

# Micro Focus Operations Bridge Database Toolkit

Overview of the utilities being made available to aid in highlighting database configuration issues in relation to Operations Bridge Manager

Date: August 2020  
Author: Andy Doran ([andy.doran@microfocus.com](mailto:andy.doran@microfocus.com))  
Todd Stein ([todd.stein@microfocus.com](mailto:todd.stein@microfocus.com))  
Revision: 2.2 (Aug-2020)  
Location: <https://marketplace.microfocus.com/itom/content/opsbridge-database-toolkit>  
Support for: Operations Bridge Manager Classic 10.12, 10.60, 10.61, 10.62, 10.63, 10.70, 2018.05, 2018.11, 2019.02, 2019.05, 2019.11, 2020.05

The OpsBridge Database Toolkit helps with the configuration and maintenance of the databases used by the Operations Bridge Manager and Suite. This document describes the use of the cross-platform scripts provided as part of that toolkit

## Contents

Overview of the Utility .....	3
Current Checks .....	3
Requirements .....	3
Using the Utility .....	5
Using run-checks.pl .....	6
Examples .....	8
Check the database server configuration on an existing DPS/Gateway server .....	8
Check a Database Server Prior to OpsBridge Install .....	8
Using OAPerl.bat .....	10

## Overview of the Utility

### Current Checks

The Database Toolkit cross platform scripts make the following checks:

Server Type	Checks	Updates Supported?
<b>Oracle</b>	Server settings for Small/Medium/Large configuration	No
<b>Postgres</b>	Server settings for Small/Medium/Large configuration	yes
<b>MS SQL Server</b>	Database index fragmentation, tempdb configuration and filesystem file fragmentation	no
<b>DPS/Gateway</b>	System Memory and Disk space for Small/Medium/Large configuration	no

The scripts support remote connections to the database servers. However, the Data Processing and Gateway Server checks are made on the local system (the system where the script runs). If this system, already has one or both of these roles installed, the existing configuration will be used when the scripts run.

**Note:** For the online database checks to be made, Java needs to be available on the system that is running the checker tool. If Java is not available, the database checks will not be made.

### Requirements

The scripts are all written in Perl. On Linux, this means that the Perl module must be installed. This is a pre-requisite for installing the OBM server components, and so should be present. To install the Perl module if required, refer to the documentation for the Linux platform being used. For example, on RedHat:

```
yum install perl -y
```

On Windows, Perl will need to be installed unless the OpsBridge Operations Agent is installed (the OA provides a Perl engine and if this is detected can be used by the scripts). If the scripts are to be executed from a Windows system without an OA present (for example when running to process offline files), then a Perl engine such as ActivePerl must be installed. This is available from:

<https://www.activestate.com/products/perl/downloads/>

If the OA Perl engine is to be used, a batch file (oaperl.bat) is provided which will locate the Perl engine. Usage is described in the next section.

## Using the Utility

The Database Toolkit scripts are provided as a zip file which should be unzipped either to Windows or Linux. Once unzipped, the script that drives the process is called “run-checks.pl”. To execute this script on Linux, simply use:

```
./run-checks.pl <options>
```

Assuming it is in the current directory – specify the path to the file if it is not. To execute the script on Windows, either use:

```
perl run-checks.pl <options>
```

Or

```
oaperl.bat run-checks.pl <options>
```

Again, this assumes the script is in the current directory – specify the path if it is not. The first example above assumes that ActivePerl (or another engine) has been installed. The second example assumes that the OA Perl engine will be used. The batch file “oaperl.bat” locates the OA Perl engine and invokes the script, passing the options to it.

Example:

```
+++++
Operations Bridge Database Toolkit
=====
Utility:      run-checks.pl (Version: 2.0)
Description:  Run database and server checks in preparation for installation or upgrade of OpsBridge
              to Version 2020.05

Common module: Version: 2.2
Additional:    BSM Module Version: 2.1, Database Module Version: 2.1
Running on:   obux.admgt.co.uk (linux Version: 3.10.0-693.17.1.el7.x86_64)
Starting at:  Fri Aug 21 15:49:09 2020
+++++

[INFO] Operations Bridge is installed on this server
[INFO] Performing Database Server checks ...
[INFO] Checking Postgresql server localhost on port 5433 (connecting as user postgres) ...
[OK] Database Server version: 9.6.15 is supported by OBM Version 2020.05
[INFO] Checking server settings...

Parameter      Current Setting  Recommended (Large Depl.)  Description
=====
shared_buffers  131072 (1GB)      524288 (4 GB)              Sets the number of shared memory buffers used by the server.

Are you sure that you wish to make the above changes (y/n)? n
[INFO] No changes will be made

[INFO] Performing OBM Server system checks ...
[INFO] Memory Considerations for Large config (DPS and Gateway role):

System Memory  Free  Minimum  Recommended  Explanation
=====
19 GB          6 GB  40 GB      52 GB        Add more system memory to support this configuration

[INFO] Disk Free Space Checks:

Location  Capacity  Free Space  Recommended Free  Minimum Free  Explanation  Information
=====
/          50 GB    29 GB       73 GB             47 GB         Below Minimum  Installation location (/opt/OV)
Temp location (/tmp)
BSM Home (/opt/HP/BSM)
Data location (/var)

[ERROR] One or more of the device locations shown do not have the enough free space for the installation of OBM Version 2020.05
```

## Using run-checks.pl

The Database Toolkit script run-checks.pl acts as a wrapper for two additional scripts (db-server-checks.pl and obm-server-checks.pl). These can each be directly invoked, but the wrapper script will drive both and read any existing configuration as it does so.

The list of supported input parameters is as follows (and can be specified in any order):

Option	Explanation	Example
<b>-ignore</b>	Use this switch to ignore the current OpsBridge configuration if running on a DPS or Gateway server (doing so means that the database server details must be provided to perform database checks)	-ignore
<b>-nodb</b>	If this switch is used, then the database checks will not be made	-nodb
<b>-nosys</b>	If this switch is used, the system memory and disk space checks will not be made	-nosys
<b>-size &lt;size&gt;</b>	Use this argument to specify the configuration size (Small/Medium/Large) that will be used in the database and system checks	-size medium
<b>-server &lt;server&gt;</b>	The server name for the database server. If the tests are being executed from a DPS or Gateway server, this argument is ignored unless the “-ignore” switch has been used	-server DBSERVER
<b>-port &lt;port&gt;</b>	The port used for the database connection. This argument will override the configured settings if the script runs on a DPS or Gateway server	-port 5433
<b>-user &lt;user&gt;</b>	The database user for connecting to the database server. This argument will override the configured settings if the script runs on a DPS or Gateway server	-user sa
<b>-pwd &lt;password&gt;</b>	The password for the specified user. This is a required parameter (unless running on Windows with a trusted connection)	-pwd P@sswOrd
<b>-trusted</b>	If the script runs on Windows and the database server is a MS SQL Server, Windows authentication can be used instead of a SQL Login. Using this switch will override the user (whether specified in the configuration or as an input arguments)	-trusted
<b>-sid &lt;SID&gt;</b>	For Oracle connections, a SID is required. However, this argument is only required if the script is not running from a DPS or Gateway server as the information will be picked up from the configuration	-sid ORCL
<b>-dbtype &lt;TYPE&gt;</b>	This argument is required if the script is not running from an existing DPS or Gateway server (or the -ignore switch is used). It is used to specify the database server type (MSSQL, Oracle or Postgres)	-dbtype oracle

<b>-set</b>	This switch will enable the database changes to be made if the Database server is Postgres and there are parameters that are not configured correctly. There will be a prompt to confirm the changes before they are made.	<b>-set</b>
<b>-force</b>	Use this switch to disable the prompt that is provided with the -set switch (use with caution)	<b>-force</b>
<b>-frag &lt;percent&gt;</b>	The database checks with SQL Server will check for index fragmentation of 30% or higher. Use this argument to change that threshold	<b>-frag 50</b>
<b>-type &lt;type&gt;</b>	For the system checks, use this argument to specify the server type (DPS, Gateway or All)	<b>-type DPS</b>
<b>-nodisk</b>	Use this switch to disable the disk space checks	<b>-nodisk</b>
<b>-nosys</b>	Use this switch to disable the memory checks	<b>-nomem</b>

The script will always perform the system (memory and disk space) checks on the local system where the script runs. If that system already has a DPS or Gateway Server installation, the configuration for that install is used in the scripts unless the “-ignore” switch is used. That configuration information is:

- Database Server – The database server host, Database type and logon credentials (not the password)
- Server – the target location for the installation (on Windows these locations can be set at installation time)

This means that on a DPS or Gateway server, the script can be run by passing just one parameter – the logon password for the database. Optionally, some of the database parameters can be overridden (this can be important as the user configured for OBM use may not have the rights to view or set the server settings, so whilst the host and database information may be correct – the user will need to be an admin user). The parameters that override the configuration settings are:-

- -user – the Logon user
- -trusted – in the same way as the -user switch overrides the Logon user, for MS SQL using a trusted connection means that the current Windows account will be used to connect
- -port – the connection port can be overridden

If the script is not running on a DPS or Gateway Server, then the database information – all related parameters – must be supplied, or the “-nodb” switch used. In terms of the system checks, these are always made on the system running the script. If this is not currently a DPS or Gateway server then the checks made depend on whether the system is Windows or Linux:

Windows – all local disks will be checked

Linux – the filesystems /opt, /var and /tmp will be checked

Using the -ignore switch has the same effect as running the script on a server that has not got the DPS or Gateway role.

In addition, the following switches can be used:

Option	Explanation	Example
<b>-nocolor</b>	The output uses colors. For example, “OK” messages are in <b>GREEN</b> whilst failure messages are in <b>RED</b> . This switch disables the use of colors if required	-nocolor
<b>-log</b>	This switch enables logging. All output sent to the screen is also captured in a log file. A log is created for each utility, will be in the current directory and the name is given as each utility completes	-log
<b>-timeout &lt;seconds&gt;</b>	The default timeout for commands issued by the scripts is 5 minutes. This can be overridden using the -timeout switch to either lengthen or shorten that time period. This only applies when the scripts run on Linux	-timeout 120
<b>-help</b>	Prints help text. If this is provided, then ONLY the help is shown – the script will immediately exit	-help

## Examples

Check the database server configuration on an existing DPS/Gateway server

To simply check the current OBM configuration Database Server, and not check the system itself, use:

```
./run-checks.pl -pwd P@ssw0rd1 -nosys
```

## Check a Database Server Prior to OpsBridge Install

You can verify that the database server being used to host the OpsBridge OBM databases has been correctly configured before installing any OBM components:

```
./run-checks.pl -server OBMDB -user postgres -pwd P@ssw0rd -dbtype postgres -nosys
```

In this case, when the port is not specified it will default to the standard port for the database type (in this case with Postgres the port will be 5432. By default, the settings for a “Small” configuration will be used. The results will be like:



```

+++++
Operations Bridge Database Toolkit
=====
Utility:      run-checks.pl (Version: 2.0)
Description:  Run database and server checks in preparation for installation or upgrade of OpsBridge
              to Version 2020.05

Common module: Version: 2.2
Additional:    BSM Module Version: 2.1, Database Module Version: 2.1
Running on:    obux.admgt.co.uk (linux Version: 3.10.0-693.17.1.el7.x86_64)
Starting at:    Fri Aug 21 16:40:51 2020
+++++

[INFO] Operations Bridge is installed on this server
[INFO] The OBM Configuration information will be ignored
[INFO] Performing Database Server checks ...
[INFO] Checking Postgresql server VIMES on port 5432 (connecting as user postgres) ...
[OK] Database Server version: 12.1, is supported by OBM Version 2020.05
[INFO] Checking server settings...

Parameter      Current Setting  Recommended (Small Depl.)  Description
=====
autovacuum_analyze_threshold  50              5000                      Minimum number of tuple inserts, updates, or deletes prior to analyze.
autovacuum_vacuum_threshold   50              5000                      Minimum number of tuple updates or deletes prior to vacuum.
checkpoint_completion_target  0.5             0.9                      Time spent flushing dirty buffers during checkpoint, as fraction of checkpoint interval.
checkpoint_timeout            300 (5min)      600 (10 min)             Sets the maximum time between automatic WAL checkpoints.
commit_delay                  0              500                      Sets the delay in microseconds between transaction commit and flushing WAL to disk.
log_min_duration_statement    -1             3000                     Sets the minimum execution time above which statements will be logged.
maintenance_work_mem         65536 (64MB)    131072 (128 MB)          Sets the maximum memory to be used for maintenance operations.
max_connections               100            1000                     Sets the maximum number of concurrent connections.
max_locks_per_transaction     64             512                      Sets the maximum number of locks per transaction.
shared_buffers                16384 (128MB)   65536 (512 MB)           Sets the number of shared memory buffers used by the server.
work_mem                      4096 (4MB)      25600 (25 MB)            Sets the maximum memory to be used for query workspaces.

[INFO] Run the utility again with the "-set" switch in order to update these parameters, or work with the dba to make changes

```

In this example, the Database Server version is Postgres V12.1 and there are a number of settings that should be changed. Running the same command, but this time using the “-set” switch will give:

```

Parameter      Current Setting  Recommended (Small Depl.)  Description
=====
autovacuum_analyze_threshold  50              5000                      Minimum number of tuple inserts, updates, or deletes prior to analyze.
autovacuum_vacuum_threshold   50              5000                      Minimum number of tuple updates or deletes prior to vacuum.
checkpoint_completion_target  0.5             0.9                      Time spent flushing dirty buffers during checkpoint, as fraction of checkpoint interval.
checkpoint_timeout            300 (5min)      600 (10 min)             Sets the maximum time between automatic WAL checkpoints.
commit_delay                  0              500                      Sets the delay in microseconds between transaction commit and flushing WAL to disk.
log_min_duration_statement    -1             3000                     Sets the minimum execution time above which statements will be logged.
maintenance_work_mem         65536 (64MB)    131072 (128 MB)          Sets the maximum memory to be used for maintenance operations.
max_connections               100            1000                     Sets the maximum number of concurrent connections.
max_locks_per_transaction     64             512                      Sets the maximum number of locks per transaction.
shared_buffers                16384 (128MB)   65536 (512 MB)           Sets the number of shared memory buffers used by the server.
work_mem                      4096 (4MB)      25600 (25 MB)            Sets the maximum memory to be used for query workspaces.

Are you sure that you wish to make the above changes (y/n)?

```

Here there is a prompt to make the changes. Accepting will then make the changes and display the settings again:

```

[INFO] Making identified configuration changes to Postgresql server...
[INFO] Reloading updated parameters...
[INFO] Configuration reloaded

Checking parameters after updates...

Parameter      Current Setting  Recommended (Small Depl.)  Restart Required
=====
autovacuum_analyze_threshold  5000            5000                      No
autovacuum_vacuum_threshold   5000            5000                      No
checkpoint_completion_target  0.9             0.9                      No
checkpoint_timeout            600 (10min)     600 (10 min)             No
commit_delay                  500             500                      No
log_min_duration_statement    3000 (3s)       3000 (3 s)               No
maintenance_work_mem         131072 (128MB)  131072 (128 MB)          No
max_connections               100            1000                     No
max_locks_per_transaction     64             512                      No
shared_buffers                16384 (128MB)   65536 (512 MB)           No
work_mem                      25600 (25MB)    25600 (25 MB)            No

[INFO] If any parameters show that a restart is required then Postgresql needs to be restarted prior to them taking effect. Be sure to shut down OBM before doing this

```

The display this time is slightly different – the changes that were made and are now in force are displayed in Green, but some changes are displayed in Yellow with an indication that a Postgres server restart is required for them to take effect. The script does not restart Postgres, so this will need to be done manually.

For Oracle, any required changes will not be made and the recommendation is to discuss making updates with the relevant dba.

## Using OAPerl.bat

On Windows, perl is not available by default. The scripts require perl to be present to run these scripts. If the OA agent has been installed (for example as part of the installation of the DPS or Gateway role) then perl is provided but is not on the path.

The batch file “oaperl.bat” can be used to locate the necessary perl engine and use it to run the scripts. An example of doing this is:

```
C:\dbtools\> oaperl.bat run-checks.pl -pwd P@ssw0rd1 -nosys
```

Here it is assumed that the tools have been unpacked into the “C:\dbtools” directory. If the OA Agent is not present (for example, the tool is running on a laptop) then the tool engine must be installed manually as described earlier in the document. It is also important to ensure that Java is available (again – this will be part of an OA agent install but may not be available if the OA agent has not been installed).

Java can be installed from a download here:

<https://java.com/en/download/win10.jsp>