session id	Command	Database	Elapsed Time [seconds]	Cpu Time [seconds]
Session	54	.Net SqlClient De		WIN-PV/M04LTC			0x767529C73620CA	0xD3EB49A89CA9		0.0	0	SELECT	adv_works	0.013	0.018
Session	59	.Net SqlClient De		WIN-PV/M04LTC			0x0000000000000000	0x0000000000000000		0.0	0	SELECT	adv_works	1.506	1.123
Session	62	.Net SqlClient De		WIN-PV/M04LTC			0x677E3020F458809	0x60305CE494AA		0.0	0	SELECT	adv_works	0.000	0.000

Below the screen of Tempdb usage sessions:

Sessions		Tempdb usage		Log usage											
Session id	Status	Program	Nt user name	Host name	Login name	Context info	Query Hash	Query Plan Hash	Elapsed time <small>(seconds)</small>	User objects size <small>(MB)</small>	Internal objects size <small>(MB)</small>	Total space used <small>(MB)</small>	Database	Wait	Wait time <small>(seconds)</small>
84	running	SQLAgent - TSQL Job cm		CRMISL31	ICicm		0x3A7436263B72D33E		4 304.0	0	1 965.4 MB	1 965.4 MB	IT	PREEMPTIVE_COM	0
155	running	SQLAgent - TSQL Job cm		CRMISL31	ICicm		0x5F8E45127D6D27E		9 697.0	960.4 MB	18.3 MB	978.7 MB	IT	CXPACKET	0.32
53	running	SQLAgent - TSQL Job cm		CRMISL31	ICicm		0x428D428EAF5959		234.0	0.0 MB	194.6 MB	194.6 MB	IT	PREEMPTIVE_COM	0
384	sleeping	MSCRMMyService	cm_isinter	CRMIS31	CMisInter	INTERicm_isinter			3 159.0	37.4 MB	0.1 MB	37.5 MB			
86	sleeping	SQLAgent - Generic F cm		CRMISL31	ICicm				232.0	7.2 MB	0	7.2 MB			
52	sleeping	Microsoft SQL Server jobskows		DEVEL02	ICyjobskows				8.0	0.2 MB	2.4 MB	2.6 MB			
399	sleeping	MSCRMReportingSer	cm_srsinter	CRMISL31	Reporting	INTERicm_srsinter	b D'Ea:CYAChwC		126.0	0.1 MB	0	0.1 MB			
37	sleeping			sa					3 413 534.0	0	0.1 MB	0.1 MB	master	BROKER_EVENTHA	20.98

Below the screen of Log usage sessions:

Sessions		Log usage															
Session id	Status	Program	Nt user name	Host name	Login name	Context info	Query Hash	Transaction name	Trans begin	Tran last run time (seconds)	Tran type	Tran state	Distributed state	Database	Database tran state	Log record count	Space used (MB)
98	sleeping	.Net SqlClient	Administrator	WIN-PVMD4T	ADMINISTRATOR			user_transacti	2018-05-07 11	0.0	Read/Write	Active		adv_works	Log Records	118	0.1 MB

The application has ability to search information about the user's session using a given type of validity. We start the search by pressing the "Hide additional filters" button and then from the list of available waits we add the ones we want to view.

After pressing the Refresh button, only those sessions that were waiting, for a wait selected by the user from the list will be presented in the given period. At the same time, you can also select other filters, e.g. such as SID session ID or Hash Value.

Sessions

Tempdb usage sessions

Log usage sessions

Sessions history

Active sessions / Tempdb sessions / Log usage sessions history

From: 2018/12/06 00:00 to: 2018/12/06 23:59 Using Query Hash: Enter query hash

Logname: Enter loginusername

Sid:

Refresh

Hide additional filters

Performance Waits

PAGEIOLATCH_EX

PAGEIOLATCH_KP

PAGEIOLATCH_NL

SOS_PHYS_PAGE_CACHE

SOS_SMALL_PAGE_ALLOC

UTIL_PAGE_ALLOC

Waits selected to filtering

PAGEIOLATCH_EX

PAGEIOLATCH_SH

PAGEIOLATCH_UP

Hostname:

Context info:

All databases -

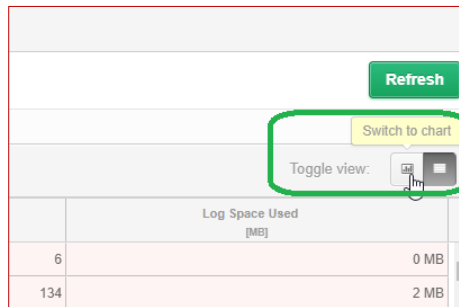
Logdate	Type	Session id	Program	Nt user name	Host name	Login name	Context info	Query Hash	Plan Hash	Wait type	Wait time (seconds)	Blocking session id	Command	Database	Elapsed Time (seconds)	Cpu Time (seconds)
2018-12-06 08:18:40	Session	142	MSCRMw3wp	cm_isinter	CRMIS32 w3wp	INTERicm_isinte		0x6A086E30708EC16	0x5E90DA7602EC4	PAGEIOLATCH_SH	0.1	126	INSERT	InterCars_MSCRM	0.097	0
2018-12-06 02:06:24	Session	56	SQLAgent - TSQL cm	CRMIS31	ICicm			0x6F75BD63A95B2F3	0x2251F93F40C0D	PAGEIOLATCH_E	0.0	0	DELETE	IT	17.916	13.704
2018-12-06 03:26:00	Session	113	SQLAgent - TSQL cm	CRMIS31	ICicm			0x8F255A6F345A2ED	0x04D15BD68F8A9	PAGEIOLATCH_E	0.0	0	DELETE	IT	576.563	104.442
2018-12-06 03:26:31	Session	113	SQLAgent - TSQL cm	CRMIS31	ICicm			0x8F255A6F345A2ED	0x04D15BD68F8A9	PAGEIOLATCH_E	0.0	0	DELETE	IT	607.183	113.528
2018-12-06 03:27:01	Session	113	SQLAgent - TSQL cm	CRMIS31	ICicm			0x8F255A6F345A2ED	0x04D15BD68F8A9	PAGEIOLATCH_E	0.0	0	DELETE	IT	637.837	122.313

In the Session> Active Session / Tempdb sessions / Log usage history tab the functionality of searching session history to find sessions blocking other sessions has been added. If there were blockades in a given snap, the Blocking sessions id column containing the session identifier blocking the given session is supplemented in such cases. In the latest version, a search blocking session mechanism has been added.

In the event of a blockade, click the "loupe" button that appears in the Blocking session id column, this will cause the table to be automatically scrolled and the row with the session which is a blocker will be highlighted.

Sessions	Tempdb usage		Log usage												
Type	Session id	Program	Nt user name	Host name	Login name	Context info	Query Hash	Plan Hash	Wait type	Wait time [Seconds]	Blocking session id	Command	Database	Elapsed Time [Seconds]	Cpu Time [Seconds]
Session	641	MSCRMw3wp	cm_isinter	CRMIS31 w3wp...	cm_isinter		0xBEACCC76496494C	0xA98B3757E25D3	LCK_M_U	2 600.9	651	UPDATE	MSCRM	2 600.940	0.015
Session	645	MSCRMw3wp	cm_isinter	CRMIS31 w3wp...	cm_isinter		0x3D3F535832D885	0x37066871CD2B2	LCK_M_U	1 965.1	651	UPDATE	MSCRM	1 965.123	0
Session	650	MSCRMw3wp	cm_isinter	CRMIS32 w3wp...	cm_isinter		0x428E4F71C0AC5B	0x4EB00C8E254A8	LCK_M_U	688.9	651	UPDATE	MSCRM	688.860	0
Session	651	SQLAgent - TSQL	crm	CRMIS0L31	ICicm		0xF8B08BF700B3965	0xC1B93FE89638	LCK_M_U_CXPA...	3 466.0		UPDATE	IT	3 472.513	7.051
Session	653	MSCRMw3wp	cm_isinter	CRMIS31 w3wp...	cm_isinter				LCK_M_U	2 055.9	691	EXECUTE	MSCRM	2 055.910	0
Session	655	MSCRMw3wp	cm_isinter	CRMIS31 w3wp...	cm_isinter				LCK_M_U	1 635.0	562	EXECUTE	MSCRM	1 634.953	0
Session	658	MSCRMw3wp	cm_isinter	CRMIS31 w3wp...	cm_isinter		0xD6B1D04D381DFD53	0x753AC7D7F3E89	LCK_M_X	1 425.2	684	INSERT	MSCRM	1 425.146	0
Session	659	MSCRMw3wp	cm_isinter	CRMIS32 w3wp...	cm_isinter		0x034C248421F42D8C	0x37066871CD2B2	LCK_M_U	2 926.2	651	UPDATE	MSCRM	2 926.150	0
Session	660	Microsoft SQL S...	crm	CRMIS31	cm_isinter		0x52B9331F8E392E2	0x2ADE3782128F3	ASYNC_NETWORK	0.2		SELECT	IT	9 800.136	3.365

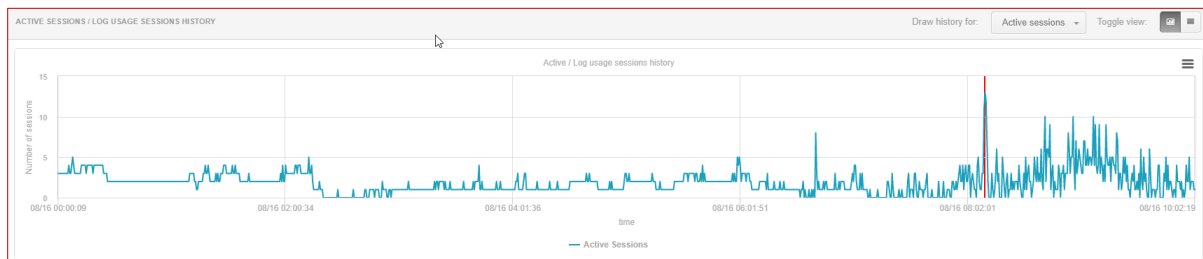
This screen also provides functionality for generating the graph, which can be cycled through by changing the view "Switch to chart".



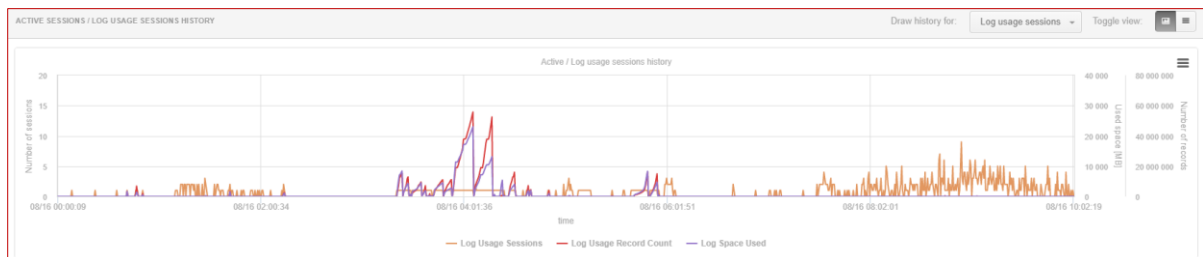
Four charts are available, these are:

- Active sessions,
- Tempdb usage sessions,
- Log usage sessions,
- All statistics.

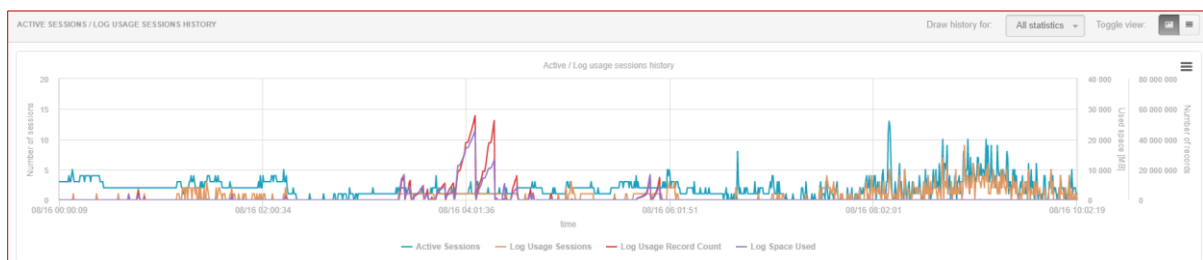
Active Sessions chart:



Log usage Sessions chart:



All statistics chart:



6.2.8 Jobs Menu

Information on Jobs is available in two tabs:

- Current Jobs (containing a list of all Jobs saved to the repository),

Job ID	Name	Description	Enabled	Dropped
53894385-38c8-4595-948b-cde6e0727889	nyspolicy_purge_history	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16e97197-4acc-4149-a444-49cc9d57e1d0	Backup full Subplan_1	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
26c3c2e5-e529-4d8f-8a63-048c48dd19	Export telefonów -> CSOL01	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
e4867267-8175-487c-b064-0935d6f0de	Backup diff Subplan_1	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
0a672134-ef15-4a66-8a67-1305037836d2	akt_jalagay_rec	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
c58e530f-a1a9-4b5e-93be-1e10950d2e3	Zadanie eksportu dostawców	Zadanie synchronizacji na prośbę Marcina Lukarskiego, eksport co 5 minut.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ba5e473f-8c17-485a-922b-35a9a32da1ca	akt_isi_isi	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2a7f4c6a-476a-4796-a245-3b747952aa0	SSIS_export -> CSOL01	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
0e40532-dae9-4796-a245-3b747952aa0	akt_wiz	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9e50d0c7-7976-448a-a1b1-3ba209478955	akt_sp_baza	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5ba30da5-6740-403f-894e-621b0a041104	indeksowanie Subplan_1	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

On the screen for the User a list of jobs downloaded from the repository is presented. The list can be freely filtered based on:

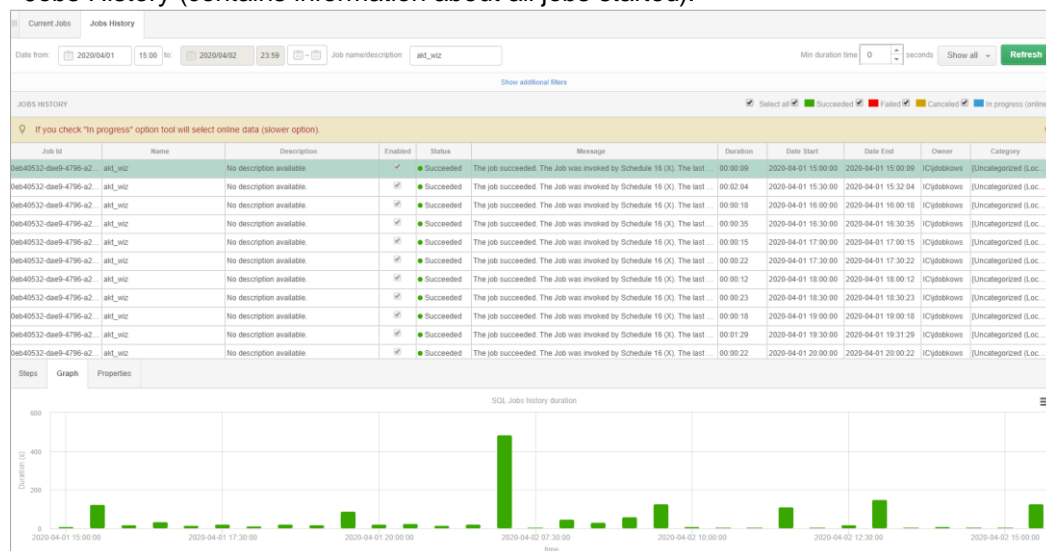
- Job names,
- Job description,
- Owner
- Category,
- Accessibility (Enabled / disabled).

Job removed from the database are still available in the repository and can be viewed by checking the checkbox: Show dropped jobs.

After clicking on a row in the table, the user will be presented with information about the job. The current configuration as well as the commissioning schedule will be displayed. In addition, the [+] button is presented in the name column for each row, after which you can add a given job to the clipboard: Add to jobs list (for later analysis) or go directly to the startup history: View jobs details.

Job ID	Name	Description	Enabled	Dropped
115a6e02-d119-4752-9c3f-8390cd09430	akt_rotacja	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16e97197-4acc-4149-a444-49cc9d57e1d0	Backup full Subplan_1	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1da52362-5825-49e8-8263-0475d4e67681	load_tak_bon_otzrymane	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
27f1158e-bc7f-42b8-aa45-3fca66e267f	akt_File	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2a7000ee-60d4-487f-a5e2-7de948796911	akt_zs_pla_isk_rzb	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3a08377c-3476-48f0-95a8-ed88a20a514	exp_ora	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
405498f3-207c-485c-9151-ac5188a5551	akt_sp_niedziela	No description available.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

- Jobs History (contains information about all jobs started).



The screen contains information about the run history for each job in the analyzed SQL instance. The table lists of jobs performed at a given time. In addition, the User can filter the list as on the previous tab using:

- Job names,
- Job description,
- Owner
- Category,
- Duration,
- Status,
- Accessibility (Enabled / disabled).

By selecting a row in the table, the User receives information on the entire process with a breakdown into individual steps (Steps tab) along with the status for each of the steps separately. The Graph tab presents all runs of the selected job in the selected time range. The Properties tab presents the configuration (steps and schedule) based on selected job.

By selecting the options: In Progress (online), we also get the opportunity to check the status of currently performed jobs on the SQL instance.

6.2.9 Backups Menu

The data in the [**Backups**] tab allow for backup performance checks, i.e. the execution time and backup histories. It is divided into three parts:

- **Statistics** – general information about backups made for each of the databases
- **Overview** – presentation of recent backups made for individual databases
- **History** – history of backups

Statistics

Overview

History

All databases

Backup type

Duration above [s]

Refresh

BACKUPS

Instance type	Instance	Database	Backup start date	Duration [seconds]	Type	Recovery model	Sql Agent status
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_W	adv_works	2015-02-27 09:29:09	215	Full	SIMPLE	Stopped
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_W	master	2015-03-08 00:00:04	0	Full	SIMPLE	Stopped
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_W	model	2015-03-08 00:00:05	0	Full	SIMPLE	Stopped
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_W	msdb	2015-03-08 00:00:05	2	Full	SIMPLE	Stopped
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_W	test_db		0	Differential	SIMPLE	Stopped

6.2.10 Locks Menu

The page contains information about blockades occurring in a given data base. The lock module consists of the following tabs:

- Online Locks – allow the current blocking analysis on the SQL Instance
- Locks history – allow to track blockades in time
- Online Locked Objects - show a list of objects on which locks are currently locked

6.2.10.1 Online Locks Tab

Online Locks tab consists of the following area:

- List of locked sessions section:
 - Tree of blocked sessions refreshed after clicking on the fragment / given point of the chart
 - at the top of the tree, blocking sessions are shown
 - in nodes below, waiting sessions blocked by sessions in the parent node
 - Details for the selected session
 - Text of the query
 - Session parameters, including transaction opening time, transaction type, etc.

An example lock screen is presented below:

The screenshot shows the 'Online Locks' tab in the DBPLUS interface. At the top, there are tabs for 'Online Locks', 'Locks History', and 'Online Locked Objects'. Below the tabs, there's a 'Database' dropdown set to 'All databases' and a 'Refresh' button. A 'List of locked sessions' section contains a table with two sessions. Session 274 is selected, and its details are shown below. The details include the SQL statement for the session, which is an INSERT INTO statement. Below the SQL statement, there's a 'SESSION DETAILS' table with various fields like Session Id, Is blocker for others, Transaction Isolation Level, Transaction Type, Transaction State, Transaction start time, User name, Command, Status, Last Request Date, and Last Request Runtime.

SESSION DETAILS	
Session Id	274
Is blocker for others	Yes
Transaction Isolation Level	Repeatable
Transaction Type	Read/write transaction
Transaction State	The transaction is active
Transaction start time	2018-11-16 16:37:35
User name	CEInavclient
Command	INSERT
Status	running
Last Request Date	2018-11-16 16:37:49
Last Request Runtime	10

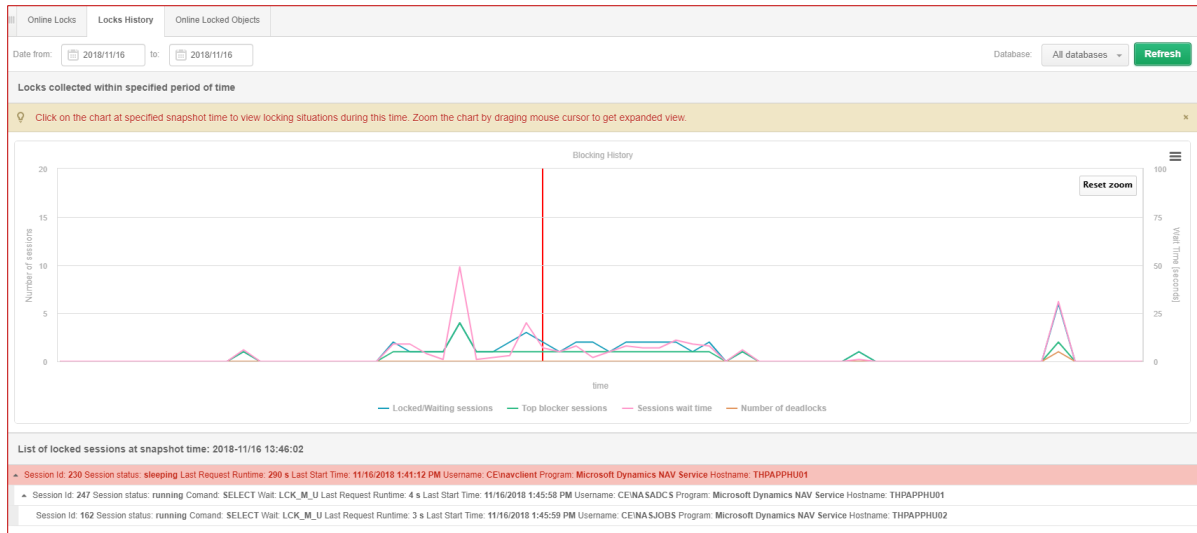
6.2.10.2 Locks history Tab

The page contains information about the history of blockades occurring SQL Instance. The screen consists of the following areas:

- The filter bar over the date range
- A graph showing the locks in time
- Tree of blocked sessions refreshed after clicking on the fragment / given point of the chart
 - at the top of the tree, blocking sessions are shown
 - in nodes below, waiting sessions blocked by sessions in the parent node
- Details for the selected session
 - Text of the query
 - Session parameters, including transaction opening time, transaction type, etc.

It's possible to search for information for any period by selecting an interesting date range (by default, the page opens with the current date set).

An example lock screen is presented below:



6.2.11 Parameters Menu

The page allows for viewing and report changes in SQL Instance parameters over time. The window presents the current status of parameters and their changes over time.

- **Instance Parameters** – instance parameters set using the command **sp_configure**
- **Instance Properties** – other server properties
- **Database Parameters** – databases parameters

Below are the example screens: Status of parameters containing the word “max”.

Server Configuration Parameters Overview

Server Configuration Parameters History

Param name

max

Param value

Refresh

PARAMETERS LISTS

Instance type	Instance	Param name	Param value
PRODUCTION SERVER	WIN-PV/M04LTCT8A/DPD_WAREHOUSE	ft crawl bandwidth (max)	100
PRODUCTION SERVER	WIN-PV/M04LTCT8A/DPD_WAREHOUSE	ft notify bandwidth (max)	100
PRODUCTION SERVER	WIN-PV/M04LTCT8A/DPD_WAREHOUSE	max degree of parallelism	0
PRODUCTION SERVER	WIN-PV/M04LTCT8A/DPD_WAREHOUSE	max full-text crawl range	4
PRODUCTION SERVER	WIN-PV/M04LTCT8A/DPD_WAREHOUSE	max server memory (MB)	2147483647
PRODUCTION SERVER	WIN-PV/M04LTCT8A/DPD_WAREHOUSE	max text repl size (B)	65536
PRODUCTION SERVER	WIN-PV/M04LTCT8A/DPD_WAREHOUSE	max worker threads	0

History of parameter changes:

Server Configuration Parameters Overview

Server Configuration Parameters History

Date from:

2015/03/15

to:

2018/05/08

Param name

Param value

Refresh

PARAMETERS LISTS

Instance type	Instance	Param name	Param value	Date change ▾
PRODUCTION SERVER	WIN-PV\M04LTCT8A\PD_WAREHOUSE	Agent XPs	0	2015-03-16 22:32:49
PRODUCTION SERVER	WIN-PV\M04LTCT8A\PD_WAREHOUSE	max server memory (MB)	2147483647	2015-03-16 22:32:49
PRODUCTION SERVER	WIN-PV\M04LTCT8A\PD_WAREHOUSE	scan for startup procs	0	2015-03-16 22:32:49
PRODUCTION SERVER	WIN-PV\M04LTCT8A\PD_WAREHOUSE	user instance timeout	60	2015-03-16 22:32:49
PRODUCTION SERVER	WIN-PV\M04LTCT8A\PD_WAREHOUSE	user instances enabled	1	2015-03-16 22:32:49

IMPORTANT: The parameter module is also available from the main menu level after exiting the Instance Analysis performance module (go through clicking [Back to dashboard]). Then the system allows analyzing parameters for all monitored databases simultaneously.

6.2.12 Logs Menu

The Logs module allows the user to check logs from the operation of the SQL Instances monitoring procedure.

6.2.12.1 DBPLUS procedure statistics Tab

In the tab, the user can check if any errors occurred while monitoring the specified database. Additionally, the duration of the monitoring procedure is shown - the number of seconds for 15 minutes.

The presentation of information on the times of collecting data from monitored databases by the DBPLUSCATCHER Windows service has been more detailed. This information relates to the procedure for monitoring the database at 15-minute intervals.

The data, as before, is available in the Logs tab at the detail level of the given database. From this version, by clicking on a row in the Snaps table runtime procedure, the User will receive detailed information on the next steps that make up the monitoring procedure.

The screenshot shows the 'Snap procedure run time' table with columns: Date, Work time, and Status. A row for '2020-03-29 18:06:16' is highlighted. To the right, the 'Snap details at 2020-03-29 18:06:16' table is shown, listing internal procedures with columns: Step, Procedure, Start, End, Duration (Seconds), and Status. The first step, '1 Check last database restart', is highlighted. Below this, there are sections for 'DETAILS FOR INTERNAL PROCEDURE: CHECK LAST DATABASE RESTART' and 'ERROR LOGS FOR SELECTED STEP: CHECK LAST DATABASE RESTART'.

Then, by pointing to the step (in the Snap details table), the User receives information about the duration of the procedure and the number of rows processed (information available only for certain steps).

The screenshot shows the 'Snap details at 2019-12-23 15:39:09' table. The 'INTERNAL PROCEDURES RUN TIME' section lists steps 1 through 11. Step 2, '2 Waits events statistics', is highlighted. Below this, the 'DETAILS FOR INTERNAL PROCEDURE: WAITS EVENTS STATISTICS' section is shown, listing statistics with columns: Type, Counter value, Start, End, and Timer Duration. The 'Read data' row is highlighted, showing a counter value of 58 and a timer duration of 0.437.

Information about the status of a given snap is contained in the Status column. If the monitoring process run correctly, a green dot will be displayed in the column.

If one of the monitoring procedure steps has not been performed or has been interrupted and the step concerned is not critical, the User receives information about the reason for the interruption of the step and the status of the entire snap is presented in orange.

Procedure statistics		Procedure Errors	
Date from:	2019/12/23	to:	2019/12/23
DBPLUS PROCEDURE STATISTICS			
Snap procedure run time		Snap details at 2019-12-23 14:06:23	
Date	Work time (seconds)	Status	
2019-12-23 16:08:03	0	● running	
2019-12-23 15:52:49	1	●	
2019-12-23 15:37:36	1	●	
2019-12-23 15:22:23	4	●	
2019-12-23 15:07:09	2	●	
2019-12-23 14:51:56	1	●	
2019-12-23 14:36:43	1	●	
2019-12-23 14:21:29	4	●	
2019-12-23 14:06:23	503	●	
2019-12-23 13:51:10	1	●	
2019-12-23 13:35:56	1	●	
2019-12-23 13:20:43	4	●	
2019-12-23 13:05:30	1	●	
2019-12-23 12:50:16	1	●	
2019-12-23 12:35:03	1	●	
2019-12-23 12:19:49	4	●	
2019-12-23 12:04:42	404	●	
2019-12-23 11:49:29	1	●	
2019-12-23 11:34:16	1	●	
2019-12-23 11:19:02	4	●	
Average time	55	-	
Min time	0	-	

INTERNAL PROCEDURE'S RUN TIME		Start	End	Duration (seconds)	Status
Step	Procedure				
1	Check last database restart	2019-12-23 14:06:23	2019-12-23 14:06:23	0	●
2	Wait events statistics	2019-12-23 14:06:23	2019-12-23 14:06:23	0.047	●
3	Latches statistics	2019-12-23 14:06:23	2019-12-23 14:06:23	0.031	●
4	Operating system information	2019-12-23 14:06:23	2019-12-23 14:06:23	0.016	●
5	Query statistics (queries, procedures) including sql text and plans	2019-12-23 14:06:23	2019-12-23 14:06:23	0.671	●
6	Database size (total, used, free space)	2019-12-23 14:06:23	2019-12-23 14:14:45	501.122	●
7	I/O operation statistics	2019-12-23 14:14:45	2019-12-23 14:14:45	0.281	●
8	Memory informations (SGA including shared pool, db cache size)	2019-12-23 14:14:45	2019-12-23 14:14:45	0.125	●
9	Merge Query statistics to day view	2019-12-23 14:14:45	2019-12-23 14:14:45	0.078	●
10	Merge I/O operations to day view	2019-12-23 14:14:45	2019-12-23 14:14:45	0.031	●
11	Parameters informations	2019-12-23 14:14:45	2019-12-23 14:14:45	0.062	●

DETAILS FOR INTERNAL PROCEDURE: DATABASE SIZE (TOTAL, USED, FREE SPACE)		Type	Counter value	Start	End	Timer Duration (seconds)
Read data		Timer		2019-12-23 14:06:23		0
Write data		Timer				0
Rows processed		Counter	0			

ERROR LOGS FOR SELECTED STEP: DATABASE SIZE (TOTAL, USED, FREE SPACE)		Date	Log message
		2019-12-23 14:14:45	Error reported in following program: StandardSnap: CatchODBSize. Execution for query SELECT /*+ ALL_ROWS */ file_id, mv(Sum(bytes),0) bytes FROM DBA_free_space GROUP BY file_id timeout-ed at DBPLU...

If there was a problem with the connection at the time of the monitoring procedure or the problem concerned a critical step for a given procedure, the status information is written in red.

DBPLUS PROCEDURE STATISTICS		Snap details at 2019-12-23 16:15:00	
Snap procedure run time		INTERNAL PROCEDURE'S RUN TIME	
Date	Work time (seconds)	Status	
2019-12-23 16:15:00		●	
2019-12-23 16:00:00		●	
2019-12-23 15:45:00		●	
2019-12-23 15:30:00		●	
2019-12-23 15:15:00		●	
2019-12-23 15:00:00		●	
2019-12-23 14:45:00		●	
2019-12-23 14:30:00		●	
2019-12-23 14:15:00		●	
2019-12-23 14:00:00		●	
2019-12-23 13:45:00		●	
2019-12-23 13:30:00		●	
2019-12-23 13:15:00		●	
2019-12-23 13:00:00		●	
2019-12-23 12:45:00		●	
2019-12-23 12:30:00		●	
2019-12-23 12:15:00		●	
2019-12-23 12:00:00		●	
2019-12-23 11:45:00		●	
2019-12-23 11:30:00		●	
Average time	0	-	
Min time	0	-	
Max time	0	-	
Count snaps	65		

INTERNAL PROCEDURE'S RUN TIME		Start	End	Duration (seconds)	Status
Step	Procedure				
1	No any steps executed for specified snapshot			0	●

DETAILS FOR INTERNAL PROCEDURE		Type	Counter value	Start	End	Timer Duration (seconds)
Please select internal procedure						

ERROR LOGS FOR SELECTED SNAPSHOT		Date	Log message
		2019-12-23 16:10:03	Error reported in following program: SessionsUndoLockSort: SnapRunnerLocks.Run. ORA-12541: TNS: No listener at OracleInternal.ConnectionPool.PoolManager'3 Get(ConnectionString coWithDirOrNewPwd, ...
		2019-12-23 16:10:05	Error reported in following program: Dashboard: SnapRunner DashboardSnapQueries. ORA-12541: TNS: No listener at DBPLUS Catcher facade SQLFacadeDashboard.DashboardSnapQueries(Boolean deleteCl...
		2019-12-23 16:10:34	Error reported in following program: SessionsUndoLockSort: SnapRunnerLocks.Run. ORA-12541: TNS: No listener at OracleInternal.ConnectionPool.PoolManager'3 Get(ConnectionString coWithDirOrNewPwd, ...

If the monitoring procedure is in progress, this information is visible in the status (running) field, as well as the Online steps refresh button is visible, after which the information on the monitoring procedure progress is refreshed.

Procedure statistics			Procedure Errors		
Date from: 2019/12/24 to: 2019/12/24			Refresh		
DBPLUS PROCEDURE STATISTICS			Snap details at 2019-12-24 10:28:11 with selected currently executed step		
Snap procedure run time			INTERNAL PROCEDURES RUN TIME		
Date	Work time (seconds)	Status	Step	Procedure	Start
2019-12-24 10:28:11	7	Running	1	Check last database restart	2019-12-24 10:28:11
2019-12-24 10:12:59	36	Completed	2	Wait events statistics	2019-12-24 10:28:11
2019-12-24 09:57:46	72	Completed	3	Latches statistics	2019-12-24 10:28:16
2019-12-24 09:42:34	35	Completed	4	Operating system information	2019-12-24 10:28:17
2019-12-24 09:27:21	34	Completed	5	Query statistics (queries/procedures) including sql text and plans	2019-12-24 10:28:17
2019-12-24 09:12:09	33	Completed			
2019-12-24 08:56:57	61	Completed			
2019-12-24 08:41:42	36	Completed			
2019-12-24 08:26:29	45	Completed			
2019-12-24 08:11:16	32	Completed			
2019-12-24 07:56:03	65	Completed			
2019-12-24 07:40:51	30	Completed			
2019-12-24 07:25:38	34	Completed			
2019-12-24 07:10:25	31	Completed			
2019-12-24 06:55:13	60	Completed			
2019-12-24 06:40:00	29	Completed			
2019-12-24 06:24:47	32	Completed			
2019-12-24 06:09:34	38	Completed			
2019-12-24 05:54:21	45	Completed			
2019-12-24 05:39:09	84	Completed			

Step	Procedure	Start	End	Duration (seconds)	Status
1	Check last database restart	2019-12-24 10:28:11	2019-12-24 10:28:11	0	Completed
2	Wait events statistics	2019-12-24 10:28:11	2019-12-24 10:28:16	5.444	Completed
3	Latches statistics	2019-12-24 10:28:16	2019-12-24 10:28:17	0.234	Completed
4	Operating system information	2019-12-24 10:28:17	2019-12-24 10:28:17	0.047	Completed
5	Query statistics (queries/procedures) including sql text and plans	2019-12-24 10:28:17		0.858	Running

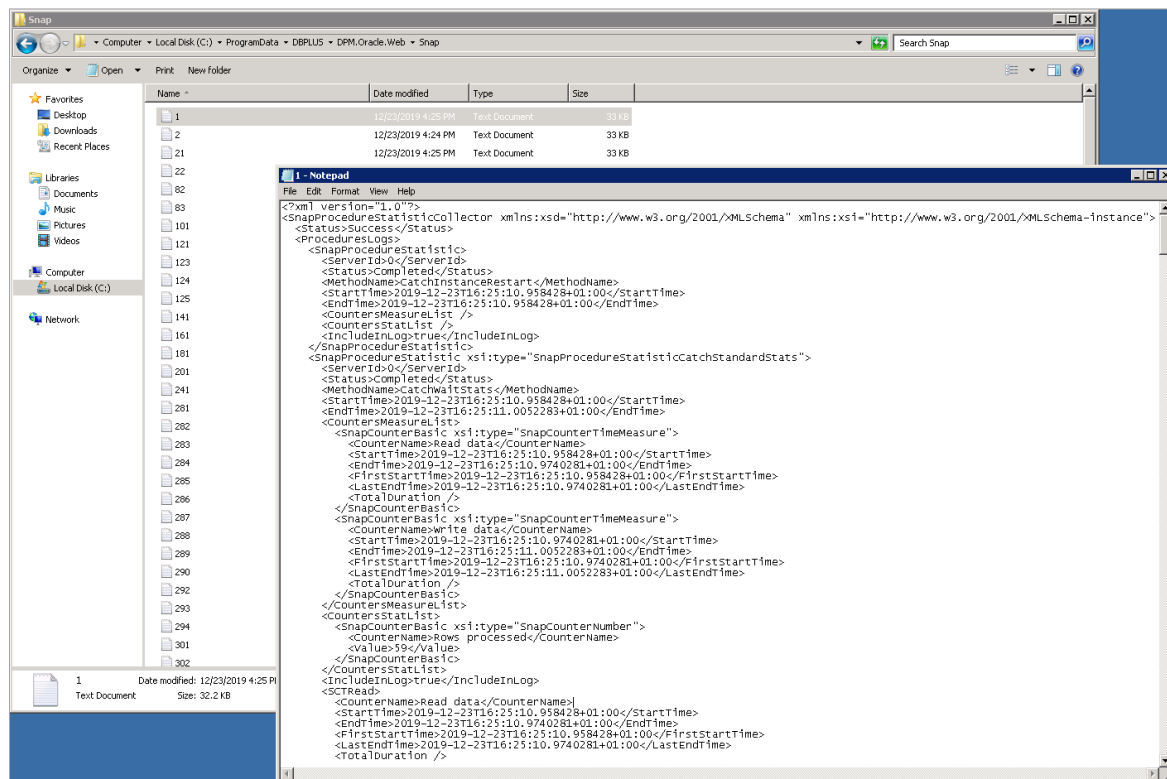
Statistics	Type	Counter value	Start	End	Timer Duration (seconds)
Read data	Timer		2019-12-24 10:28:11	2019-12-24 10:28:16	0
Write data	Timer		2019-12-24 10:28:16	2019-12-24 10:28:16	0
Rows processed	Counter	62			

Date	Log message

In addition, all problems related to the monitoring procedure are available in the form of a list on the Procedure Errors tab.

Information on the monitoring procedure is also included in the form of a file on the application server. The file contains information about the last snap performed on a given database. The file is in the folder: C: \ ProgramData \ DBPLUS \ DPM.Web \ Snap

Each file is marked with a digit assigned to the database when it is included in the monitoring (dbplus_central_servers table in the DBPLUS schema in the repository database).



6.2.12.2 SQL Server Logs Tab

An example screen is shown below:

Dbplus procedure statistics		SQL Server logs
Current log:	0	Type: Error log Text to search: <input type="text"/>
<input type="button" value="Refresh"/>		
SQL SERVER LOGS		
Date	Process Info	Log Message
2018-04-11 03:50:33	Server	Microsoft SQL Server 2012 - 11.0.2100.60 (X64) Feb 10 2012 19:39:15 Copyright (c) Microsoft Corporation Express Edition (64-bit) on Windows NT 6.1 <X64> (Build 7601: Ser
2018-04-11 03:50:33	Server	Server process ID is 3128.
2018-04-11 03:50:33	Server	Authentication mode is MIXED.
2018-04-11 03:50:33	Server	Logging SQL Server messages in file 'C:\Program Files\Microsoft SQL Server\MSSQL11.PD_WAREHOUSE\MSSQL\Log\ERRORLOG'.
2018-04-11 03:50:33	Server	The service account is 'NT Service\MSSQL\$PD_WAREHOUSE'. This is an informational message; no user action is required.
2018-04-11 03:50:33	Server	Registry startup parameters: -d C:\Program Files\Microsoft SQL Server\MSSQL11.PD_WAREHOUSE\MSSQL\DATA\master.mdf -e C:\Program Files\Microsoft SQL Server\MSS
2018-04-11 03:50:33	Server	Command Line Startup Parameters: -s "PD_WAREHOUSE"
2018-04-11 03:50:33	Server	All rights reserved.
2018-04-11 03:50:33	Server	(c) Microsoft Corporation.
2018-04-11 03:50:33	Server	System Manufacturer: 'System manufacturer', System Model: 'System Product Name'.
2018-04-11 03:50:36	Server	Detected 16375 MB of RAM. This is an informational message; no user action is required.
2018-04-11 03:50:36	Server	SQL Server detected 1 sockets with 4 cores per socket and 8 logical processors per socket, 8 total logical processors; using 8 logical processors based on SQL Server licensing
2018-04-11 03:50:36	Server	SQL Server is starting at normal priority base (=7). This is an informational message only. No user action is required.
2018-04-11 03:50:36	Server	Using conventional memory in the memory manager.
2018-04-11 03:50:38	Server	This instance of SQL Server last reported using a process ID of 3156 at 2018-04-11 03:44:01 (local) 2018-04-11 01:44:01 (UTC). This is an informational message only; no user
2018-04-11 03:50:38	Server	Node configuration: node 0: CPU mask: 0x00000000000000ff:0 Active CPU mask: 0x00000000000000ff:0. This message provides a description of the NUMA configuration for t
2018-04-11 03:50:38	Server	Using dynamic lock allocation. Initial allocation of 2500 Lock blocks and 5000 Lock Owner blocks per node. This is an informational message only. No user action is required.

6.2.13 Reports Menu

The following reports are available in the Reports module:

- Performance Report,

6.2.13.1 Performance report

The report presents the performance of the SQL Instances in the selected time period. The report contains information about:

- Top queries operating in the database for:
 - Duration: Elapsed Time
 - Utilization Processors: CPU Time
 - Readings from disk devices
 - Block reads from memory
 - Number of queries
- The duration of blockades in an hourly manner
- Top wait lists
- Top latches

6.3 Space monitor Menu

Space Monitor module allows users to analyze the storage space on servers. The module is divided into two basic groups:

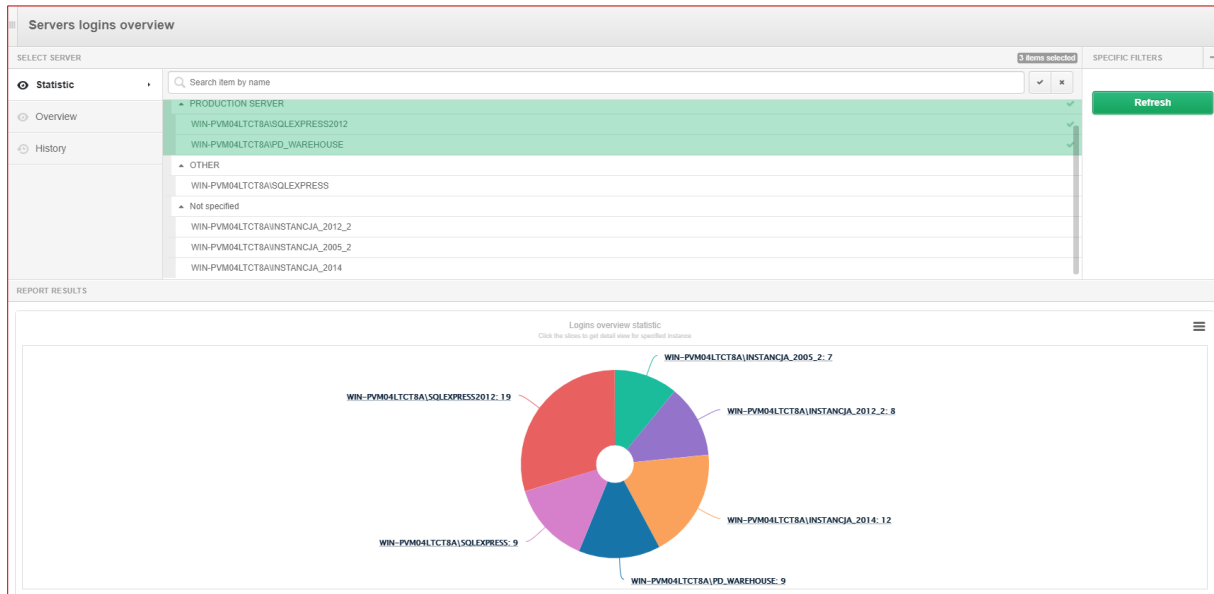
- Harddisk space (contains information on disk space utilization on disks)
- Database space (contains estimates of disk space usage based on increment statistics)

As part of the preview, three options follows:

- Current - verification of the current status of occupancy,
- Overview - presents the occupation of databases for a given period of time in tabular form,
- History - presents the occupation of databases for a given day.

6.4 Accounts Menu

The module contains basic information about users logging into a given SQL instance. The application allows access to information on user accounts with the ability to verify the number of users for a given SQL instance, and information of their current status and type of access.



Information on the type of access, type of access and status available in the tabular form.

Servers logins overview

SELECT SERVER

Statistic

Overview

History

ALL SERVERS

PRODUCTION SERVER

WIN-PVM04LTCT8A/SQLXPRESS2012

WIN-PVM04LTCT8A/PO_WAREHOUSE

OTHER

WIN-PVM04LTCT8A/SQLXPRESS

Not specified

WIN-PVM04LTCT8A/INSTANCJA_2012_2

WIN-PVM04LTCT8A/INSTANCJA_2005_2

3 items selected

SPECIFIC FILTERS

Login type

Login name

Status

Role

Login exists

Refresh

REPORT RESULTS

Instance type	Instance	Login	Default database	Type	Is Admin	Is Disabled	Date creation	Last Date changed	Users count
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLXPRESS	##MS_PolicyEventProcessing	master	SQL_LOGIN	No	Yes	2012-02-10 21:07:46	2014-03-06 08:33:08	2
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLXPRESS	##MS_PolicyTsqExecutionLog	master	SQL_LOGIN	No	Yes	2012-02-10 21:07:46	2014-03-06 08:33:08	1
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLXPRESS	ADDBPLUS\mrodeske	master	WINDOWS_LOGIN	Yes	No	2017-04-20 10:42:27	2017-04-20 10:47:23	0
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLXPRESS	bigos	master	SQL_LOGIN	No	No	2014-01-31 23:03:08	2014-03-06 08:33:08	17

The presented data can also be viewed for a given in history.

6.5 Backups Menu

The data in the **[Backups]** menu allow for backup performance checks, i.e. the execution time and backup histories. It is divided into three parts:

- **Statistics** – general information about backups made for each of the databases
- **Overview** – presentation of recent backups made for individual databases
- **History** – history of backups

The information presented below about historically performed backups on the SQL instance.

Backups overview

SELECT SERVER / DATABASE

59 items selected

PERIOD

SPECIFIC FILTERS

Statistic

Search item by name

✓

✕

Overview

ALL SERVERS

PRODUCTION SERVER

WIN-PVM04LTCT8A\SQLEXPRESS2012

WIN-PVM04LTCT8A\PD_WAREHOUSE

OTHER

WIN-PVM04LTCT8A\SQLEXPRESS

History

Date from

2014/01/14

Date to

2018/11/21

Backup type

Duration above [s]

Refresh

REPORT RESULTS

Instance type	Instance	Database	Backup start date	Duration [Seconds]	Type	Recovery model
PRODUCTION SERVER	WIN-PVM04LTCT8A\SQLEXPRESS	DBPLUS_WEB	2016-12-05 11:27:19	1 604	Full	SIMPLE
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_WAREHOUSE	model	2015-03-08 00:00:05	0	Full	FULL
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_WAREHOUSE	msdb	2015-03-08 00:00:05	2	Full	SIMPLE
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_WAREHOUSE	master	2015-03-08 00:00:04	0	Full	SIMPLE
PRODUCTION SERVER	WIN-PVM04LTCT8A\PD_WAREHOUSE	adv_works	2015-02-27 09:29:09	215	Full	SIMPLE

6.6 Parameters Menu

The page allows for viewing and report changes in all connected to monitoring SQL Instance parameters over time. The window presents the current status of parameters and their changes over time.

- **Instance Parameters** – instance parameters set using the command **sp_configure**
- **Instance Properties** – other server properties
- **Database Parameters** – databases parameters

From this level, it is possible to view information about given parameters simultaneously for all instances connected to monitoring.

The screen below shows information about the "IsReadCommittedSnapshot" parameter for all servers of the "PRODUCTION SERVER" type, broken down by individual databases.

Databases parameters overview

SELECT SERVER / DATABASE

59 items selected

SPECIFIC FILTERS

Overview

History

Search item by name

✓

✕

ALL SERVERS

PRODUCTION SERVER

WIN-PVM04LTCT8A/SQLEXPRESS2012

WIN-PVM04LTCT8A/IPD_WAREHOUSE

OTHER

WIN-PVM04LTCT8A/SQLEXPRESS

REPORT RESULTS

Instance type	Instance	Database	Param name	Param value
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	acs_repair	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	adv_works	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	advworks	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	crm	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	db_test	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	DBPLUS_JC	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	DBPLUS_WEB	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	erp_statistics	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	factory	IsReadCommittedSnapshot	OFF
PRODUCTION SERVER	WIN-PVM04LTCT8A/SQLEXPRESS2012	load	IsReadCommittedSnapshot	OFF

HISTORY FOR SELECTED PARAMETER

Date change	Param value
2017-04-28 15:00:04	OFF

6.7 Reports Menu

In this menu, the application allows you to make reports based on data from monitored servers. Reports concern two groups:

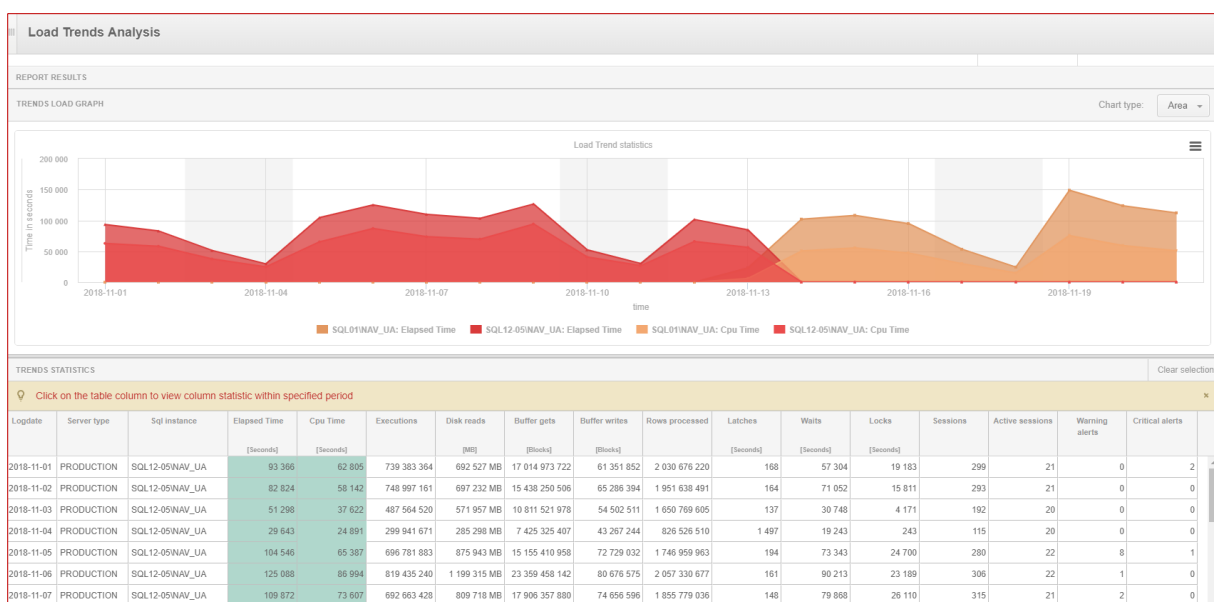
- infrastructure analysis,
- trend analysis.

Below is an example of a report with an analysis of infrastructure. The report contains basic information about the amount of CPU, databases as part of the SQL instance or the space used.

The data is presented in tabular form as well as in the form of graphs.

REPORT RESULTS																	
Overview Analysis																	
SQL INSTANCES																	
Server Type	Server	SQL Server information				Monitoring start date	Virtual CPU Cores			Databases count			Databases space (MB)			Begin	End
		Sql Instance	Version	Installation date			Begin	End	Growth	Begin	End	Growth	Begin	End	Growth		
Not specified	WIN-PVM04LTCT8A	INSTANCIA_2005_2	2005	2014-05-07		2014-05-07	2	2	0	11	11	0	42.5	42.5	0		
Not specified	WIN-PVM04LTCT8A	INSTANCIA_2014	2014	2014-07-25		2014-07-25	8	8	0	11	11	0	11155	11155	0		
Not specified	WIN-PVM04LTCT8A	INSTANCIA_2012_2	2012	2017-02-05		2017-02-05	8	8	0	4	4	0	37.8	37.8	0		
PRODUCTION SERVER	WIN-PVM04LTCT8A	SQLXPRESS2012	2012	2013-06-20		2014-03-06	8	8	0	20	20	0	14603.6	14678.6	-5		
PRODUCTION SERVER	WIN-PVM04LTCT8A	PD_WAREHOUSE	2012	2015-03-16		2014-09-11	8	8	0	6	6	0	26282	26282	0		
Totals							8	8	0	52	52	0	52200.9	52195.9	-5		
DATABASES SPACE																	
SQL Server information		Databases total space at the begin (MB)			Databases total space at the end (MB)			Growth (MB)		Databases used space at the begin (MB)				Databases used space at the end (MB)			
Sql Instance		Total space	Data space	Log space	Total space	Data space	Log space	Total Space		Used space	Data space	Log space	Space free	Used space	Data space	Log space	Space free
WIN-PVM04LTCT8A	INSTANCIA_2005_2	42.5	30.1	12.4	42.5	30.1	12.4	0		27	21.7	5.3	15.5	26.9	21.7	5.2	15.6
WIN-PVM04LTCT8A	INSTANCIA_2012_2	37.8	29.7	8.1	37.8	29.7	8.1	0		23.4	21.2	2.2	14.4	23.5	21.2	2.3	14.3
WIN-PVM04LTCT8A	INSTANCIA_2014	11155	10122.8	1032.2	11155	10122.8	1032.2	0		5769	5682	77	5386	5768.8	5682	76.8	5386.2
WIN-PVM04LTCT8A	PD_WAREHOUSE	26282	9076.7	17205.3	26282	9076.7	17205.3	0		8223.3	8163.8	59.5	10958.7	7090.3	7026.2	64.1	19191.7
WIN-PVM04LTCT8A	SQLXPRESS2012	14603.6	13027	1656.6	14678.6	13022	1656.6	-5		11603.2	11562.1	41.1	3080.4	11655.4	11613	42.4	3023.2
Totals		52200.9	32286.3	19914.6	52195.9	32291.3	19914.6	-5		25645.9	25468.8	185.1	26555	24564.9	24374.1	190.8	27631
HARDDISK SPACE																	
Server information		Harddisk total space (MB)			Harddisk used space (MB)			Harddisk free space (MB)									
Server		Begin	End	Growth	Begin	End	Growth	Begin	End	Growth	Begin	End	Growth	Begin	End	Growth	
WIN-PVM04LTCT8A		476835	476835	0	430187.7	432303.4	2115.7	46647.3	44531.6								
Totals		476835	476835	0	430187.7	432303.4	2115.7	46647.3	44531.6								

The other report type is "Load trends" report concerns the values of the main statistics calculated for each SQL instance. As part of this report, we can compare statistics for multiple instances at the same time.



6.8 Servers Monitor Menu

Information on the performance of the DBPLUS Performance Monitor is available on the website. Two submenus available from there:

- Application architecture
- SQL Server architecture
- Logs

6.8.1 Application architecture

The module contains information on the status of monitored SQL instances. For each of the monitored instances, information about the date of the last collected snapshot and the last action (operations from the level of the DBPLUS Application towards the database) is available.

In the middle part of page **Monitoring service section**, information about the status of the DBPLUSCATCHERSERVICE monitoring service is also available. Is also collected information about the amount of memory used and the CPU utilization of the server on which the DBPLUS Performance Monitor application is installed.

The Dbplus Performance Monitor module consists information about which instance is the monitoring system repository.

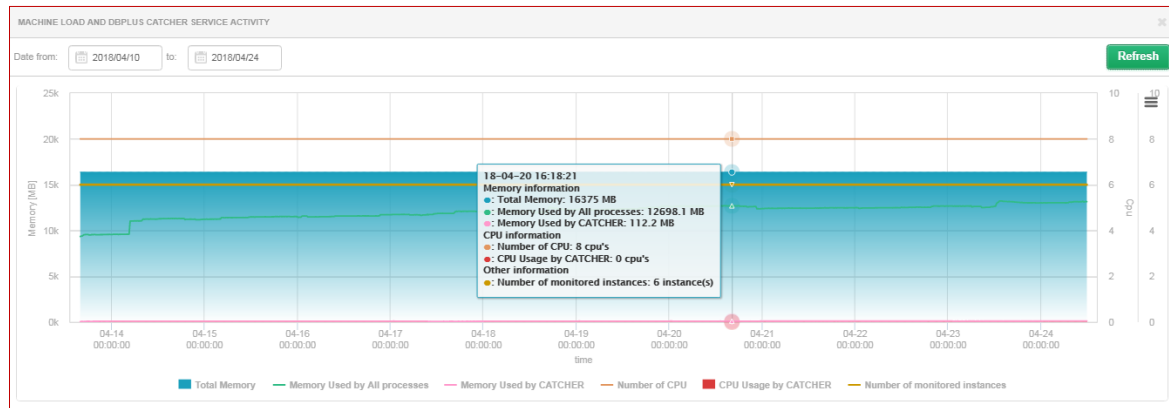
An example screen is shown below:

Application architecture

List of monitored sql instances			Monitoring service	Dbplus Performance Monitor
Instance name	Last snapshot date	Last activity		
WIN-PV\M04LCT8A\INSTANCJA_2005_2	2018-04-24 11:53:03	2018-04-24 12:01:15	DBPLUS Catcher Service status: ● Running Last service activity: 2018-04-24 12:01:32 Machine Total Memory: 16375 MB Machine Memory Usage: 13165 MB DBPLUSCATCHER Memory Usage: 118 MB DBPLUSCATCHER CPU Usage: 0 % View service activity	Repository Information Sql Instance: Jsglexpress2012 Database: DBPLUS_WEB
WIN-PV\M04LCT8A\INSTANCJA_2012_2	2018-04-24 11:52:48	2018-04-24 12:01:30		
WIN-PV\M04LCT8A\INSTANCJA_2014	2018-04-24 11:52:37	2018-04-24 12:01:30		
WIN-PV\M04LCT8A\IPD_WAREHOUSE	2018-04-24 11:52:58	2018-04-24 12:01:30		
WIN-PV\M04LCT8A\SQLSERVER	2018-04-24 11:52:58	2018-04-24 12:01:15		
WIN-PV\M04LCT8A\SQLSERVER2012	2018-04-24 11:52:05	2018-04-24 12:01:15		
dbplusem1 (Excluded from monitoring)				
WIN-PV\M04LCT8A\EVAL_ORNG (Excluded from monitoring)				
WIN-PV\M04LCT8A\INSTANCJA_2012_1 (Excluded from monitoring)				
WIN-PV\M04LCT8A\INSTANCJA_2012_2 (Excluded from monitoring)				
WIN-PV\M04LCT8A\SQLSERVER2008 (Excluded from monitoring)				

● SQL Instance is monitored
 ● SQL Instance is disabled (to enable please go to Configuration module)
 ● SQL Instance is not available or DBPLUS Catcher service is not running

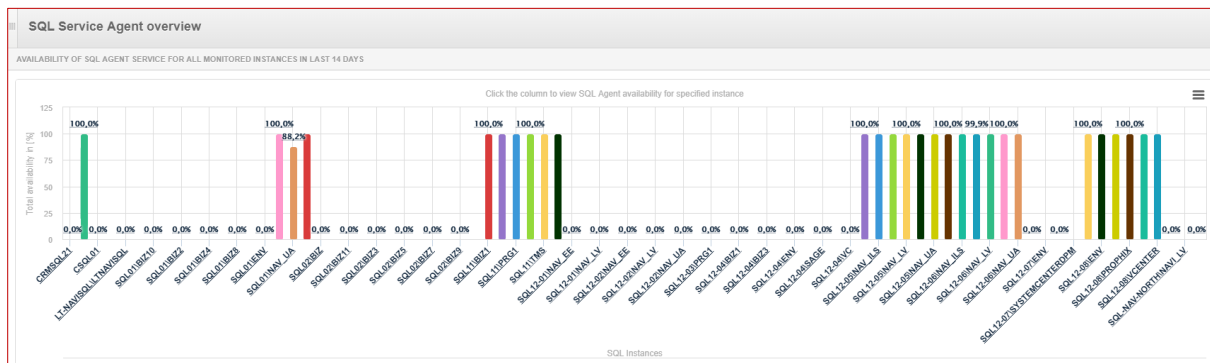
In the following slide, the history of the DBPLUSCATCHER service activity, after clicking the **[View service activity]** button:



6.8.2 SQL Server Agent

As part of this page, the user has access to information on the reliability / availability of the SQL Agent site for each SQL instance, presented in the form of a graph.

A score below 100% means a break in the operation of SQL Agent at the turn of the last 14 days.



6.8.3 Logs

The module contains information about any irregularities or errors related to the monitoring activity. Information is available about the name of SQL instances on which the problem was detected as well as the date and content of the reported error.

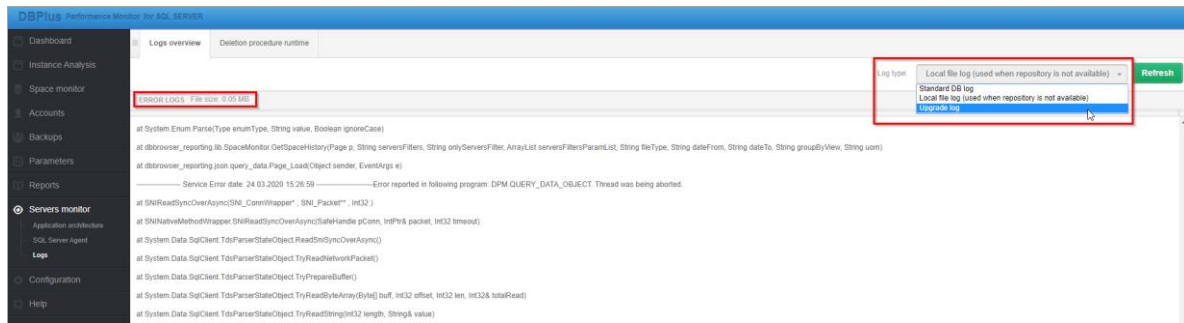
Logs overview tab

After entering the Logs overview tab, the User will be presented with default logs saved in the Repository database (Standard DB Log), that contain information about problems with possible monitoring.

The User will also have the option of displaying information available in the logs available locally on the application server (Local file log). Information about problems is saved there when it is not possible to save this information in the repository database.

The next log concerns information related to the application update process. This file is created during the application upgrade process (downloading the new version). We save information about changes made to the data model as well as the update process.

In addition, information about the size of the file is displayed for each file.



Deleting historical data process

Depending on the KEEP_SNAPSHOT_HISTORY_DAYS parameter (Performance in the Settings menu), the Performance Monitor application deletes historical data - older than the number of days in the parameter. Details (collected every 15 minutes) are deleted. Aggregated data for the whole day is never deleted and is stored throughout the monitoring period.

The process of deleting detailed data is also monitored. The process runs every snap (15 minutes). Details about the process are available in the main menu Servers monitor> Logs in the Deletion tab runtime procedure.

The page contains information about the total duration of the entire procedure, and about the time of individual components and the number of deleted records from dedicated tables from the DBPLUS schema in the repository database.

Logs overview

Deletion procedure runtime

Date from:2019/12/23to:2019/12/24

Refresh

DELETION PROCEDURE RUNTIME

Procedure run time

Date	Work time [seconds]	Status
2019-12-24 11:42:39	14	
2019-12-24 11:27:26	13	
2019-12-24 11:12:13	21	
2019-12-24 10:57:00	15	
2019-12-24 10:41:47	18	
2019-12-24 10:26:34	17	
2019-12-24 10:11:21	15	
2019-12-24 09:56:08	16	
2019-12-24 09:40:55	13	
2019-12-24 09:25:42	14	
2019-12-24 09:10:29	17	
2019-12-24 08:55:13	11	
2019-12-24 08:40:00	13	
2019-12-24 08:24:46	16	
2019-12-24 08:09:34	18	
2019-12-24 07:54:21	15	

Snap deletion details at 2019-12-24 11:12:13

INTERNAL STEPS STATISTICS		Start	End	Duration [seconds]	Deleted rows	Status
Step	Procedure					
1	DeletionSettings	2019-12-24 11:12:13	2019-12-24 11:12:14	0.468		
2	Delete dbplus_errlog	2019-12-24 11:12:14	2019-12-24 11:12:14	0	0	
3	Delete dbplus_alert_mails	2019-12-24 11:12:14	2019-12-24 11:12:14	0	2	
4	Delete dbplus_tab4_log	2019-12-24 11:12:14	2019-12-24 11:12:14	0.016	0	
5	Delete dbplus_tab_catcher	2019-12-24 11:12:14	2019-12-24 11:12:14	0	1	
6	Delete snap table: dbplus_tab4	2019-12-24 11:12:14	2019-12-24 11:12:16	2.371	5 000	
7	Delete snap table: dbplus_tab2	2019-12-24 11:12:16	2019-12-24 11:12:16	0.156	4 144	
8	Delete snap table: dbplus_tab17	2019-12-24 11:12:16	2019-12-24 11:12:16	0.047	2 609	
9	Delete snap table: dbplus_tab16	2019-12-24 11:12:16	2019-12-24 11:12:17	0.577	27 078	
10	Delete snap table: dbplus_tab19	2019-12-24 11:12:17	2019-12-24 11:12:18	1.357	9 637	
11	Delete snap table: dbplus_tab16	2019-12-24 11:12:18	2019-12-24 11:12:18	0.016	0	
12	Delete snap table: dbplus_tab18_rowc	2019-12-24 11:12:18	2019-12-24 11:12:18	0.109	2 471	
13	Delete snap table: dbplus_tab4_inspect	2019-12-24 11:12:18	2019-12-24 11:12:18	0	0	
14	Delete snap table: dbplus_tab_reason_log	2019-12-24 11:12:18	2019-12-24 11:12:18	0.047	918	
15	Delete snap table: dbplus_snaps_tab1	2019-12-24 11:12:18	2019-12-24 11:12:18	0	0	

6.9 Configuration Menu

In this menu the application provides the possibility to modify the configuration regarding the performance of the DBPLUS Performance Monitor. Several submenus are available:

- Settings,
- Servers,
- Reference lists,
- Security,
- Alert settings.

6.9.1 Settings

This submenu allows to modify the parameters controlling the operation of the application. The Settings tab presents the basic configuration parameters. Some parameters are set globally for all monitored data bases.

Depending on the quality of queries and the type of problems in the system, following options can be enabled:

- **KEEP_SNAPSHOT_HISTORY_DAYS** - number of storage days for the retail history of SQL Instances performance,
- **LOCKING_HISTORY_DAYS**- number of days how long to keep locking statistics history
- **LOCKING_SNAPSHOT_FREQUENCY** - changing the frequency of collecting block history,
- **MONITOR_DDL_STATEMENTS** – process to collect load for DLL statements,
- **and many others**

Settings	Dashboard Icon	Dashboard TV Parameters	
			DATABASES_CATEGORIZATION_PLUGIN to on value, otherwise to off
KEEP_SNAPSHOT_HISTORY_DAYS	14	Number of days how long to keep detail statistics for sql statement executions, waits, latches, performance counters.	Edit
LOCKING_HISTORY_DAYS	60	Number of days how long to keep locking statistics history.	Edit
LOCKING_SNAPSHOT_FREQUENCY	15	The interval time in seconds between each snapshot of locks made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.	Edit
MONITOR_DDL_STATEMENTS	OFF	Enable or disable process to collect load for DDL statements like ALTER,CREATE,DROP,DBCC,BACKUP,RESTORE. Utilization of such statements is calculated based on running sessions due to lack of any statistic information in sys.dm_exec_query_stats view.	Edit
MONITOR_TRACEWRITE_STATEMENTS	OFF	Enable or disable in monitoring process TRACEWRITE statements ran from Sql Profiler tool. It's recommended to disable such type of statements to have better visibility on the Instance Load for bussiness sql queries.	Edit
MONITOR_WAITFOR_STATEMENTS	OFF	Enable or disable in monitoring process WAITFOR/RECEIVE statements usually ran in sql server queues. It's recommended to disable such type of statements to have better visibility on the Instance Load for bussiness sql queries.	Edit
SECURITY	OFF	Application can work in SECURITY mode set to ON or to OFF. It means that application uses (or doesnt use) user authentication. Setting the SECURITY to on, it requires at least one user created.	Edit

IMPORTANT: To change the configuration for a dedicated SQL Instance, select the instances at the bottom of the page and make changes by clicking the **[Edit]** button.

INSTANCE PARAMETERS - PLEASE SELECT A SERVER			
WIN-PVM04LTCT8AIPD_WAREHOUSE (1 param/s overwritten)			
Parameter	Value	Description	
CURSOR_SNAPSHOT_FREQUENCY	6	The interval time in seconds between each snapshot of FETCH API open cursor statements, made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.	Edit
LOCKING_SNAPSHOT_FREQUENCY	15	The interval time in seconds between each snapshot of locks made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.	Edit Restore
MONITOR_DDL_STATEMENTS	OFF	Enable or disable process to collect load for DDL statements like ALTER,CREATE,DROP,DBCC,BACKUP,RESTORE. Utilization of such statements is calculated based on running sessions due to lack of any statistic information in sys.dm_exec_query_stats view.	Edit
MONITOR_TRACEWRITE_STATEMENTS	OFF	Enable or disable in monitoring process TRACEWRITE statements ran from Sql Profiler tool. It's recommended to disable such type of statements to have better visibility on the Instance Load for business sql queries.	Edit
MONITOR_WAITFOR_STATEMENTS	OFF	Enable or disable in monitoring process WAITFOR/RECEIVE statements usually ran in sql server queues. It's recommended to disable such type of statements to have better visibility on the Instance Load for business sql queries.	Edit

Information collection interrupted queries is controlled by parameters. These parameters are available in the main menu Configuration> Settings and they are:

- CANCELED SNAPSHOT FREQUENCY – the time interval between snapshots that retrieve information about broken queries. The shorter the time, the more accurate the monitoring of interrupted queries,
- MONITOR_CANCELED_LOCKING – queries interrupted due to blockage,
- MONITOR_CANCELED_QUERIES – queries interrupted by the user.

Only monitoring of queries interrupted due to blocking is enabled (MONITOR_CANCELED_LOCKING parameter set to ON) by default. To enable monitoring of user interrupted queries, manually change the MONITOR_CANCELED_QUERIES parameter to ON (default OFF).

Dashboard	Settings	Waits settings	Dashboard icon	Dashboard Tv Parameters
Instance Analysis	APPLICATION PARAMETERS			
Space monitor	Parameter	Value	Description	
Accounts	CANCELED_SNAPSHOT_FREQUENCY	10	The interval time in seconds between each snapshot of checking active sessions with canceled statements/events. The parameter can be setup separately for each instance.	Edit
Backups	MONITOR_CANCELED_LOCKING	ON	Enable or disable logic to collect load for canceled locks, locking timeout or deadlock. Utilization of such operations is gathered using extended event feature	Edit
Parameters	MONITOR_CANCELED_QUERIES	OFF	Enable or disable logic to collect load for canceled statements, client-interrupt requests, timeouted due to deadlock. Utilization of such operations is gathered using extended event feature	Edit
Reports	DATABASE_CATEGORIZATION_PLUGIN	OFF	In DPM tool each database can have 3 different categories: database type, system category, vendor description. The list of categories and database assignment to those categories can be managed in Configuration module (pages: References lists, Servers). Categorization is useful for environments with large number of sql instances and databases. It's used for filterselection purposes and such categorizations can be included in the reports. To use this functionality, please set parameter DATABASES_CATEGORIZATION_PLUGIN to on value, otherwise to off	Edit
Servers monitor				
Configuration				
Settings				
Current				

6.9.2 Servers

This page allows to configure which sql instances should be monitored and set the SQL instance type. The correct type setting for each sql instances allows the user to use this grouping in various functions of the DBPLUS Performance Monitor application, such as Space Monitor, when present the size of instances assigned to a given group.

On the website it is also possible to set, among others:

- database visibility in monitoring
- which databases are available for a given instance?
- additional information about connect the instance to the DBPLUS Performance Monitor application

View servers and connections				
SERVER LIST			DETAILS FOR SELECTED SERVER	
Search servers by name			Basic Connection prop Databases list	
Server	Instance name	Type	Enabled	
dbplusrm1	dbplusrm1	Not Specified	<input type="checkbox"/>	Connection name: 2005_2
serverazure	serverazure	Not Specified	<input type="checkbox"/>	Server name: WIN-PVM04LTCT8AINSTANCE_200
WIN-PVM04LTCT8A	WIN-PVM04LTCT8AIEVAL_ORNG	Not Specified	<input type="checkbox"/>	Type: Not specified
WIN-PVM04LTCT8A	WIN-PVM04LTCT8AINSTANCE_2005_2	Not Specified	<input checked="" type="checkbox"/>	Enabled: Yes
WIN-PVM04LTCT8A	WIN-PVM04LTCT8AINSTANCE_2012_1	Not Specified	<input type="checkbox"/>	
WIN-PVM04LTCT8A	WIN-PVM04LTCT8AINSTANCE_2012_2	Not Specified	<input checked="" type="checkbox"/>	
WIN-PVM04LTCT8A	WIN-PVM04LTCT8AINSTANCE_2012_2	Not Specified	<input type="checkbox"/>	

6.9.3 Reference lists

This tab contains the system dictionaries used in the application. Existing dictionary data can be freely modified.

6.9.4 Security

This tab provides the option of setting access for a user, group of users or profiles. Access is granted at the SQL Instance level and at the level of available pages in the menu.

The PROFILE access object allows assigning appropriate access to the profile and then granting rights by assigning the profile to the user or groups.

In order to create a new object, e.g. a profile (PROFILE), click on [Add new object], then select the object type "PROFILES" and give the name of the object.

To assign permissions to a given object, select it from the list on the left side of the screen. After clicking on the object on the right side, the page with the access configuration will be displayed.

First you need to choose whether the permissions will be:

- own (Use own permissions).
- inherited permissions form parents.

DETAILS AND PRIVILEGES FOR SELECTED OBJECT

Object name
DESKTOP\ARTUR

Object Type
USER

Permissions Type
Use own permissions

6.9.4.1 Own permissions

If you choose (own permissions), you have three tabs to configure permissions:

- Function rights,
- Databases access,
- Custom privileges

Functional settings allow you to give rights to pages or functionality in the application at the global level for a given user / group or profile for all databases. You can override these rights by granting custom permissions for a specific database. Custom permissions can only be changed for the Instance Analysis module. Custom permission is superordinate to a given SQL Instance in relation to functional rights. If you assign custom permissions, the (permissions overwritten) message will be displayed next to the SQL Instance name.

Functions rights
Databases access
Custom privileges

Function privileges for Database Analysis module overwrite main function rights

Privileges for selected database
XE_2 (permissions overwritten)

☒ Database Analysis
☒ Performance
☒ I/O Stats
☒ Space monitor
☒ Memory
☒ Sessions
☒ Sessions
☐ Kill sessions
☒ Session Resources
☒ Backups
☒ Locks
☐ Kill sessions
☒ Parameters
☒ Logs
☒ Reports
☒ Performance report
☒ Not used indexes
☒ Plan Explorer
☒ Anomaly monitor

In addition, you can restrict access to specific databases. To do this, in the Database access tab, select the appropriate check boxes for a given SQL Instance or select ALL_INSTANCES. If certain bases are restricted, this will also limit the Custom privileges tab.

Functions rights	Databases access	Local privileges
Object access to databases		
Access	Database	
<input type="checkbox"/>	ALL INSTANCES	
<input checked="" type="checkbox"/>	CRMSQL31 on machine CRMSQL31	
<input type="checkbox"/>	SQL01\NAV_EE on machine SQL01	

6.9.4.2 Inherited permissions form parents

If you choose inherited rights, you can specify which profile or profiles to use for a given user or user group. Each profile contains a list of objects and access to which. Granting permissions to multiple profiles for the user will result in the entitlement for a given user being the sum of rights for selected profiles.

Profiles assignment	
Permissions to inherited from assigned profiles	
Access	Profile Name
<input type="checkbox"/>	ADMIN
<input type="checkbox"/>	ADMIN2
<input type="checkbox"/>	ADMIN3

Attention! In order to enable the functionality of limited access to the application, you must change the settings at the level of the DBPLUS Configuration Wizard> Applications settings> Applications Options> Configure. As well as change the status of the SECURITY parameter to ON.

Dashboard	Settings	Dashboard icon settings	Dashboard Tx Parameters
Database Analysis	List of configuration parameters. Please click on the edit button to change parameter value.		
Space monitor	APPLICATION PARAMETERS		
Parameters	Parameter	Value	Description
Reports	SECURITY	ON	Application can work in SECURITY mode set to ON or to OFF. It means that application uses (or doesn't use) user authentication. Setting the SECURITY to on, it requires at least one user created.
Server's monitor	DASHBOARD_ANIMATE_PARAMETERS	ON	Setting is valid for DPM dashboard displayed in television mode. Based on it each sql server icon can toggle/animate automatically its parameters like (server cpu, waits, sessions, etc.)
Configuration	LOCKING_SNAPSHOT_FREQUENCY	300	The interval time in seconds between each snapshot of locks made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.
Settings			Save
Databases			Edit
References lists			Edit
Security			
About settings			

Below screen with the DBPLUS Configuration Wizard:

Application pool settings (AppPoolDPM)

It's recommended to use the same user type/account for DBPLUSORACLECATCHER service, IIS application and oracle instances monitoring purposes.
Please do not use account with administrator privileges.

In application security tab please specify if application should be available in anonymous mode (for every user who enter the application url) or in secure mode (for users who authenticate)

If you want to change the protocol, you have to do it directly in IIS manager.

Login type: **LocalSystem**

Username:

Password:

Website settings (DBPLUS Website)

Protocol: **http** Binding property: **Default**

Port: **80** Host name:

Application path: **C:\Program Files (x86)\DBPLUS.Or** **Select application folder**

Application security

☐ Use windows authentication in access to application

Save configuration **Test settings** **Close**

After saving the settings, the next steps to manage the settings are made from the level of the web system application in the option **Configuration>Settings> Security parameter**.

DBPlus Better performance for ORACLE

Settings Dashboard parameters

List of configuration parameters. Please click on the edit button to change parameter value.

Parameter	Value	Description	
SECURITY	OFF	Application can work in SECURITY mode set to ON or to OFF. It means that application uses (or doesn't use) user authentication. Setting the SECURITY to on, it requires at least one user created.	Save
DASHBOARD_ANIMATE_PARAMETERS	ON	Setting is valid for DPM dashboard displayed in television mode. Based on it each sql server icon can toggle/animate automatically its parameters like (server cpu, sql instance cpu, waits, sessions, etc.)	Edit
DATABASE_CATEGORIZATION_PLUGIN	OFF	In DPM tool each database can have 3 different categories: database type, system category, vendor description. The list of categories and database assignment to those categories, can be managed in Configuration module (pages: References lists, Servers). Categorization is useful for environments with large number of instances and databases. It's used for filter/selection purposes and such categorizations can be included in the reports. To use this functionality, please set parameter DATABASES_CATEGORIZATION_PLUGIN to on value, otherwise to off	Edit
LOCKING_SNAPSHOT_FREQUENCY	30	The interval time in seconds between each snapshot of locks made by DBPLUS CATCHER service. The parameter can be setup separately for each instance. In a case of frequent locks, please consider lower value for LOCKING_SNAPSHOT_FREQUENCY. In a case of rarely occurred locks, please use bigger value for it.	Edit
MONITOR_LITERAL_QUERIES	OFF	DBPLUSCATCHER service can monitor literal queries executed on your databases. Set to [ON] to run feature on all databases or make such change for specified database.	Edit
KEEP_SNAPSHOT_HISTORY_DAYS	30	Number of days how long to keep detail statistics for sql statement executions, waits, latches, performance counters.	Edit

6.9.5 Alert Settings

The alert module is available from the main menu, i.e. **Configuration-> Alert settings**. From this tab users have the ability to:

- Parameter settings related to mailing - i.e. data of the mail server and account from which alert messages will be sent,
- Making general module settings,
- Define alerts,
- Specify the list of alert recipients.

6.9.5.1 Mail settings Tab

For the information about an alert to be sent via email, user must configure the SMTP server settings.

As part of the configuration, users have the option to set the frequency of sending information about the event, depending on the configuration it is from 1 minute to 1 hour.

IMPORTANT: Email alerts for all databases are sent from one email account.

6.9.5.2 General settings Tab

In this tab, users can make general settings of the alert module. User has the option to configure parameters related to the alert mechanism.

General parameters:

- **Elapsed Time greater than** - alerts will be calculated when in a given snap-time the duration for all queries exceeds 200 seconds.
- **History Days** - defining the days of the week that will be considered when examining performance problems.
- **Number of Days Back in History** - The number of historical days on the basis of which the system will test the performance of the current day.
- **Minimal History Days** - Specifies the minimum time after which trend-based alerts will be calculated

Statements Settings:

- **Number of Top Queries to check** - the number of top queries in individual snaps to be tested for performance problems, **Chosen by Elapsed Time / CPU Time** - the choice according to which the statistics will be selected Elapsed Time queries or CPU Time processor utilization time.
- **Number of Days Back in History** - The number of historical days based on which the system will analyze the performance of top queries on the current day.

Wait Events Settings:

- **Number of Top Waits to check** - used to handle waits calculated on the basis of the trend. The number of top waits depending on this parameter is taken into account for the calculation.
- **Number of Days Back in History** - how many days back, are taken into account for the calculation of history.

6.9.5.3 Alerts definition Tab

Defining alerts in the application has been divided into two stages:

- selection and configuration of appropriate CRITICAL / WARNING thresholds for a given type of alert,
- a rule definition based on configured alerts, and the attribution of the cause of the problem.

Website displays the information in columns:

- type of alert,
- description of the alert,
- availability,
- warning level,
- critical level.









The website presents only alerts that have been added to the configuration. If the alert has not been configured, please add it using the **[Add new alert]** button

Mail settings General settings **Alerts definition** Reasons & Problems definition Events subscription

Refresh



List of alerts which apply to all oracle databases. Please be aware that Online alerts are calculated every 30 seconds other alerts every 15 minutes. Any changes in below lists are recognizes by DBPLUS Catcher monitoring service up to 15 minutes

ALERTS CONFIGURATION Add new alert

Alert type	Alert description	Enabled	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	 <input checked="" type="checkbox"/>		
Online	Total Waits	 <input checked="" type="checkbox"/>	200 %	400 %
Online	Lock waits	 <input checked="" type="checkbox"/>	200 %	400 %
Online	Latches	 <input checked="" type="checkbox"/>	100 %	200 %
Online	Server CPU utilization	 <input checked="" type="checkbox"/>	300 %	500 %
Load Trends	Elapsed Time	 <input checked="" type="checkbox"/>	50 %	100 %
Load Trends	Wait Time	 <input checked="" type="checkbox"/>	30 %	80 %
Load Trends	Lock Time	 <input checked="" type="checkbox"/>	20 %	50 %

List of alerts on the instance level which are specific for particular database. Below settings overwrite main configuration. Those alerts which are marked in light gray color, are inherited from main configuration

INSTANCE ALERTS CONFIGURATION - PLEASE SELECT A DATABASE: T14 Add new alert Restore defaults

Alert type	Alert description	Enabled	Override	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	 <input checked="" type="checkbox"/>	<input type="checkbox"/>		
Online	Total Waits	 <input checked="" type="checkbox"/>	<input type="checkbox"/>	200 %	400 %

Alerts can be configured for all databases or for a dedicated database. At any time, user can delete the previously configured alert by using the [Key] button and selecting an option "Delete", this will delete the given alert from the configured list.




The second option is to disable the alert by unmark the "Enabled" checkbox. This can also be done by pressing the [Key] button and selecting the Edit option.

Mail settings General settings **Alerts definition** Reasons & Problems definition Events subscription

Refresh

List of alerts which apply to all oracle databases. Please be aware that Online alerts are calculated every 30 seconds other alerts every 15 minutes. Any changes in below lists are recognizes by DBPLUS Catcher monitoring service up to 15 minutes

ALERTS CONFIGURATION Add new alert

Alert type	Alert description	Enabled	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available	 <input checked="" type="checkbox"/>		
Online	Total Waits	 <input checked="" type="checkbox"/>	200 %	400 %
Online	Lock waits	 <input checked="" type="checkbox"/>	200 %	400 %

As part of the alert definition, user does not make the alert dependent on other alerts. Depending on the type of alert, threshold values are set in various ways.

Collecting data about problems in the application has been divided into 5 alert categories:

- **Online alerts** - calculated every 30 seconds,
- **Load Trends alerts** - calculated every 15 minutes based on general performance statistics,
- **Alerts type IO Stats** - calculated every 15 minutes on the basis of read / write statistics from / to disk devices,
- **Sql Query alerts** - calculated every 15 minutes based on statistics of top queries,
- **DB Size alerts** - calculated every 15 minutes based on space occupancy.

Alerts can be defined at the general level (for all bases) and at the level of individual databases. Two alarm thresholds can be defined for each alert:

- **WARNING** event - warning alert level
- **CRITICAL** event - high alert level - critical alert

For example: setting for the Load Trends category for the CPU Time alert.

Load Trends	Cpu Time		<input checked="" type="checkbox"/>	50 %	100 %
-------------	----------	--	-------------------------------------	------	-------

If the CPU utilization of the server exceeds 50%

- generate an alert at the warning level,

If the CPU utilization of the server exceeds 100%

- generate a critical alert

In other cases, there is no alert.

Window below presents the main list of alerts:

Mail settings
General settings
Alerts definition
Reasons & Problems definition
Events subscription

Refresh

List of alerts which apply to all oracle databases. Please be aware that Online alerts are calculated every 30 seconds other alerts every 15 minutes. Any changes in below lists are recognizes by DBPLUS Catcher monitoring service up to 15 minutes

ALERTS CONFIGURATION						Add new alert
Alert type	Alert description	Enabled	Level value WARNING	Level value CRITICAL		
IO Stats	Single Block Write time		20 %	50 %		
Load Trends	Elapsed Time		50 %	100 %		
Load Trends	Wait Time		30 %	80 %		
Load Trends	Lock Time		20 %	50 %		
Load Trends	Wait Event Time - [log file sync]		50 %	100 %		
Load Trends	Cpu Time		50 %	100 %		
Load Trends	Wait Event Time - [TCP Socket%]		50 %	100 %		
Load Trends	Wait Event Time - [job file sequential read]		50 %	100 %		

List of alerts on the instance level which are specific for particular database. Below settings overwrite main configuration. Those alerts which are marked in light gray color, are inherited from main configuration

INSTANCE ALERTS CONFIGURATION - PLEASE SELECT A DATABASE
EBAZY (1 alert/s overwritten)

Alerty definiowane na poziomie bazy danych

Możliwość przywrócenia ustawień ogólnych

Alert type	Alert description	Enabled	Override	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available		<input checked="" type="checkbox"/>		
Online	Total Waits		<input type="checkbox"/>	200 %	400 %

6.9.5.3.1 Online Alerts

The Online list includes the following alerts:

- **Active Sessions** - number of active sessions,
- **Number of Active Sessions with Elapsed Time longer than** - the number of active sessions with a duration longer than seconds
- **Lock waits** - lock type expectations,
- **Total Waits** - all expectations together,
- **Specific Wait** - an alert for a specific expectation,
- **Latches**
- **Server CPU utilization** - utilizing server CPU,
- **Custom alert calculated based on sql statement** - an alert calculated based on a freely arranged query,
- **Alert if SQL Instance is not available.**

The example alert tab of the alert looks like this:

Please note that the field specifying the type of alert (Online, Load Trends, IO Stats, Sql Query) is changeable only when creating a new definition. When re-editing the alert, the field is in read-only mode. Depending on the rule chosen, the list of available and required fields to be completed is changed.

For the alert: **Specific Wait** should be completed - the name of the wait for which the alert should react.

The following example will appear in the presented example:

an alert warning when the sum of expectations with a name containing reads exceeds at least 4 seconds / 1 second (a valid alert is not calculated here in percent).

critical alert when the sum of expectations with the name containing reads exceeds at least 10 seconds / 1 second (a valid alert is not calculated here in percent).

For the alert: **Custom alert calculated based on sql statement**, enter the query text.

ALERT DEFINITION

Alert

Online

Custom alert calculated based on sql statement

Enabled

☒

Sql statement query

Select count(*) from v\$session a ,v\$transaction b where a.saddr=b.ses_addr and a.status='INACTIVE' and a.last_call_et>500

The sql statement will be calculated every 30 seconds and needs to return single value which will be compare to alert level values. The test is made on database with dbplus repository

Test query

Alert Levels

Notifications & Conditions

Other settings

Set level to WARNING when Calculated value is above

10

Set level to CRITICAL when Calculated value is above

40

OK

Cancel

IMPORTANT: the query must return a single-column record. The alert will occur when the value returned by the query exceeds the thresholds according to the given definition.

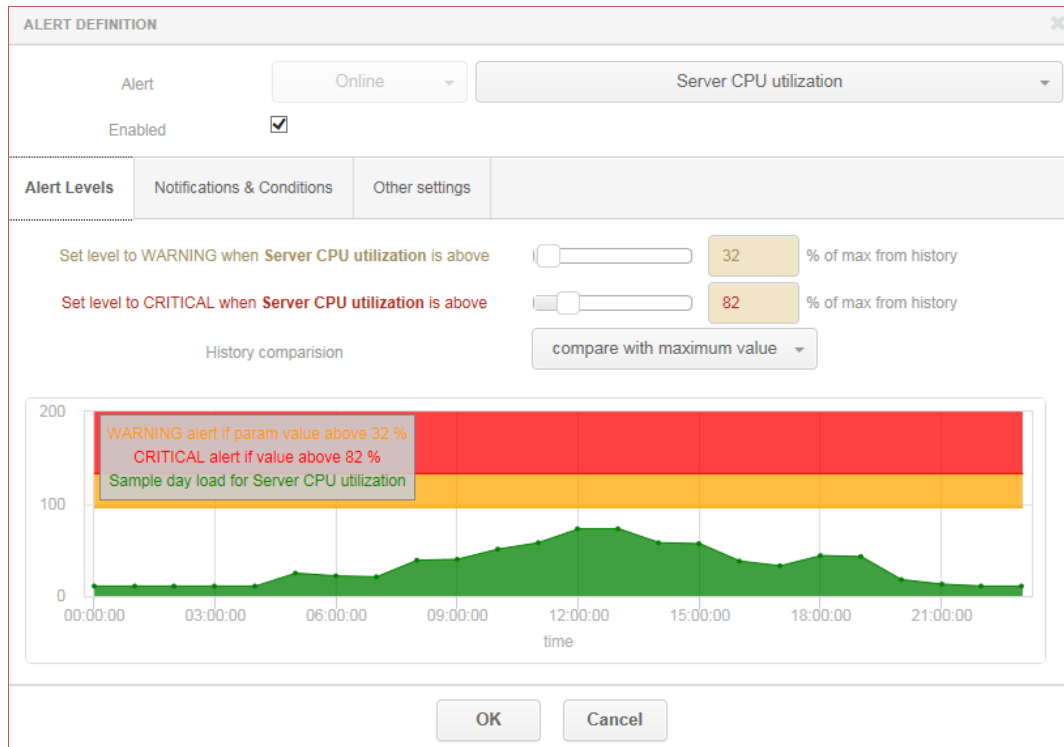
The following example will appear in the presented example:

- alert warning when the number of inactive sessions with an open transaction in the SQL Instance exceeds at least 10 sessions
- critical alert when the number of inactive sessions with an open transaction in the SQL Instance exceeds at least 40 sessions

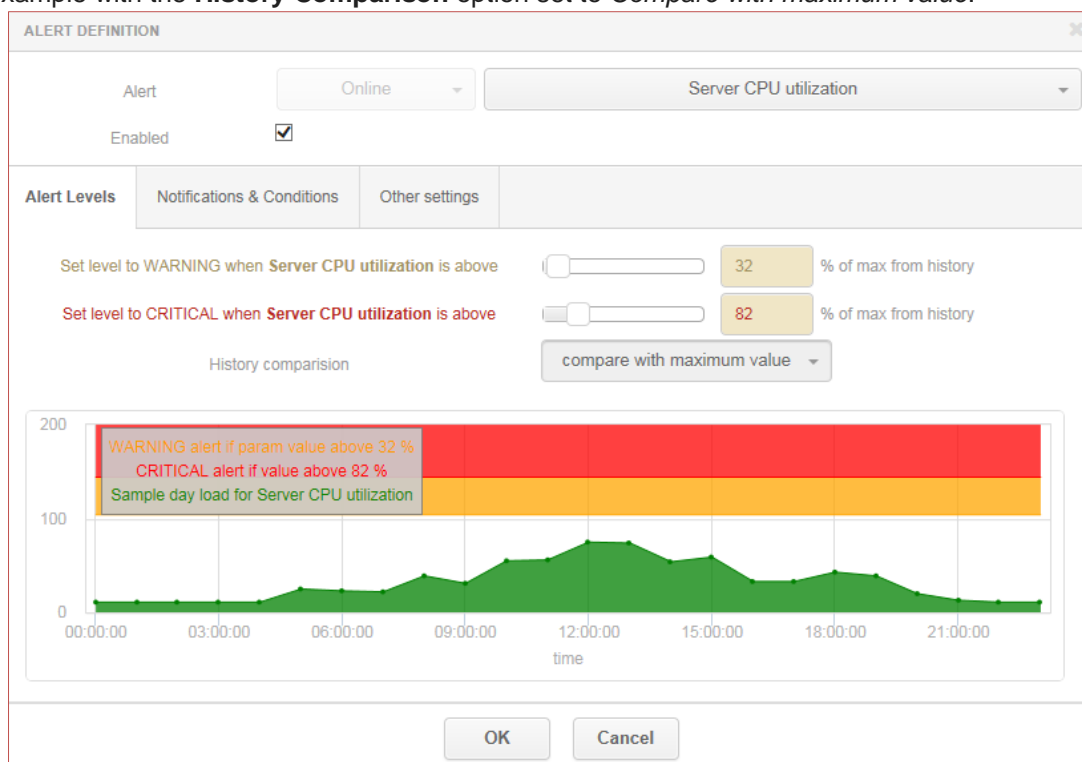
For the alert: Server CPU utilization, defines standard parameters, i.e.

- Alert thresholds **WARNING**, **CRITICAL**
- The way of calculating and reaction of the Alert on the History Comparison event (comparison of the performance of a given parameter with the history)
 - Compare to average value in similar time - the performance of a parameter is compared to the statistics history at similar times
 - Compare with maximum value - the performance of a parameter is compared with the maximum values that were present for a given statistic.

The following screen with the option of **History Comparison** set to *Compare to average value in similar time*:



An example with the **History Comparison** option set to *Compare with maximum value*:



This slide will showcase:

Alert warning when the disposal of server processors will be 32% greater than the maximum historical value.

Critical alert when the utilization of server processors will be 82% greater than the maximum historical value.

In the alert edit tab, additional settings can be found in the Notification & Condition tab:

- **Mail Notification Interval** - how often to generate an email notification when an alert occurs
- **Number of snapshots to check** - the number of 30 seconds of snapshots in which there must be a "problem" for a given parameter. If a given statistic, e.g. Total Waits - stays at a high level and exceeds the alert threshold by X snapshots, then the system will generate an alert
- **Use Low Constant Value** - the minimum value that must be met first. According to the example screen below - within the dashboard snapshot (started in a 30-second cycle) the value of all wait-time must be at least 30 seconds.
- **Use High Constant Value** - the value at which the alert will always be generated, even if the WARNING, CRITICAL alert thresholds are not met.

Alert Levels	Notifications & Conditions	Other settings
Alert Calculation Interval		once per 30 seconds
Mailing Notification Interval		once per 5 minutes
Filter conditions		
Use Low Constant Value		30 s. Every alert with value below entered will be skipped
Use High Constant Value		60 s. Every alert with value above entered will be shown
Snapshot conditions		
Number of snapshot to check		5 in which property must exceed alert level value

6.9.5.3.2 Load Trends, I/O Stats alerts type

The Load Trends, I / O Stats rules refer to performance indicators available on website (functionalities) with the same names.

In the case of **Load Trends**, the system allows users to alter the following indicators:

- Session active count
- Buffer gets
- Buffer writes
- CPU Time
- Disk reads
- Elapsed Time
- Executions
- Latch time
- Lock time
- Rows Processed
- Session count
- Wait time
- Wait Event Time

For **IO Stats**, users have the following indicators:

- MB Reads
- MB Writes
- Disk Reads
- Read Time
- Single Block Read Time
- Single Block Write Time

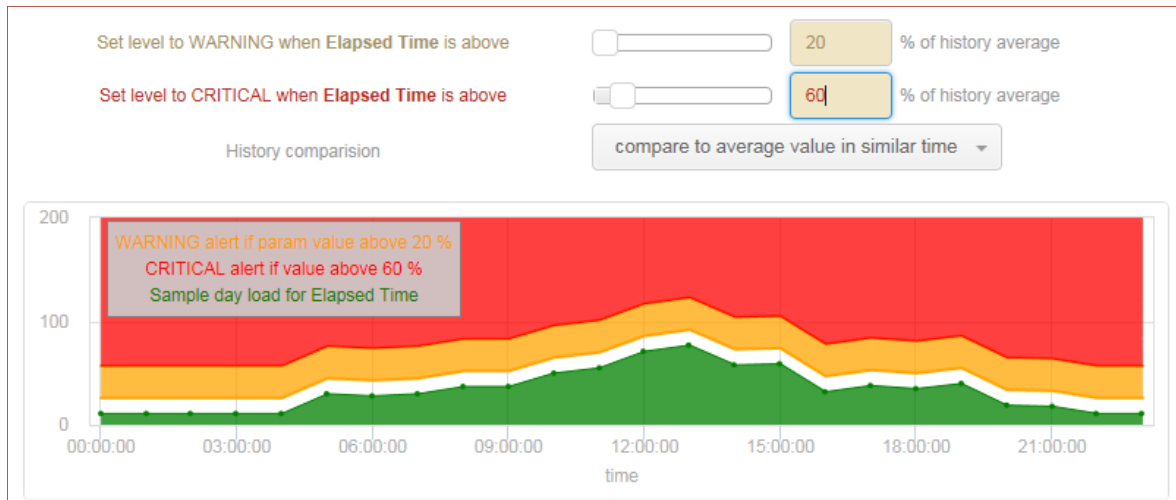
- Disk Writes
- Write Time

The edit tab of such alert looks like the below:

In the form, the user specifies the following:

- Type of alert (according to the indicators given above)
- Is enabled
- Own name - **Other settings** tab
- Message format - **Other settings** tab
- E-mail settings - spam protection in case of an ongoing alert - Notification & Conditions tab
- **When and with what threshold an alert will occur:**
 - The rule is calculated as a percentage.
 - The alert will occur when the given alert threshold is exceeded by X% in relation to the average over the past period.
 - In the **Filter condition** section, we have additional filter settings, i.e.:
 - **Use Low Constant Value** - e.g., alert when Elapsed Time will deteriorate from X% in relation to the average, but in a situation where Elapsed Time is greater than 500 seconds.
 - **Use High Constant Value** - as above

Below are some examples of definitions for the **Elapsed Time** parameter - with the option of **History Comparison** set to Compare to average value in similar time:



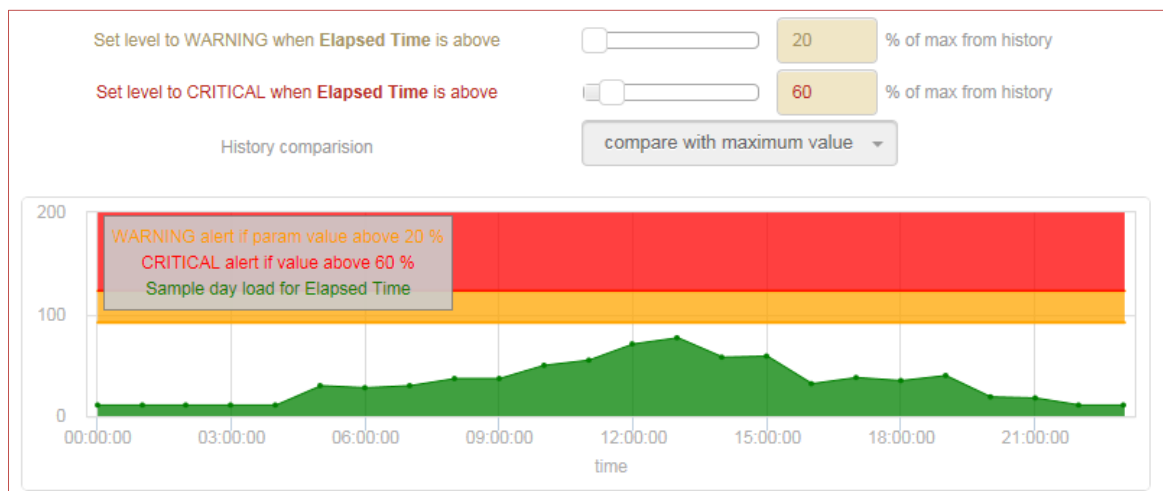
According to the above, the other load occurs during business hours and outside business hours. For example:

The duration of all queries, i.e. Elapsed Time at 08:00, is historically 1000 s in a 15-minute snapshots.

The duration of all queries, i.e. Elapsed Time at 12:00, is historically 5000 seconds in a 15-minute snapshots.

Alert warning type WARNING for a defined threshold $\geq 20\%$ will occur at 08:00, when the duration of all queries exceeds 1200 seconds, while around 12:00, when Elapsed time exceeds 6000 seconds.

For the second case with the History Comparison option set to Compare with maximum value:



In this example:

WARNING for the defined threshold $\geq 20\%$ will occur only if the duration of all queries exceeds 6000 seconds (reference to the maximum value of the day) regardless of the time of day.

6.9.5.3.3 Sql Query alerts type

SQL Query rules apply to performance indicators available for SQL queries and contain a similar list as for Load Trends.

For SQL queries, the system allows users to alter the following indicators:

- Buffer gets
- Buffer writes
- CPU Time
- CPU time per 1 exec
- Disk reads
- Elapsed Time
- Elapsed Time Per 1 Exec
- Execution
- Rows Processed
- Wait Time

In addition, the list of rules also includes:

- New statement Elapsed Time
- New statement CPU Time

The SQL Query Alert Definition tab looks like this:

ALERT DEFINITION

Alert: Sql Query Elapsed Time per 1 exec

Enabled: ☒

Alert Levels: Notifications & Conditions Other settings

Set level to WARNING when **Elapsed Time per 1 exec** is above % of max from history

Set level to CRITICAL when **Elapsed Time per 1 exec** is above % of max from history

☐ Show Plan Changes Only

OK Cancel

In the form, the user specifies similar parameters as in the alert definition for Load Trends statistics, IO Stats. In addition, user can indicate whether the alert reacts only when the execution plan is changed - the **Show Plan Changes Only** flag (assuming that the indicator has deteriorated in relation to the history).

Alert type	Alert description	Enabled	Level value WARNING	Level value CRITICAL
Sql Query	Execution	<input checked="" type="checkbox"/>	50 %	100 %
Sql Query	Elapsed Time (for plan changes only)	<input checked="" type="checkbox"/>	50 %	100 %
Sql Query	Elapsed Time per 1 exec (for plan changes only)	<input checked="" type="checkbox"/>	50 %	100 %
Sql Query	Disk reads (for plan changes only)	<input checked="" type="checkbox"/>	50 %	100 %
Sql Query	Execution (for plan changes only)	<input checked="" type="checkbox"/>	50 %	100 %
Load Trends	Elapsed Time	<input checked="" type="checkbox"/>	50 %	100 %
Load Trends	Wait Time	<input checked="" type="checkbox"/>	30 %	80 %

For alerts with the **New Statement** prefix, the thresholds are determined at the level of the share in the sql instance load.

The application allows the dependence of an alert instance on the general trend (for the entire database) for a given statistic in the snap. This option is only available for SQL Query type alerts. For the configuration shown in the picture below, this means for the SQL Query Rows processed type alarm:

- the alarm will be skipped if the value of Rows processed for a given snap for a specific Query Hash is below 10 and if the number of returned rows processed for a given query is less than 15% of all returned rows for queries (the number depends on the Number of Top Queries to check). Additionally, the condition of exceeding the WARNING / CRITICAL alarm threshold must be met.
- the alarm will occur if the value of Rows processed for the given snap in the query is above 25%. The alarm will occur even if the alarm threshold has not been exceeded (then WARNING will occur with the Above max constant comment ...).

6.9.5.3.4 Alert settings at the SQL Instance level

The list of alerts can be set for each base independently. By default, alerts are inherited from general settings. If any alert parameter is changed then the information appears in the Override column about overriding this rule.

As the example below:

ALERTS CONFIGURATION					Add new alert
Alert type	Alert description		Enabled	Level value WARNING	Level value CRITICAL
Online	Alert if database is not available		<input checked="" type="checkbox"/>		
Online	Number of active sessions with Elapsed time longer than 0,03 seconds		<input checked="" type="checkbox"/>	2	5

List of alerts on the instance level which are specific for particular database. Below settings overwrite main configuration. Those alerts which are marked in light gray color, are inherited from main configuration

INSTANCE ALERTS CONFIGURATION - PLEASE SELECT AN INSTANCE: WIN-PVM04LCT8A/SQLSERVER (1 alert's overwritten)						Add new alert	Restore defaults
Alert type	Alert description	Enabled	Override	Level value WARNING	Level value CRITICAL		
Online	Alert if database is not available (test)	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
Online	Total Waits	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200 %	400 %		

The system will generate an alert for all databases except this one. At the WIN SQL Instance Alert level, *Alert if database is not available*, has been disabled (Enabled = false).

6.9.5.4 Reasons and Problems definition Tab

The next stage of alarm configuration consists of, assigning rules and defining the dedicated cause of the problem. Screen below shows an example of a list of alarms defined by default by DBPLUS analysts. Definitions can be assigned at a general level to all databases or create dedicated definitions for selected databases.

Mail settings General settings Alerts definition **Reasons & Problems definition** Events subscription

Refresh

List of performance problems which apply to all oracle databases. Please be aware that Online issues are calculated every 30 seconds other problems every 15 minutes. Any changes in below lists are recognizes by DBPLUS.Catcher monitoring service up to 15 minutes

REASON & PROBLEMS CONFIGURATION Add new definition

Type	Class	Reason/Problem description	Enabled	
Trends	Lock	Problems cause locking wait	<input checked="" type="checkbox"/>	Trends:Lock Time AND Trends:Wait Event Time
Trends	I/O	Problems with Disk reads increase cause query change plan	<input checked="" type="checkbox"/>	(Trends:Cpu Time AND Trends:Elapsed Time) AND ((SQLQuery:Cpu Time (for plan changes only) AND SQLQuery:Cpu Time pe
Trends	Other	Problems with Query CPU Time Increase cause query change plan	<input checked="" type="checkbox"/>	Trends:Cpu Time AND (SQLQuery:Cpu Time per 1 exec (for plan changes only) OR (SQLQuery:Cpu Time (for plan changes only)
Trends	Other	Problems cause Query CPU Time Increase	<input checked="" type="checkbox"/>	Trends:Cpu Time AND (SQLQuery:Cpu Time AND SQLQuery:Cpu Time per 1 exec)
Online	Online	Increase of waits events (cause of Locks) on database in last 3 minutes	<input checked="" type="checkbox"/>	Online:Lock waits
Trends	Other	Problems cause wait: PAGEIOLATCH_SH	<input checked="" type="checkbox"/>	Trends:Wait Time AND Trends:Wait Event Time - [PAGEIOLATCH_SH]
Trends	I/O	Problems cause increase Executions and Disk Reads.	<input checked="" type="checkbox"/>	(Trends:Cpu Time AND Trends:Elapsed Time) AND ((SQLQuery:Cpu Time AND SQLQuery:Cpu Time per 1 exec AND SQLQuer

To add a new rule, first define the reason for the problem (Reason description) for which the rule will be defined. Next, choose the type of calculation (Calculation type) - based on the trend or online and Reason class.

REASON DEFINITION

Reason description: Network problem not caused by I/O disk storage issues

Calculation Type: Based on Trends

Reason Class: I/O

Enabled: ☒

Rules & Formulas

Notifications & Conditions

AND OR Add rule Add group

Trends:Wait Event Time - [TCP Socket%] Delete

AND OR Add rule Add group Delete

AND OR Add rule Add group Delete

NOT:IO:Disk reads Delete

NOT:IO:Single Block Read time Delete

Rules preview: Trends:Wait Event Time - [TCP Socket%] AND ((NOT:IO:Disk reads AND NOT:IO:Single Block Read time) OR (NOT:IO:Disk writes AND NOT:IO:Single Block Write time))

OK Cancel

The most important element of the configuration is to create the cause of the problem and then define the appropriate rules based on alerts. To add a configuration, from the previously defined alerts (Alerts definition tab), create a rule using groups (Add group), AND, OR operators. In some cases, it is necessary to use negation, they are presented in the list of alerts marked in red and start with the NOT operator.

After defining the rule, correctly selecting the operators and completing all added alarms, the rule will be displayed below.

Rules preview: (Trends:Elapsed Time AND Trends:Wait Time AND Trends:Execution) AND (NOT:SQLQuery:New Statement Cpu Time OR NOT:SQLQuery:New Statement Elapsed Time) AND NOT:IO:Single Block Read time AND SQLQuery:Elapsed Time

6.9.5.5 Events subscription Tab

In the last tab of the module user has the ability to manage the list of recipients, i.e. people who will receive alert messages.

The list of subscribers can be:

- a single email address or multiple addresses separated by a separator
- assigned recipient's email address to all or selected databases.

6.9.5.6 Visibility of alerts

Alerts are visible from the Anomaly Monitor menu and also from the:

- **Dashboard Level:**
 - the base icon contains information about the number of alert and critical alerts
 - after selecting a given SQL instance in the **Alerts** and **Instance Load** tab
- after clicking **[Instance Analysis]** on the **Instance Load** graph
 - if any Alert have occurred on the Elapsed Time line, relevant information is displayed about their number
- after clicking on a given time point (snapshot) - a list of alerts is displayed

6.10 Help menu

The site contains information about licenses and information about changes in applications made in the last year.

6.11 Additional information

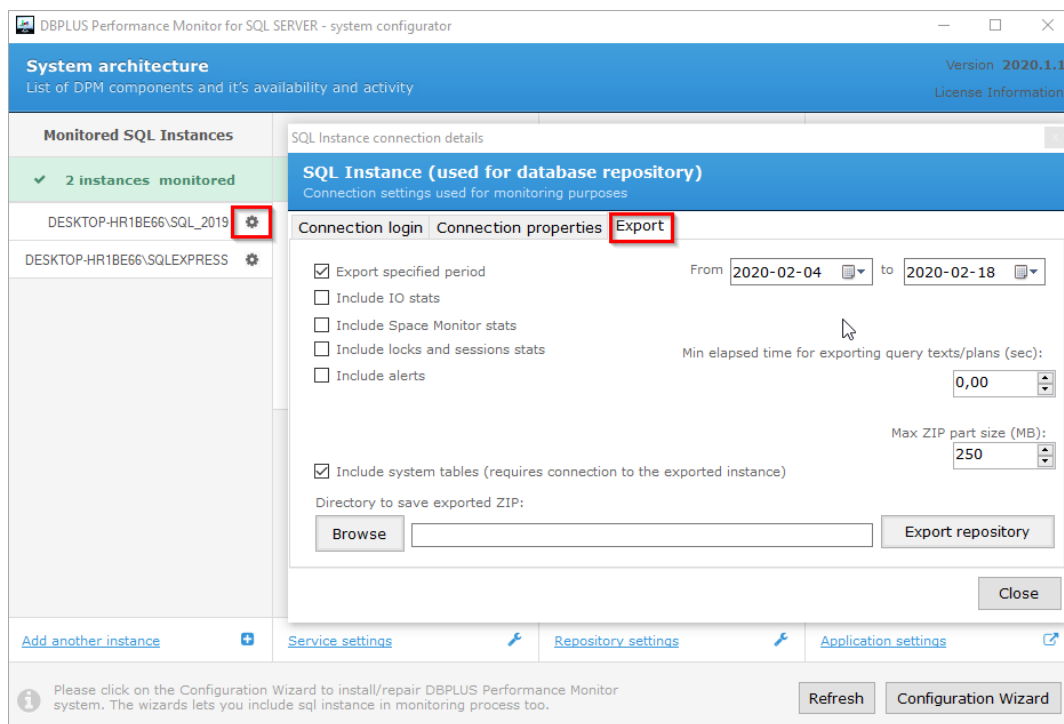
6.11.1 Export / Import statistics of monitored databases

In the latest version we have added the ability to export / import data collected by DBPLUS Performance Monitor. The user makes both export and import within the same platform, in this case SQL Server. In the current version, the mechanism allows to export the entire database (without date range) as well as selected periods and monitoring modules. Import requires a previously configured repository database, and the result is the addition imported database as a new Sql instance to the current repository.

The imported database is added to the monitoring by default. Collecting data through the "Scheduled Outages" mechanism is blocked (performance statistics are not collected). Enabling statistics collection of an imported database is possible by changing the settings in the Configuration> Scheduled Outages menu for the appropriate SQL Instance.

SQL Instance export

The export can be made from the level of DBPLUS Configuration Wizard. After starting the program, the user goes to the details screen of the instance whose data user wants to export by clicking the button [cog] next to the given database. Then it goes to the Export tab, like below:



The next step is to select the export option. The user can configure:

- date range for which statistics going to be exported (if selected),
- minimum duration of queries (queries below this value will not be exported),
- maximum export file size (another file will be created above this size).

Additionally, the user can choose which data will be exported by select options:

- Include IO stats
- Include Space Monitor Stats

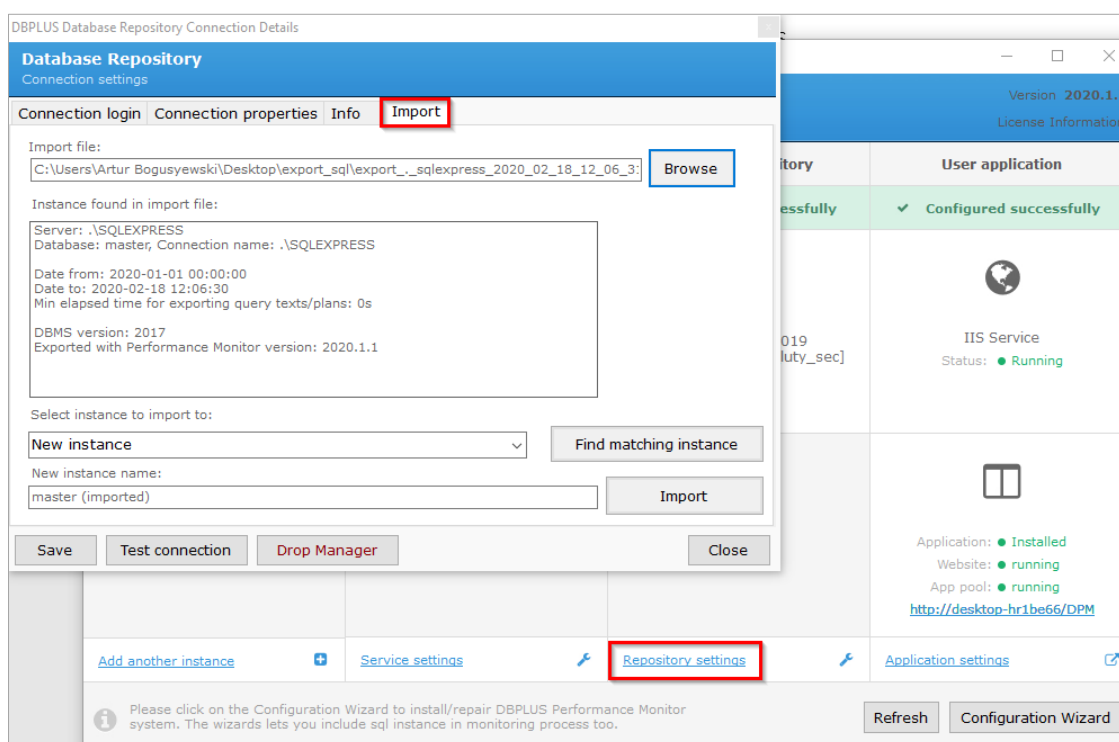
- Include locks and sessions stats
- Include alerts
- Include system tables.

If the "Include system tables" option is selected, online access to the exported database is required to download the data.

After selecting the data for export, indicate the directory where the data file will be saved. A log file will be created in the export directory with information on what data has been exported, and saved data with the exported file or files (depending on the size of the export).

SQL Instance import

The import process can also be performed from the Dbplus Configuration Wizard. The imported database can be added as a new instance as well as part of an instance previously configured for monitoring. The import is started by clicking the "Repository settings" link from the Configuration Wizard and then user goes to the Import tab as shown below:



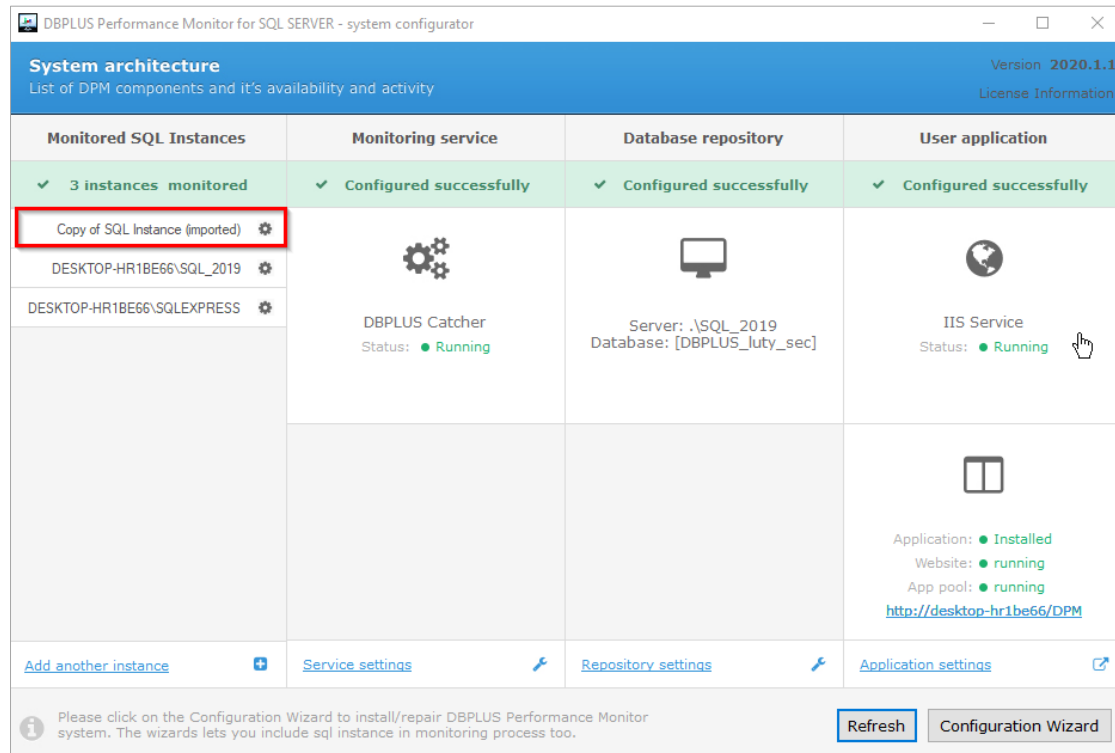
The first step in importing an SQL Instance is to point to the file with the previously exported database. In case of many export files (the export may contain several files), as the file to be imported, the user indicates file with the extension *.zip. After selecting file, information about the content of files will be presented.

Then indicate whether the user wants to create a new instance after import or add statistics to an existing database in monitoring.

Important! If you select an existing instance, remember to monitor the continuity of monitored data. If the date ranges overlap, some data may be overwritten and lost without being able to be restored.

After selecting the appropriate option, the Sql instance import from the file begins by clicking the [Import] button. After the import is completed, a message about successful import will be presented or error information will be presented, and the details will be saved to the log file created in the import directory.

After successful import, the instance will be added to the repository and visible in the Configuration Wizard, as shown below:



The instance will also be visible from the Dashboard screen. All imported statistics will be available on the Dbplus Performance Monitor screens, similarly to a running instance. Some screens present data directly by downloading it from the monitored database, in such cases information will only be visible if such a connection is possible.