Configure Microsoft SQL Server Collectors

On this page:

To monitor Microsoft SQL Server with Database Visibility, you must be running the 2005 version or newer.

Connection Details

Field	Description
Database Type	The database type that you want to monitor.
Database Agent	The Database Agent that manages the collector.
Name	The name you want to identify the collector by.
Hostname or IP Address	The hostname or IP address of the machine that your database is running on.
Failover Partner	The hostname or IP address of the failover partner.
Listener Port	The TCP/IP address of the port on which your database communicates with the Database Agent
Custom JDBC Connection String	The JDBC connection string generated by the database agent, for example, `jdbc:sqlserver://`. You can also specify a custom connection string, which is useful for setting custom authentication options.
Window Authentication	Click to enable Windows authentication when connecting to the database.
Username	The name of the user who is connecting to and monitoring the database through the Database Agent. The user should have the permissions described in User Permissions for Microsoft SQL Server.
Password	The password of the user who is connecting to and monitoring the database through the Database Agent.
CyberArk	Click to enable CyberArk for database username and password. When CyberArk is enabled, information about Safe, Folder, and Object is required to fetch the username and password for your database. To use CyberArk with Database Visibility, you must download the JavaPasswordSDK.jar file from the CyberArk web site and rename the file to cyberark-sdk-9.5.jar. Then, you must copy the JAR file to the lib directory of the database agent zip file.
Exclude Databases *(New in 4.3.4.1)*	The databases that you want to exclude, separated by commas.

Logging Enabled	Click to enable verbose mode logging, which logs all communications between the Controller and the Collector. Enable only during troubleshooting because logging consumes a lot of disk space. If you have enabled logging, you can click the logging icon in the Log column of the Collector Administration window to view the log file. The log files are located in the <db_agent_home>\agent directory and have the format <CollectorName>_out.log and <CollectorName>_err.log.

User Permissions for Microsoft SQL Server

The user account used for monitoring can be a Windows authenticated account (if the Database Agent is running on Windows) or SQL Server authenticated (if AppDynamics Database Visibility is running on Windows or Linux).

Required Permissions to See Execution Plans

The SQL Server user, specified in the Create Collector > Connection Details section must be a SQL Server Authenticated user that is a member of the sysadmin server role or a Windows Authenticated Account with SHOWPLAN access on each database.

For more information, see Showplan Security and SHOWPLAN Permission and Transact-SQL Batches in the SQL Server documentation.

Minimum Permissions Required for SQL Server Logon

You can use the procedure below to create a SQL Server user with the minimum permissions required.

Use the following to create a SQL Server logon user that provides the minimal level of permissions required in order to gain full AppDynamics Database Visibility/SQL Server functionality.

1. Using SQL Server Management Studio, create a new login for the AppDynamics SQL Server Database Collector, such as DBMon_Agent_User.
2. From the User Mapping tab, map the new user to the master and msdb databases.

   > **Viewing Object Information**
   >
   > To view object information on the Database > Objects Browser, map the monitoring user to the databases of interest.

3. Once you have created the login, give the following privileges to the user, substituting DBMon_Agent_User with the name you specified on the Login - New window:

Note: You can execute the following as a batch from a query window in Management Studio. The example shows grants to DBMon_Agent_User; remember to change this if you have set up a different login.

```
use master
GRANT VIEW ANY DATABASE TO DBMon_Agent_User;
GRANT VIEW ANY definition to DBMon_Agent_User;
GRANT VIEW server state to DBMon_Agent_User;
GRANT SELECT ON [sys].[sysaltfiles] TO DBMon_Agent_User
GRANT execute on sp_helplogins to DBMon_Agent_User
GRANT execute on sp_readErrorLog to DBMon_Agent_User
```

```
use msdb
GRANT SELECT on dbo.sysjobsteps  TO DBMon_Agent_User
GRANT SELECT on dbo.sysjobs  TO DBMon_Agent_User
GRANT SELECT on dbo.sysjobhistory  TO DBMon_Agent_User
```

where DBMon_Agent_User is the name of the SQL Server user account specified in Create New Collector, Connection Details, Username field.

SQL Server Authentication

If you are running AppDynamics Database Visibility on Linux then you must use SQL Server authentication.

If your SQL Server database allows mixed-mode authentication, then the SQL Server AppDynamics Database Visibility uses to monitor the SQL Server database can use a SQL Server username/password authenticated account. If you would like to lock the role/permissions for the account down, then the account running AppDynamics Database Visibility requires:

- View any database
- View any definition
- View server state

One additional requirement for I/O monitoring is to give permissions on a System view called sys.sysaltfiles. To do this you need to select the **master database > Views > System Views > Properties** for sys.sysaltfiles and then give select permissions on the object to the Public role.

Windows Authentication

If you would like to use a Windows authenticated account to connect to the SQL Server database, the following is required:

- When creating the collector from the Create New Collector dialog, do not specify Username and Password in the database Connection Details.
- Also, the agent must be started with the path to its authentication library. For more information see, Windows Authentication for Microsoft SQL Server.

Access Rights

If you choose not to grant View Server State permissions, then you must grant permissions individually for the following objects in order to monitor SQL Server:

```
GRANT execute on xp_msver to DBMon_Agent_User
GRANT SELECT on sys.dm_exec_requests to DBMon_Agent_User
GRANT SELECT on sys.dm_exec_sessions to DBMon_Agent_User
GRANT SELECT on sys.dm_os_performance_counters to DBMon_Agent_User
GRANT SELECT on sys.dm_exec_query_stats to DBMon_Agent_User
GRANT SELECT on sys.fn_virtualfilestats to DBMon_Agent_User
GRANT SELECT on sys.sysaltfiles to DBMon_Agent_User
GRANT SELECT on sys.configurations to DBMon_Agent_User
GRANT SELECT on sys.dm_exec_sql_text to DBMon_Agent_User
GRANT SELECT on sys.sysperfinfo to DBMon_Agent_User
GRANT SELECT on sys.sysprocesses to DBMon_Agent_User
GRANT SELECT on sys.syscurconfigs to DBMon_Agent_User
GRANT SELECT on sys.fn_get_sql to DBMon_Agent_User
GRANT SELECT on sys.partitions to DBMon_Agent_User;
GRANT SELECT on sys.objects to DBMon_Agent_User;
GRANT SELECT on sys.indexes to DBMon_Agent_User;
GRANT SELECT on sys.tables to DBMon_Agent_User;
GRANT SELECT on sys.dm_db_database_page_allocations to DBMon_Agent_User;
GRANT SELECT on master.sys.dm_exec_procedure_stats to DBMon_Agent_User;
GRANT SELECT on sys.dm_os_ring_buffers to DBMon_Agent_User;
GRANT SELECT on sys.dm_os_sys_memory to DBMon_Agent_User;
GRANT SELECT on sys.master_files to DBMon_Agent_User;
GRANT SELECT on sys.dm_io_virtual_file_stats to DBMon_Agent_User;
GRANT SELECT on sys.dm_exec_query_plan to DBMon_Agent_User;
GRANT SELECT on sys.dm_exec_text_query_plan to DBMon_Agent_User;
GRANT SELECT on sys.syscolumns to DBMon_Agent_User;
GRANT SELECT on sp_spaceused to DBMon_Agent_User;
GRANT SELECT on sys.sysusers to DBMon_Agent_User;
GRANT SELECT on master.dbo.sysconfigures to DBMon_Agent_User;
GRANT SELECT on msdb.dbo.sysjobhistory to DBMon_Agent_User;
GRANT SELECT on sys.sysdatabases to DBMon_Agent_User;
```