

DBPLUS  
Performance Monitor for Oracle  
description of changes in version 2022.2

*Date: July 10, 2022*

*Table of Contents*

<b>1</b>	<b><i>Session Trace profiler</i></b> .....	<b>3</b>
<b>2</b>	<b><i>REST API – Performance Monitor</i></b> .....	<b>5</b>
	2.1. <i>Performance Counters</i> .....	5
	2.2. <i>IO Stats</i> .....	6
<b>3</b>	<b><i>Anomaly monitor improvements</i></b> .....	<b>8</b>
<b>4</b>	<b><i>Oracle AWS Cloud support</i></b> .....	<b>9</b>
<b>5</b>	<b><i>Bug fixes and improvements</i></b> .....	<b>9</b>
	5.1. <i>Incomplete query text in the repository</i> .....	9
	5.2. <i>Implementation of TLS1.1 and TLS1.2 support</i> .....	9
	5.3. <i>Query Advisor and SQL Parser update</i> .....	9

Below is a list of changes to the DBPLUS Performance Monitor system for Oracle database monitoring.

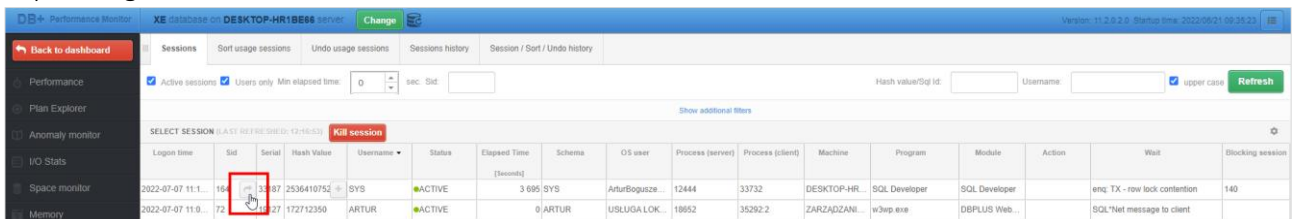
## New in 2022.2 version

### 1 Session Trace profiler

In the latest version of the application, we have added the functionality of session monitoring using Trace Profiler. This option is available from the level of each monitored instance from the **Sessions** menu. The functionality allows you to monitor the users' session regardless of the functionality available in the application by default.

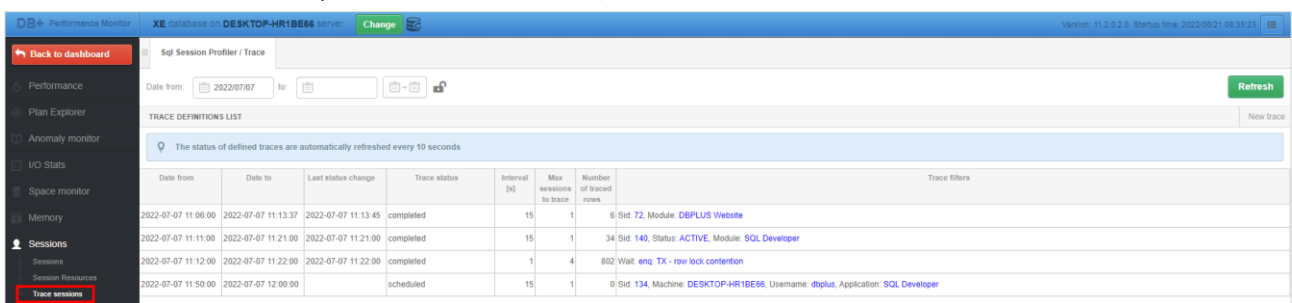
Session monitoring can be started in two ways:

- by clicking the button in the **Sid** column for a dedicated session on the online session screen:



Logon time	Sid	Serial	Hash Value	Username	Status	Elapsed Time	Schema	OS user	Process (server)	Process (client)	Machine	Program	Module	Action	Wait	Blocking session
2022-07-07 11:11:16	72	33887	2536410752	SYS	ACTIVE	3:085	SYS	ArturBogusze...	12444	33732	DESKTOP-HR...	SQL Developer	SQL Developer		enq: TX - row lock contention	140
2022-07-07 11:0...	72	172712350	172712350	ARTUR	ACTIVE	0	ARTUR	USLUGA LOK...	18652	35292.2	ZARZADZANI...	w3wp.exe	DBPLUS Web...		SQL*Net message to client	

- from the **Sessions**> **Trace profiler** menu by clicking **[New trace]**:



Date from	Date to	Last status change	Trace status	Interval [s]	Max sessions to trace	Number of traced rows	Trace filters
2022-07-07 11:06:00	2022-07-07 11:13:37	2022-07-07 11:13:45	completed	15	1	6	Sid: 72, Module: DBPLUS Website
2022-07-07 11:11:00	2022-07-07 11:21:00	2022-07-07 11:21:00	completed	15	1	34	Sid: 140, Status: ACTIVE, Module: SQL Developer
2022-07-07 11:12:00	2022-07-07 11:22:00	2022-07-07 11:22:00	completed	1	4	802	Wait: enq: TX - row lock contention
2022-07-07 11:50:00	2022-07-07 12:00:00		scheduled	15	1	0	Sid: 134, Machine: DESKTOP-HR1BE66, Username: dbplus, Application: SQL Developer

In both cases, after clicking, a dedicated window will appear in which we can set the conditions with which the session monitoring is to be started.

The basic settings include:

- Start date, End date - start and end time of session monitoring,
- Trace interval - the interval of retrieving information about sessions,
- Max number of session to trace - the maximum number of monitored sessions.

The minimum interval for session monitoring is 1 second. It means that every second a command with given conditions will be executed checking information about the session.

By setting up session monitoring, we can configure filters for:

- Sid - session ID
- Machine - server name
- Session status - the session status
- User name - login name
- Wait name - name of the wait
- Application / Program - the name of the application / program
- Os user - Windows user name
- Module
- Action

**TRACE DEFINITION**

Start date: 2022/07/07 11:50

End date: 2022/07/07 12:00

Trace interval: 15 second(s)

Max number of sessions to trace: 1

---

**Filters**

Sid: 134

Machine: DESKTOP-HR1BE66

Session status: Not selected

User name: dbplus

Wait name:

Application/Program: SQL Developer

Os user:

Module:

Action:

After starting the session monitoring, the session data will be presented at the bottom of the screen after clicking on the selected row. The monitoring screen refreshes automatically once every 10 seconds. During the session monitoring, the user has the option to modify the filters or stop the monitoring currently being performed.

When the session monitoring is completed, the (Trace status) will change to completed.

The user can delete previously performed monitoring sessions, or wait for them to be deleted automatically based on the parameter set for the length of data storage in the repository in the menu **Configuration> Settings> History settings section, Session / Locks statistics**.

### Access to the Trace Profiler menu

If the **Trace profiler** option is not visible in the **Sessions** menu, please verify that the access to this menu has been granted (by default the Trace profiler menu is invisible). To grant access, on the Dashboard screen in the main menu, select **Configuration> Security** and then grant appropriate permissions.

The screenshot shows the DBPLUS configuration interface. On the left, the 'Configuration' menu is expanded, and 'Security' is selected. The main area displays a table of users and their permissions. The 'DBPLUS User' is highlighted in green. The 'Functions rights' section is expanded, showing a list of permissions with checkboxes. The 'Trace sessions' checkbox is checked and highlighted with a red box.

Name	Description	Type	Permissions
CEVmkas		USER	Own
CEVmonito		USER	Own
CEVvbare		USER	Own
icabogusze	DBPLUS User	USER	Own
ICIDMARKO		USER	Own
icipol		USER	Own
icirmakuch		USER	Own

**Functions rights**

- License
- About
- Manual
- Instance Analysis
  - Performance
  - I/O Stats
  - Space monitor
  - Memory
- Sessions
  - Sessions
    - Kill sessions
    - Cpu monitor
    - Online backup
    - Trace sessions
- Backups
- Locks / Deadlocks
  - Kill sessions
- Parameters

## 2 REST API – Performance Monitor

In the latest version of the application, we have added new methods to the REST API:

- get PerfCounters performance statistics information,
- get information about IO Stats disk array statistics.

### 2.1. Performance Counters

<b>Method</b>	GET
<b>Database platform</b>	Oracle, MS SQL (from version 2022.2)
<b>URL</b>	/perfcounters
<b>Action</b>	Gets information about performance statistics (Performance Counters)
<b>Input data:</b>	
view	<ul style="list-style-type: none"> <li>▪ last_snapshot</li> <li>▪ history</li> </ul> <p>* in the case of the history option, additional filters must be completed (group_time, date_from, date_to)</p>
performance_counter	Counter name * supports condition like '% name%' (returns max. 3 statistics that meet the condition)
server_id	Server identifier in the DBPLUS repository
group_time	Grouping of returned data (return date format): <ul style="list-style-type: none"> <li>▪ year (YYYY)</li> <li>▪ month (YYYY-MM)</li> <li>▪ day (YYYY-MM-DD)</li> <li>▪ hour (YYYY-MM-DD HH24)</li> <li>▪ snap (YYYY-MM-DD HH24:MI:SS)</li> </ul>
date_from	Date from which statistics will be downloaded *In format YYYY-MM-DD HH24:MI:SS
date_to	Date by which statistics will be downloaded *In format YYYY-MM-DD HH24:MI:SS
<b>Output data:</b>	
PerfCounterList	Counters list
PerfCounterRecord	Statistic record
ClassGroup	Statistics class
Name	Statistics name
Value	The value of the statistics
Logdate	Date for the given statistic
<b>Example data output [xml]:</b>	
<pre>&lt;?xml version="1.0" encoding="utf-16"?&gt; &lt;Root xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"&gt;   &lt;PerfCounterList&gt;     &lt;PerfCounterRecord&gt;       &lt;ClassGroup&gt;Debug&lt;/ClassGroup&gt;       &lt;Name&gt;background timeouts&lt;/Name&gt;       &lt;Value&gt;0&lt;/Value&gt;       &lt;Logdate&gt;2022-06-02 08&lt;/Logdate&gt;     &lt;/PerfCounterRecord&gt;   &lt;/PerfCounterList&gt; &lt;/Response&gt;   &lt;Status&gt;OK&lt;/Status&gt;   &lt;Message /&gt; &lt;/Response&gt; &lt;/Root&gt;</pre>	

**Example data output [JSON]:**

```
{
  "PerfCounterList": [
    {
      "ClassGroup": "SQL Statistics",
      "Name": "Batch Requests/sec",
      "Value": 2227,
      "Logdate": "15.06.2022 10:21:33"
    }
  ],
  "Response": {
    "Status": "OK",
    "Message": ""
  }
}
```

**2.2. IO Stats**

<b>Method</b>	GET
<b>Database platform</b>	PostgreSQL, Oracle, MS SQL (from version 2022.2)
<b>URL</b>	/iostats
<b>Action</b>	Gets information about IO statistics
<b>Input data:</b>	
View	<ul style="list-style-type: none"> <li>▪ last_snapshot</li> <li>▪ history</li> </ul> <p>* in the case of the history option, additional filters must be completed (group_time, date_from, date_to)</p>
server_id	Server identifier in the DBPLUS repository
group_time	Grouping of returned data (return date format): <ul style="list-style-type: none"> <li>▪ year (YYYY)</li> <li>▪ month (YYYY-MM)</li> <li>▪ day (YYYY-MM-DD)</li> <li>▪ hour (YYYY-MM-DD HH24)</li> <li>▪ snap (YYYY-MM-DD HH24:MI:SS)</li> </ul>
group_type	Group type: <ul style="list-style-type: none"> <li>▪ Database</li> <li>▪ Tablespace</li> <li>▪ File</li> </ul> <p>* no field completed means options without grouping</p>
date_to	Date by which statistics will be downloaded *In format YYYY-MM-DD HH24:MI:SS
date_from	Date from which statistics will be downloaded *In format YYYY-MM-DD HH24:MI:SS
database	Database name * used in MSSQL, PostgreSQL
tablespace	Tablespace name *used in Oracle
file_name	File name *used in MSSQL, Oracle
<b>Output data:</b>	
IODataList	List of IO statistics
IOStatRecord	Record of IO statistics
Database	Database name
Tablespace	Tablespace name
File	File name
Logdate	Date for the given statistic

NumberOfReads	Number of reads
NumberOfWrites	Number of writes
BytesReads	The number of bytes read
BytesWrites	Number of bytes written
MBytesReads	The number of reads in [MB]
MBytesWrites	The number of writes in [MB]
BlockReads	The number of blocks read
BlockWrites	Number of blocks written
ReadTime	Reading time
WriteTime	Writing time
SingleMByteReadTime	Single MB read time
SingleMByteWriteTime	Single MB write time
SingleBlockReadTime	Single block read time
SingleBlockWriteTime	Single block write time

#### Example output data [xml]:

```
<?xml version="1.0" encoding="utf-16"?>
<Root xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <IODataList>
    <IOStatRecord>
      <Database>All databases</Database>
      <Tablespace>N/A</Tablespace>
      <File>All files</File>
      <Logdate>2022-06-15 11:28:50</Logdate>
      <NumberOfReads>739</NumberOfReads>
      <NumberOfWrites>637</NumberOfWrites>
      <BytesReads>48234496</BytesReads>
      <BytesWrites>5242880</BytesWrites>
      <MBytesReads>46</MBytesReads>
      <MBytesWrites>5</MBytesWrites>
      <BlockReads>5888</BlockReads>
      <BlockWrites>640</BlockWrites>
      <ReadTime>1.61</ReadTime>
      <WriteTime>2.112</WriteTime>
      <SingleMByteReadTime>0.035</SingleMByteReadTime>
      <SingleMByteWriteTime>0.4224</SingleMByteWriteTime>
      <SingleBlockReadTime>0.000273</SingleBlockReadTime>
      <SingleBlockWriteTime>0.0033</SingleBlockWriteTime>
    </IOStatRecord>
  </IODataList>
  <Response>
    <Status>OK</Status>
    <Message />
  </Response>
</Root>
```

#### Example output data [JSON]:

```
"IOStatList": [
  {
    "Database": "All databases",
    "Tablespace": "N/A",
    "File": "%C:\\Program Files\\Microsoft SQL Server\\MSSQL15.SQL_2019\\MSSQL\\DATA\\ABCD%",
    "Logdate": "2022-06-28 14:17:10",
```

```

        "NumberOfReads": 85,
        "NumberOfWrites": 25875,
        "BytesReads": 4194304,
        "BytesWrites": 238026752,
        "MBytesReads": 4,
        "MBytesWrites": 227,
        "BlockReads": 512,
        "BlockWrites": 29056,
        "ReadTime": 0.343,
        "WriteTime": 69.303,
        "SingleMByteReadTime": 0.08575,
        "SingleMByteWriteTime": 0.305300,
        "SingleBlockReadTime": 0.000670,
        "SingleBlockWriteTime": 0.002385
    }
  ],
  "Response": {
    "Status": "OK",
    "Message": ""
  }
}

```

### 3 Anomaly monitor improvements

In the latest version of the application, we have made changes to the process of generating the Anomaly Monitor report containing the performance anomalies detected by the Performance Monitor applications. The changes consist in including in the report only the largest cases of a given problem in the analyzed period of time. The change consists in adding a dedicated parameter which is responsible for filtering out the occurrences of a given problem in the period for which the report is generated and taking into account only those occurrences that exceed the threshold indicated in the parameter. The parameter value is set to **10%** by default. This will allow the Anomaly Monitor report not to include information about problem occurrences that are not the main performance problem in the monitored instance.

If the user would like the report to describe all performance problems and all occurrences of the problem, it is enough to change the value for a given parameter to **Show All**, which will mean that all problems will be described in the report.



## 4 Oracle AWS Cloud support

In the latest version, we have added support for monitoring Oracle databases in the AWS version. The scope of functionality available in the DBPLUS Performance Monitor application is the same for the AWS database version.

When choosing AWS solutions, we recommend that both the databases and other components of the environment are configured on the same platform. The reason for this is the extended waiting time of the transmitted data between the components of the environment.

## 5 Bug fixes and improvements

### 5.1. Incomplete query text in the repository

In the latest version, we fixed the problem that truncated query texts in the repository. The problem occurred in some cases with literal queries in which e.g. Polish or special characters were used. The problem was writing incomplete query content to the repository.

The fix fixes the problem, and a security has been introduced that will collect the query text again and save it in the repository. The update of the query texts will be performed during the application update.

**Note that the process of re-downloading the query texts may extend the application update process.**

### 5.2. Implementation of TLS1.1 and TLS1.2 support

Support for TLS 1.1 and TLS1.2 (Transport Layer Security) has been added to the latest version of the application. The latest version of the application has been coded in .Net 4.7.2.

### 5.3. Query Advisor and SQL Parser update

In the latest version of SQL Parser, we have introduced a number of improvements to improve the parsing and highlighting of query objects on the Show Plan Objects screen.

Another version of the Query Advisor mechanism has also been published, i.e. a functionality that verifies query performance and suggests performance tips for analyzed queries.