



Microsoft®

System Center Operations Manager

Guide to Microsoft System Center Management Pack for SQL Server 2017+

Microsoft Corporation

Published: June 2018

The Operations Manager team encourages you to provide any feedback on the management pack by sending it to sqlmpsfeedback@microsoft.com.

Copyright

This document is provided "as is". Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

© 2018 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Windows, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are the property of their respective owners.

Contents

Guide to Microsoft System Center Management Pack for SQL Server 2017+	4
Changes History	4
Get Started	9
What's New?	9
Supported Configurations	10
Management Pack Scope	10
Prerequisites	11
Files in This Management Pack	12
Monitoring Configuration	13
Management Pack Purpose	28
Monitoring Types	29
Monitoring Scenarios	32
How Health Rolls Up	33
Configure the Management Pack	34
Mandatory Configuration	34
Best Practice: Create a Management Pack for Customizations	35
How to Enable Agent Proxy Option	36
How to Import a Management Pack	36
Security Configuration	36
View Information in the Operations Manager Console	59
Version-Independent (Generic) Views and Dashboards	59
SQL Server Views	61
SQL Server Reporting	62
Appendix: Known Issues and Troubleshooting	66

Guide to Microsoft System Center Management Pack for SQL Server 2017+

This guide is based on version 7.0.7.0 of the Management Pack for Microsoft SQL Server 2017+.

Changes History

Release Date	Changes
June 2018 (version 7.0.7.0 RTM)	<ul style="list-style-type: none">• Fixed issue: "The agent is suspect. No response within last minutes" alerting rule does not catch appropriate events due to the wrong source
June 2018 (version 7.0.5.0 CTP)	<ul style="list-style-type: none">• Implemented an ability to monitor SQL Server Cluster instances locally; formerly, it was possible only in Agentless and Mixed modes• Added the SSIS monitoring• Added the "Exclude List" property in DB Engine Discovery in order to filter instances, which are not subject to monitoring• Added the "Exclude List" property in DB Discovery in order to filter databases, which are not subject to monitoring• Implemented a feature: both "Exclude List" properties support usage of the asterisk character to make the filter more flexible; e.g. "**temp" is used to exclude instances/databases ending with "temp" and having anything in the beginning• Added the "Computers" view• Added the "ClusterName" property to the AG class and updated AG alerts in order to display the property within• Updated the "SP Compliance" monitor in order to support the Modern Servicing Model for SQL Server: the monitor will check build number instead of Service Pack number• Updated the "SPN Status" monitor so that it requires only a single SPN record when only TCP/IP is enabled and the instance is the default one• Updated the "Database Backup Status" monitor: it is disabled by default now

Release Date	Changes
	<ul style="list-style-type: none"> • Updated the DB Space monitors so that their alert descriptions include the actual value of space available • Updated the "Configuration Security" section in the guide • Fixed issue: the "Database Health Policy" monitor ignores the "Critical" state (on Windows only) • Fixed issue: the "Alert severity" property of the "DB File Free Space Left" monitor has incorrect Default Value • Fixed issue: the "DB Filegroup Fx Container" rollup monitor has an alert parameter with a wrong value within • Fixed issue: "Resource Pool Memory consumption" monitor may not change its state to "Critical" for the "default" resource pool • Fixed issue: "Number of Samples" parameter of "Resource Pool Memory consumption" alert displays incorrect data • Fixed issue: missed image resources in the SQL Server 2017+ Core Library
November 2017 (version 7.0.0.0 RTM)	<ul style="list-style-type: none"> • Introduced a number of improvements and bug fixes.
October 2017 (version 6.7.65.0 RC1)	<ul style="list-style-type: none"> • The management pack was reimplemented in order to enable monitoring SQL Server 2017 and all upcoming SQL Server versions • Reduced the number of files in the management pack • Removed dependency on SQL Server Dashboards management pack • Introduced a number of fixes and improvements to functionality, performance, and display strings
June 2017 (version 6.7.60.0 RC0)	<ul style="list-style-type: none"> • Implemented Always On monitoring on Windows and Linux • Implemented Disk Latency workflows • Added new "Login failed" alerting rule for SQL Server event #18456 • Added support for AD Credentials in Agentless Mode on Windows • Fixed issue: different file location info from "sys.master_files" and "sysfiles" causes error when Availability Group secondary database files are in different path • Fixed issue: workflows cannot connect to an instance when only Shared Memory protocol is enabled

Release Date	Changes
	<ul style="list-style-type: none"> • Fixed issue: Previously deleted SQL Server on Windows/Linux custom user policies do not disappear in SCOM • Introduced a number of improvements to the management pack
May 2017 (version 6.7.55.0 CTP3)	<ul style="list-style-type: none"> • Always On monitoring for Windows: implemented basic workflows similar to SQL Server 2016 management pack • SQL Cluster is now supported in Mixed monitoring mode (it used to be supported in Agentless monitoring mode only) • The management pack can now use SQL Server authentication credentials in Agent / Mixed monitoring modes (previously, SQL credentials could be used in Agentless monitoring mode only) • Implemented tasks execution • Implemented monitoring of SQL Server Agent • Implemented monitoring of Resource Pools • Added several monitors and performance rules for Memory-Optimized Tables monitoring • Implemented monitoring of Custom User Policies • Improved SQL queries in the database discovery • Fixed issue: "Failed to replace parameter while creating the alert for monitor state change" warnings for workflows of Memory-Optimized Tables containers • Fixed log reader issues • Fixed "SPN Configuration" issue • Updated and fixed the Knowledge Base articles and display strings
February 2017 (version 6.7.40.0 CTP2)	<ul style="list-style-type: none"> • Implemented "Discovery Data Mapper"; improved queries and datasources • Implemented support for full cookdown for all discoveries on Linux and Windows • Implemented Log Shipping monitoring • Implemented new monitors and rules: <ul style="list-style-type: none"> ○ "Service Pack Compliance" monitor ○ "SQL Server Windows Service" monitor ○ "CPU Utilization (%)" monitor ○ "SQL Server Service (database)" monitor ○ "Database Health Policy (Critical)" monitor ○ "Database Health Policy (Warning)" monitor

Release Date	Changes
	<ul style="list-style-type: none"> ○ “WMI Health” monitor ○ “Memory Used By Tables (MB) rule ○ “Memory Used By Indexes (MB)” rule ○ “MSSQL on Windows: DB Engine CPU Utilization (%)” rule ○ “SQL Server DB Engine is restarted” rule ● Implemented new DB Space performance rules: <ul style="list-style-type: none"> ○ MSSQL: DB Allocated Space Unused (MB) ○ MSSQL: DB Free Space Total (%) ○ MSSQL: DB Free Outer Space (MB) ○ MSSQL: DB Transaction Log ○ Free Space Total (%) ○ MSSQL: DB Free Space Total (MB) ○ MSSQL: DB Allocated Space Used (MB) ○ MSSQL: DB Allocated Space (MB) ● Implemented new classes: <ul style="list-style-type: none"> ○ SQL Server DB FILESTREAM Filegroup on <Platform> ○ SQL Server DB Memory-Optimized Data Container on <Platform> ○ Generic SQL Server Custom User Policy ○ SQL Server Custom User Policy on <Platform> ○ SQL Server Database Critical Policy on <Platform> ○ SQL Server Database Warning Policy on <Platform> ○ SQL Server Resource Pool Group on <Platform> ○ Generic SQL Server Resource Pool ○ SQL Server Resource Pool on <Platform> ○ SQL Server Internal Resource Pool on <Platform> ○ SQL Server User Resource Pool on <Platform> ○ SQL Server User-Defined Resource Pool on <Platform> ● Implemented new monitor and rule for FILESTREAM objects: <ul style="list-style-type: none"> ○ MSSQL on <Platform>: DB FILESTREAM Filegroup Free Space Total (MB) ○ MSSQL on <Platform>: DB FILESTREAM Filegroup Free Space Total (%) ● Implemented new performance rules for Memory-Optimized Data filegroups: <ul style="list-style-type: none"> ○ MSSQL on Windows: DB Memory-Optimized Data Filegroup Free Space Total (MB)

Release Date	Changes
	<ul style="list-style-type: none"> ○ MSSQL on Windows: DB Memory-Optimized Data Filegroup Free Space Total (%) ● Implemented “MSSQL LogReader” module ● Implemented Event Collection monitoring for Linux and Windows (more than 400 workflows) ● Implemented discoveries, rollups, and icons for the new classes; updated Filegroup and child classes’ icons ● Implemented all XTP counters (more than 200 workflows) ● Implemented “Empty Bucket percent” in the hash index monitor ● Implemented “Average length of the row chains” in the hash buckets monitor ● Implemented “SQL Full-text Filter Daemon Launcher Service”; added “NetworkName” property to local dbengine; refactored Windows monitoring folder structure ● Added “Local Database” class on Windows ● Improved the architecture: split “Windows.DBEngine” and “Windows.LocalDBEngine” classes ● Improved error logging ● Improved the error handling (connectivity issues) ● Remounted “LocalDiscoverySeed” discovery to support long names and not support wow64 ● Updated Add Monitoring Wizard: fixed layouts issues, improved multithreading, and performance, implemented background loading progress ● Updated and fixed the Knowledge Base articles and display strings; unified the workflows naming template ● Fixed alerts for classes managed by local agent ● Fixed Smart Connect issues connected with cached data (WMI connection) ● Fixed Linux modules to skip smart connect ● Fixed issue: “Total Transactions Per Second” rule runs twice in one interval ● Fixed issue: SQL Server on Windows database objects may get rediscovered ● Fixed issue: filegroups get undiscovered in SQL Express instance ● Fixed issue: in some situations, Add Monitoring Wizard cannot detect that test connection task is completed

Release Date	Changes
	<ul style="list-style-type: none"> Fixed issue: the already discovered database objects are undiscovered if database state is changed to "Offline"
December 2016 (version 6.7.18.0 CTP1)	The original release of this management pack

Get Started

In this section:

- [What's New?](#)
- [Supported Configurations](#)
- [Management Pack Scope](#)
- [Prerequisites](#)
- [Files in This Management Pack](#)

What's New?

This management pack provides the following new features:

- This management pack is intended to monitor SQL Server 2017 and all upcoming SQL Server versions.
- Cross-platform operation: SQL monitoring is possible as on Windows, so as on Linux operating systems.
- Agentless and mixed monitoring modes are now available along with traditional agent monitoring. Please note that both agent monitoring and mixed monitoring are available for SQL Server on Windows only.
- Implemented a new Database Files Space Usage Forecast report. This report provides collection of the data regarding the space used by the database files along with the amount of free space left. In addition, this report provides a forecast on the space used by the database files. For more details, see [SQL Server Reporting](#) section.

Supported Configurations

The following table details the supported configurations for the management pack:

Configuration	Supported
Platforms	Windows Server 2012 Windows Server 2012 R2 Windows Server 2016 Ubuntu 16.04 x64 Red Hat Enterprise Linux 7.3 or 7.4 SUSE Linux Enterprise Server v12 SP2 Docker Engine 1.8+
SQL Server	SQL Server 2017
SCOM	System Center Operations Manager 2012 R2 System Center Operations Manager 2016
Clustered servers	Yes
Agent monitoring	On Windows OS only
Agentless monitoring	Yes
Virtual environment	Yes



Note

Localized (non-English) versions of SQL Server are not supported.

Upgrade from the previous version of the management pack is not supported. Please use a clean installation.

SMB fileshares are supported as a storage option. For more information, see [Description of support for network database files in SQL Server](#) article.

Management Pack Scope

Management Pack for Microsoft SQL Server enables the monitoring of the following features:

- SQL Server Database Engines (supported editions: Evaluation, Developer, Express)
- SQL Server Databases (including filegroups, data files, and transaction log files)

- SQL Server Always On Availability Groups
- SQL Server Memory-Optimized Tables (SQL Server In-Memory OLTP)

 **Important**

We recommend that you monitor no more than 50 databases and 150 database files per System Center Operations Manager agent to avoid spikes in CPU usage that may affect the performance of monitored servers. Note that [Distributed availability groups](#) are not supported by this management pack.

 **Note**

Please refer to “[Monitoring Scenarios](#)” section for a full list of monitoring scenarios supported by this management pack.

For more information and detailed instructions on setup and configuration, see “[Configure the Management Pack](#)” section of this guide.

Prerequisites

As a best practice, you should import the Windows or Linux Server Management Pack for the operating system you are using. The Server Management Packs monitor aspects of the operating system that influence the performance of computers running SQL Server, such as disk capacity, disk performance, memory utilization, network adapter utilization, and processor performance.

 **Note**

A dedicated Operations Manager management group is not required for this management pack. Installation of .NET Framework 4.5 or newer is required.

 **Important**

The following Linux workflows are temporary disabled because they are not provided with the necessary data by the SQL Server:

Rules:

- MSSQL on Linux: DB Memory-Optimized Data Filegroup Free Space Total (MB)
- MSSQL on Linux: DB Memory-Optimized Data Filegroup Free Space Total (%)
- MSSQL on Linux: DB FILESTREAM Filegroup Free Space Total (%)

- MSSQL on Linux: DB FILESTREAM Filegroup Free Space Total (MB)
- MSSQL on Linux: DB Filegroup Free Space Total (%)
- MSSQL on Linux: DB Filegroup Free Space Total (MB)
- MSSQL on Linux: DB Filegroup Allocated Free Space (%)
- MSSQL on Linux: DB Filegroup Allocated Free Space (MB)
- MSSQL on Linux: DB Free Outer Space (MB)
- MSSQL on Linux: DB Allocated Free Space (MB)
- MSSQL on Linux: DB Transaction Log Free Space Total (%)
- MSSQL on Linux: DB Allocated Space Used (MB)
- MSSQL on Linux: DB Free Space Total (%)
- MSSQL on Linux: DB Free Space Total (MB)
- MSSQL on Linux: DB Allocated Space (MB)

Monitors:

- DB Free Space Left
- DB Space Percentage Change
- Transaction Log Free Space (%)
- DB FILESTREAM Filegroup Free Space

Files in This Management Pack

The Management Pack for Microsoft SQL Server includes the following files:

- Microsoft.SQLServer.Core.Library.mp
- Microsoft.SQLServer.Core.Views.mp
- Microsoft.SQLServer.Visualization.Library.mpb
- Microsoft.SQLServer.Linux.Views.mp
- Microsoft.SQLServer.Linux.Discovery.mpb
- Microsoft.SQLServer.Linux.Monitoring.mpb
- Microsoft.SQLServer.Windows.Views.mpb
- Microsoft.SQLServer.Windows.Discovery.mpb
- Microsoft.SQLServer.Windows.Monitoring.mpb



Note

Along with the present guide, the Management Pack delivery also contains a guide to configuration and customization of the SQL Server dashboards and an appendix containing information about the management pack objects and workflows.

Monitoring Configuration

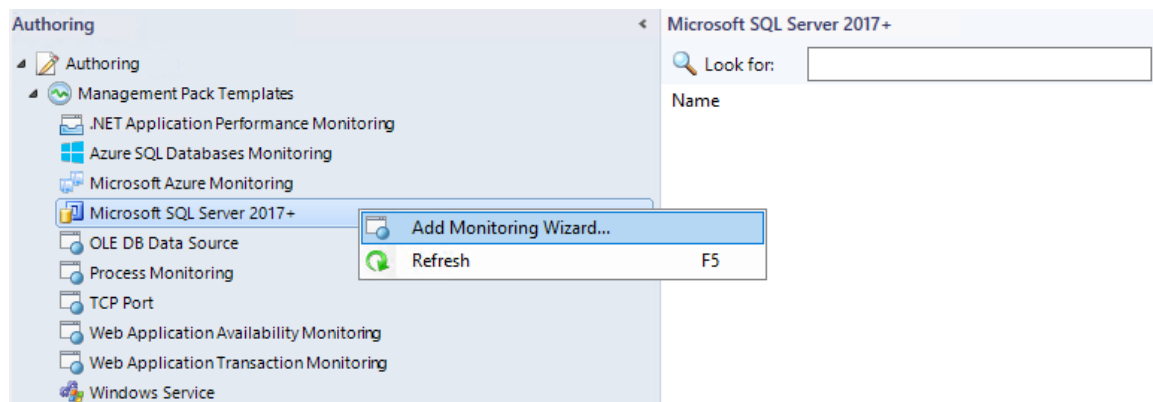
This management pack provides three monitoring types: local agent monitoring, agentless monitoring and mixed monitoring. This section provides guidance on configuring the monitoring by means of this management pack:

- [Configure Agentless Monitoring by Add Monitoring Wizard](#)
- [Enable Mixed Monitoring for SQL Server on Windows by Override](#)
- [Configure SQL Server Monitoring Pool](#)

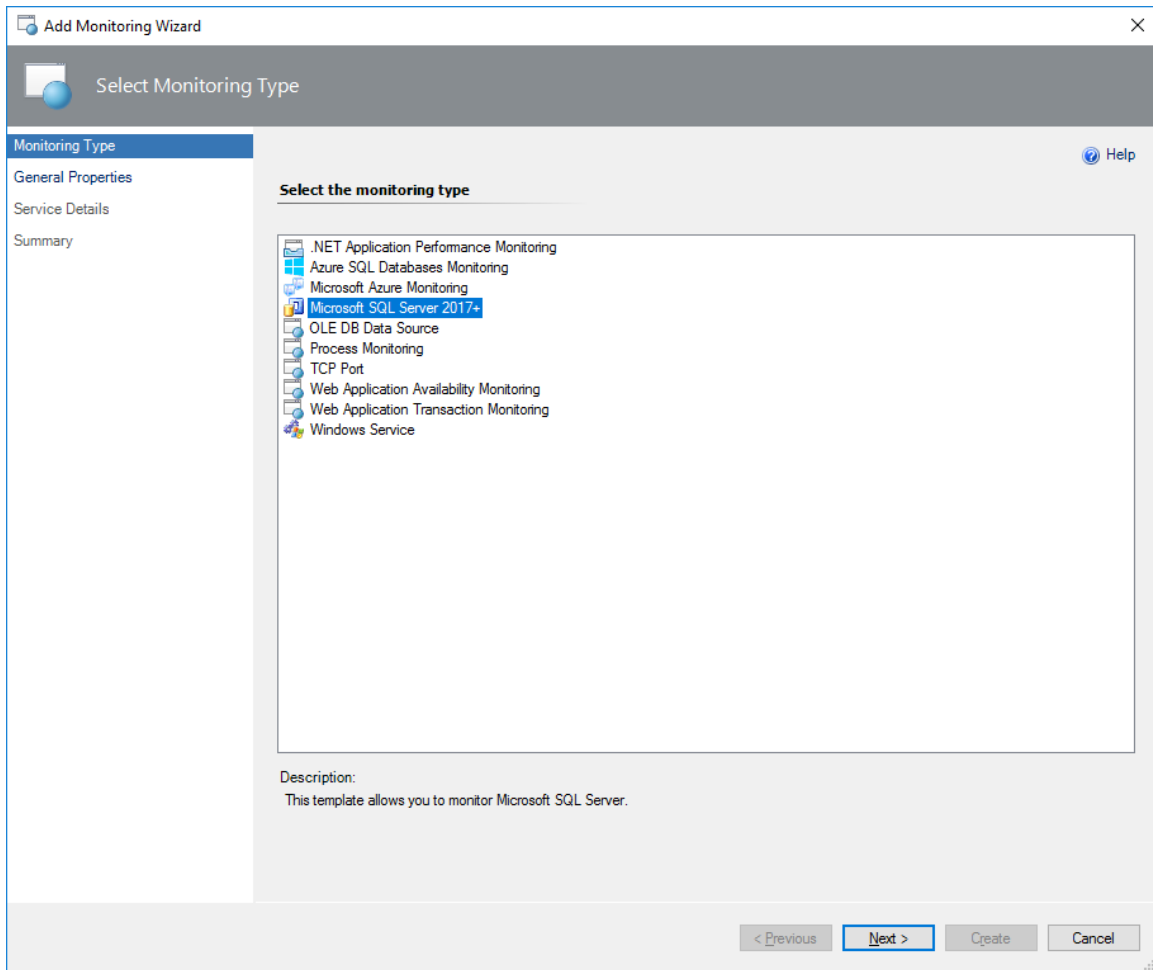
Configure Agentless Monitoring by Add Monitoring Wizard

To begin agentless monitoring of SQL Server instances, perform the following steps:

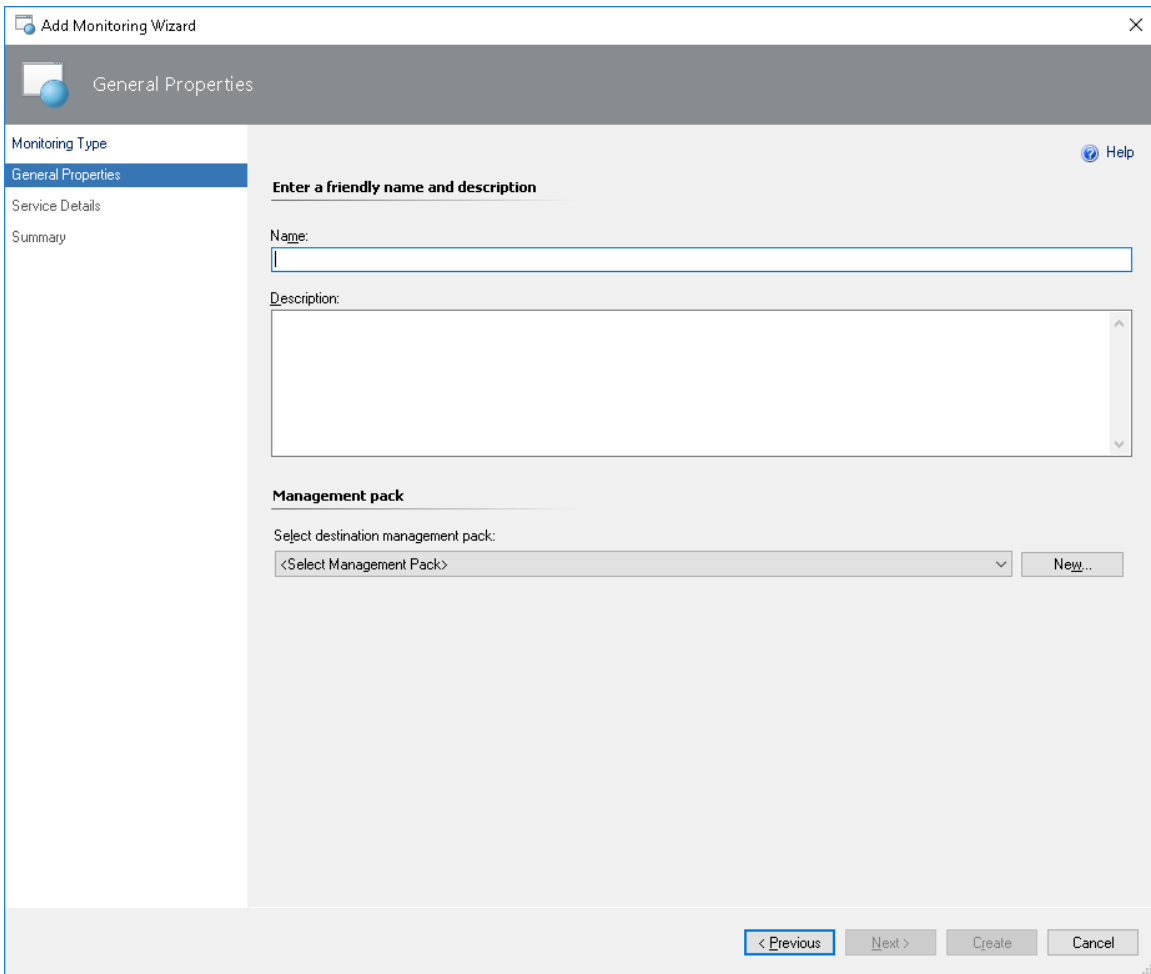
In the Operations Manager, navigate to **Authoring | Management Pack Templates**, right-click **Microsoft SQL Server 2017+** and select **Add Monitoring Wizard...**



In **Monitoring Type** window, select **Microsoft SQL Server 2017+** and click the **Next** button.



In **General Properties** window, you must provide your template with **Name** and **Description**, as well as **Select destination management pack** where the template will be stored.



You can also create a new destination management pack by clicking the **New...** button.

Create a Management Pack

General Properties

General Properties

Knowledge

Management Pack General Properties

ID :

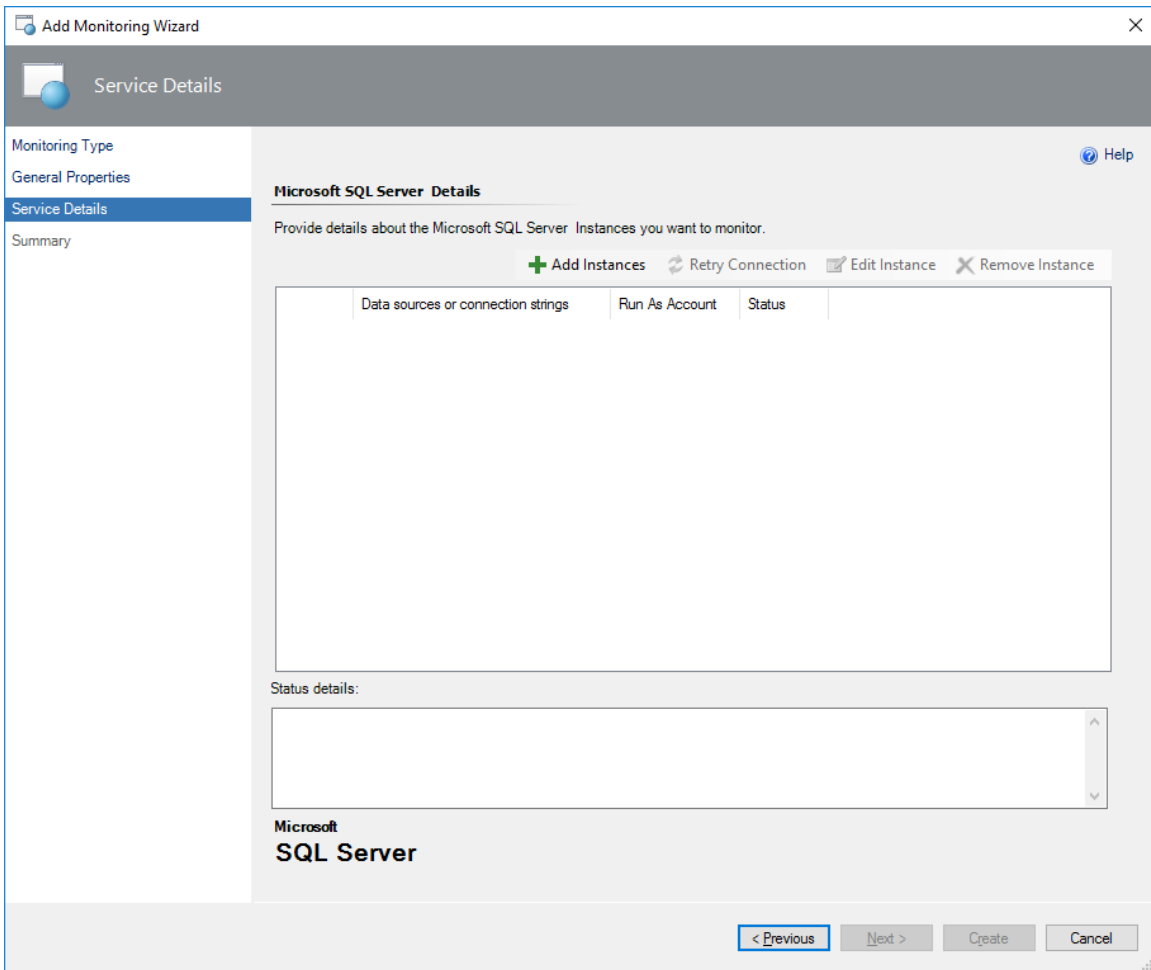
Name :

Version : 1.0.0.0 For example, 1.0.0.0

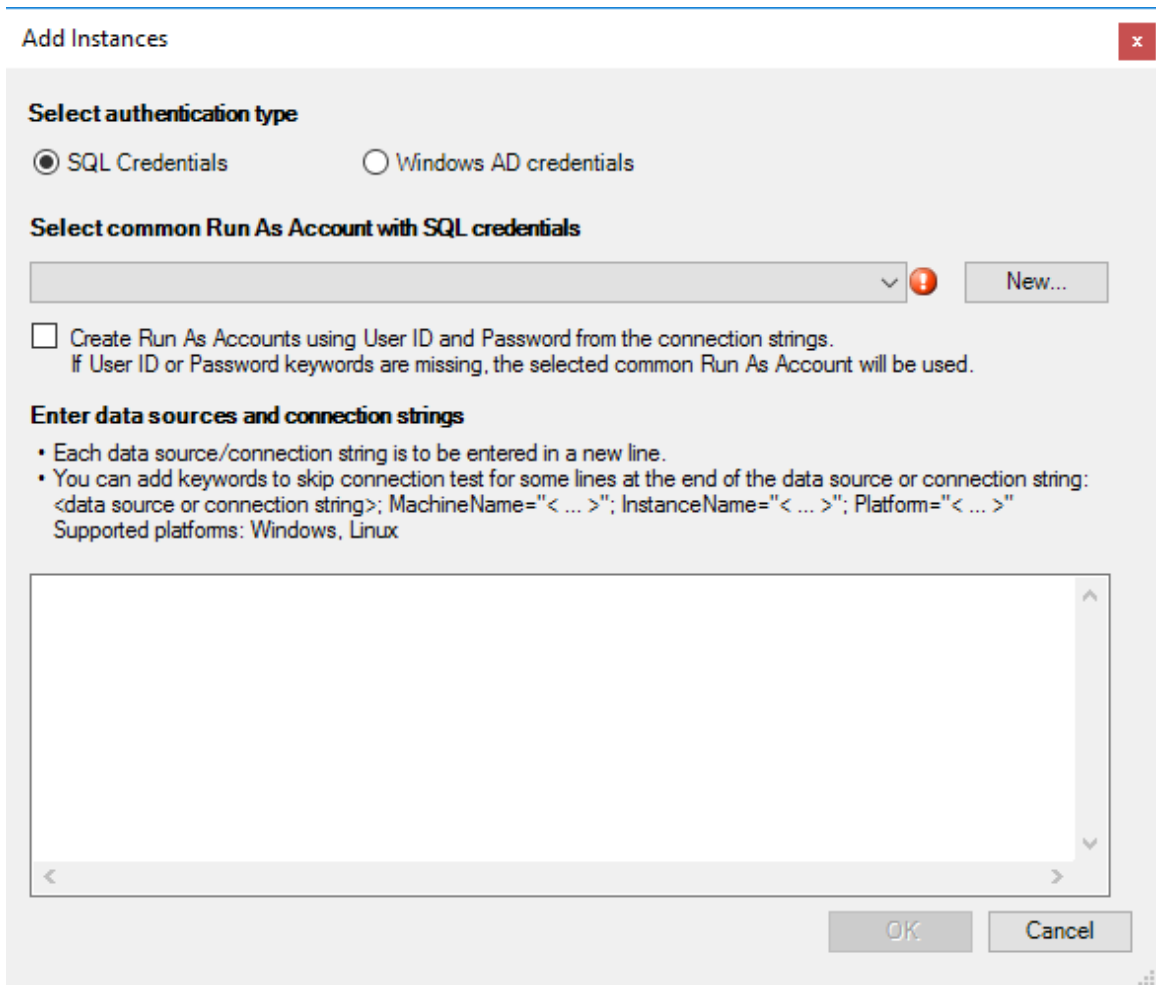
Description :

< Previous Next > Create Cancel

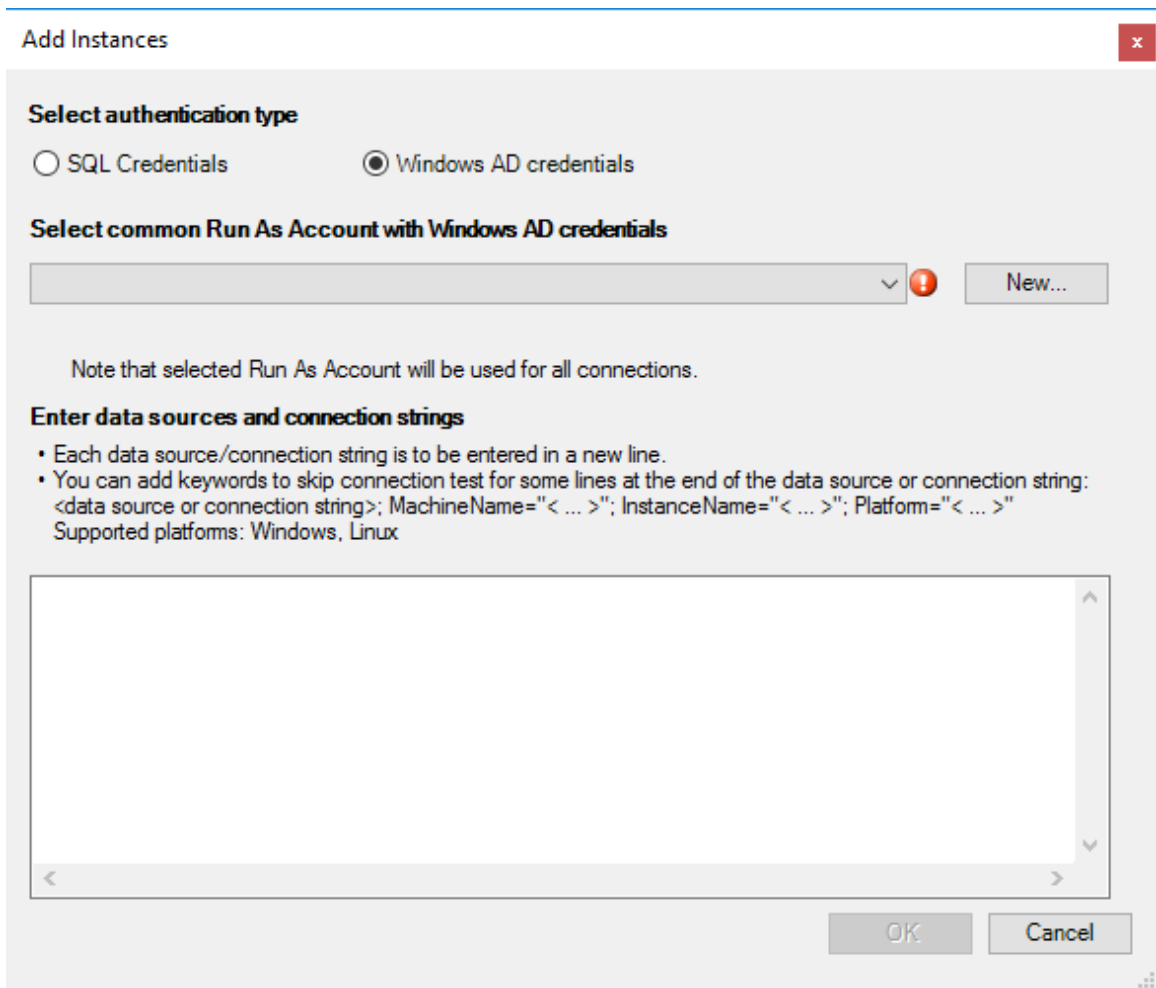
In **Service Details** window, you should provide the corresponding details about the instances you want to monitor.



Click the corresponding button to **Add Instances** for monitoring.



In this window, select a preferable authentication type: SQL or Windows AD credentials. The latter should be used when the SQL servers run on Windows servers, which are part of an Active Directory domain.



In this window, you must also select a common Run As Account created in the Operations Manager with appropriate credentials, or create a new one by clicking the **New...** button.

Account name: [] !

Login: []

Password: []

Confirm password: []

OK Cancel

In the corresponding window, enter your new Run as Account name and credentials of the SQL server you want to monitor, and click the **OK** button.

Then, enter the data sources and (or) connection strings in the corresponding field. Please, follow the instructions provided in this window to avoid errors and skip the excessive connection testing.

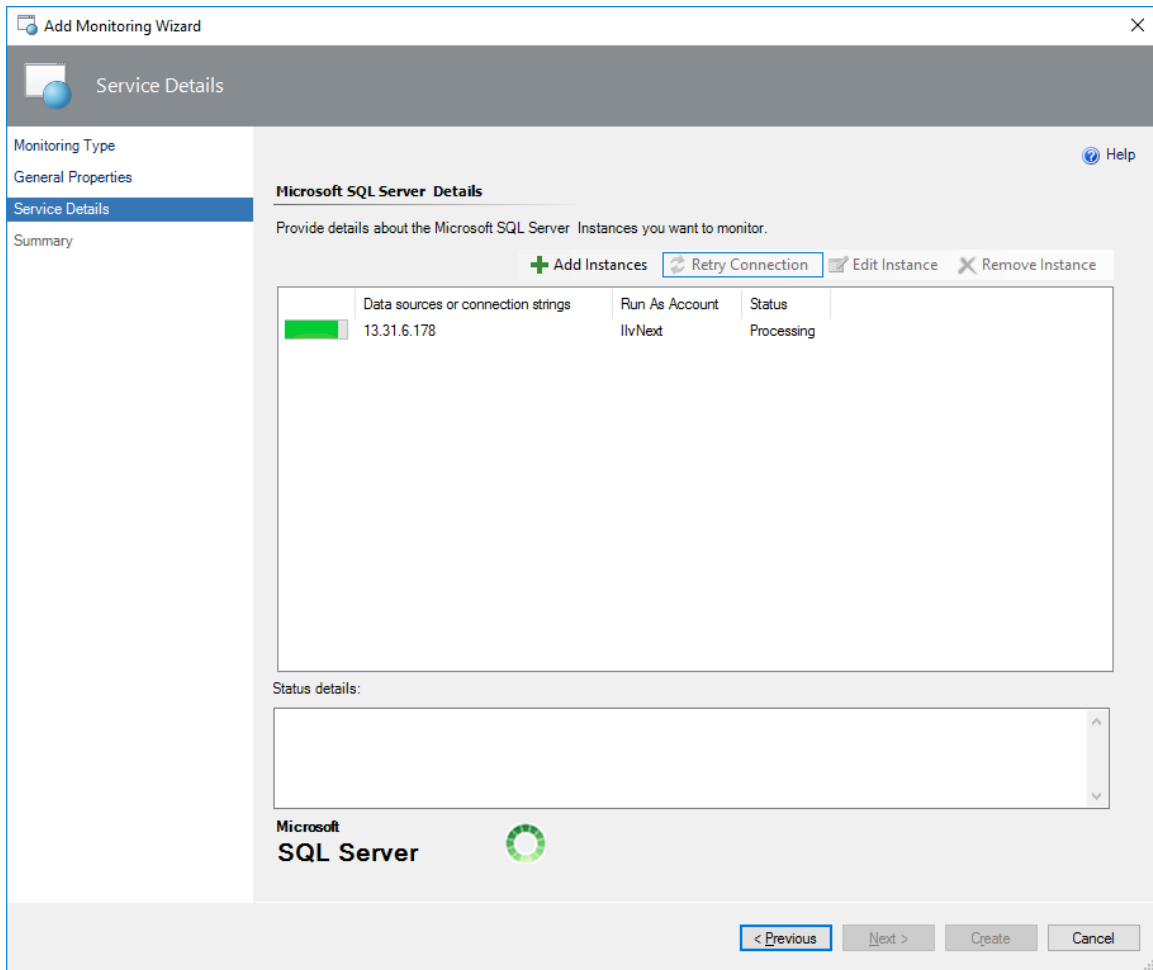
The data is to be entered in the format provided in the examples below:

172.31.2.133;MachineName="W12BOX-839";InstanceName="MSSQLSERVER";Platform="Windows"

172.31.2.133,50626;MachineName="W12BOX-839";InstanceName="SQLEXPRESS";Platform="Windows"

172.17.5.115;MachineName="ubuntu";InstanceName="MSSQLSERVER";Platform="Linux"

Click the **OK** button to submit the entered data.



Note: When adding a Linux-based instance, the connection test fails if IP address is specified as a connection string and the authentication type is "Windows AD credentials". In this case, specify the name of the machine as a connection string.

When the connection testing is completed, you can view and edit properties of the added instance. To do that, select the instance and click the **Edit Instance** button.

Edit Instance configuration ✕

Data source or connection string:

Run As Account with SQL credentials:
 New...

Select authentication type:
 SQL Credentials Windows AD credentials

Run As Account with SQL credentials:
 New...

Skip Test Connection and enter the data manually

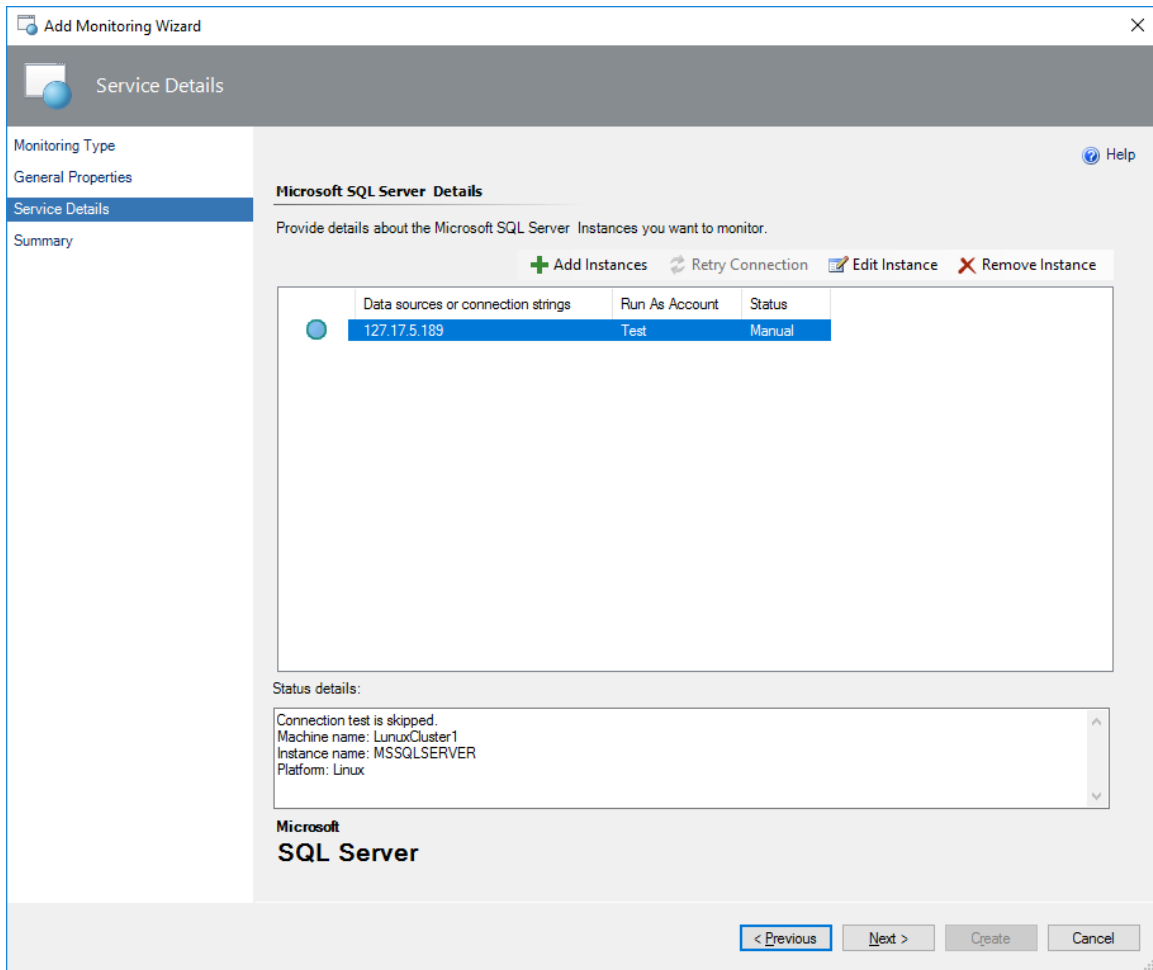
Machine name:

Instance name:

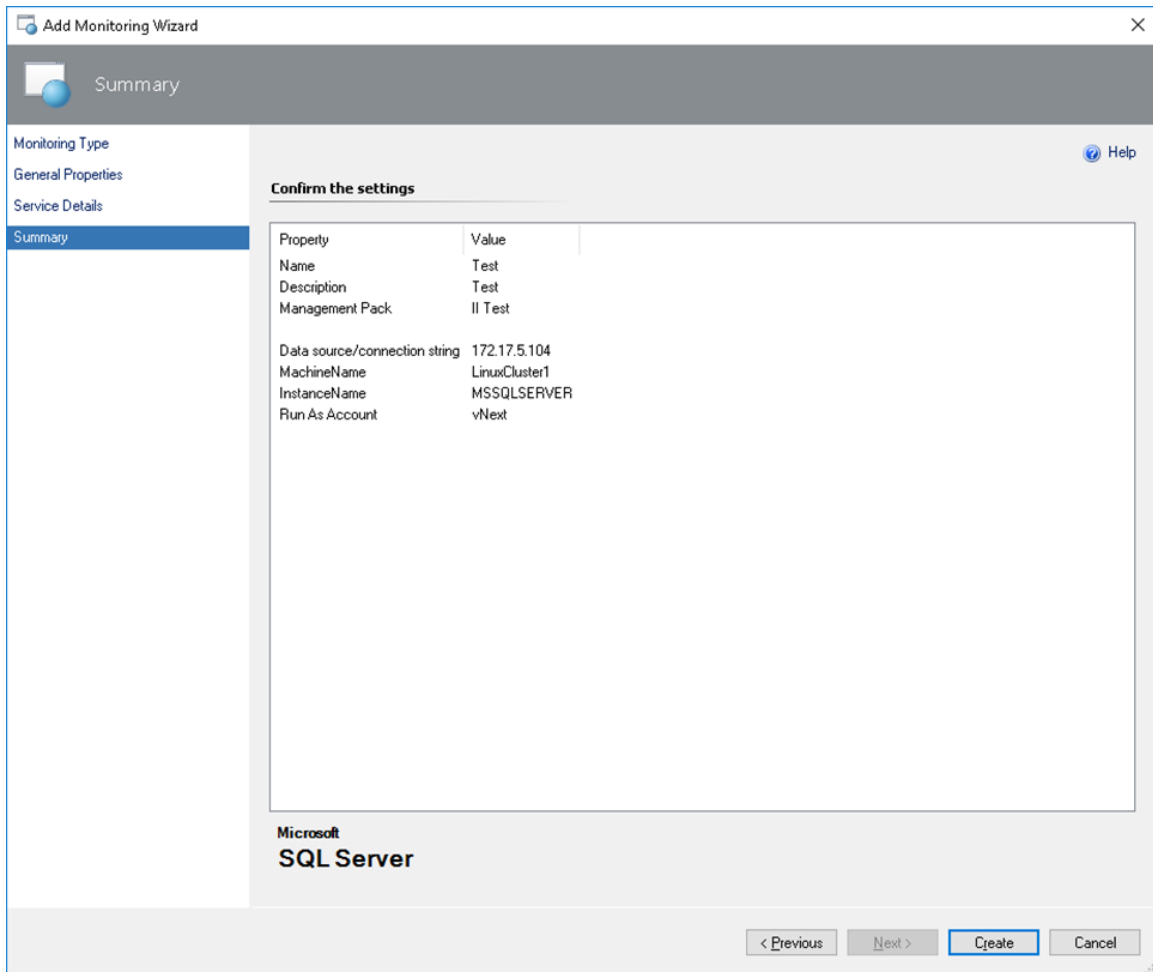
Platform:
 ▼
Windows
Linux

OK Cancel

To skip connection testing and enter the data manually, check the corresponding box in this window. If you do that, the status of your instance will be changed to “Manual”:



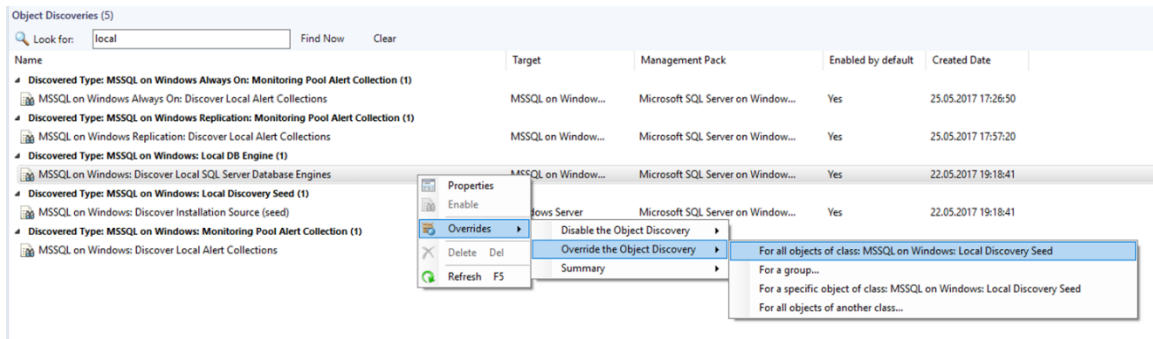
In **Summary** window, you can view your monitoring settings and confirm them by clicking the **Create** button.



After that, your monitoring template will be successfully created.

Enable Mixed Monitoring for SQL Server on Windows by Override

If you want to start monitoring SQL Server instances in Mixed Mode, in the Operations Manager navigate to **Authoring | Management Pack Objects**, select **Object Discoveries** and find **MSSQL: Discover Local SQL Database Engines on Windows** object discovery. Right-click this discovery and select the following action: **Overrides > Override the Object Discovery > For all objects of class: MSSQL on Windows: Local Discovery Seed**.



As a result, **Override Properties** window will be displayed. In this window, enable override for **Mixed Monitoring** parameter and enter the names of the instances in the **Override Value** field to switch them to agentless monitoring. Please note that the names of the instances should be separated by commas. If you want to add all the instances, enter asterisk character (*) in the field. Therefore, all instances (even those with the same names on different servers) will be monitored on the pool in the mixed mode.

Override Properties ✕

Object Discovery name: MSSQL on Windows: Discover Local SQL Server Database Engines
 Category: Discovery
 Overrides target: Class: MSSQL on Windows: Local Discovery Seed

Override-controlled parameters: Show Object Discovery Properties...

	Override	Parameter Name ▲	Parameter Type	Default Value	Override Value	Effective Value	Change Status
▶	<input checked="" type="checkbox"/>	Enabled	Boolean	True	True	True	[No change]
	<input type="checkbox"/>	Interval (seconds)	Integer	14400	14400	300	[Deleted]
	<input type="checkbox"/>	Mixed Monitoring	String				[Deleted]
	<input type="checkbox"/>	Synchronization Time	String				[No change]
	<input type="checkbox"/>	Timeout (seconds)	Integer	300	300	300	[No change]

Details:

Enabled	Description	Edit...
<input checked="" type="checkbox"/>	The parameter is not set by a custom override or by a management pack. The effective value of this parameter is the default value of this parameter.	

Management pack

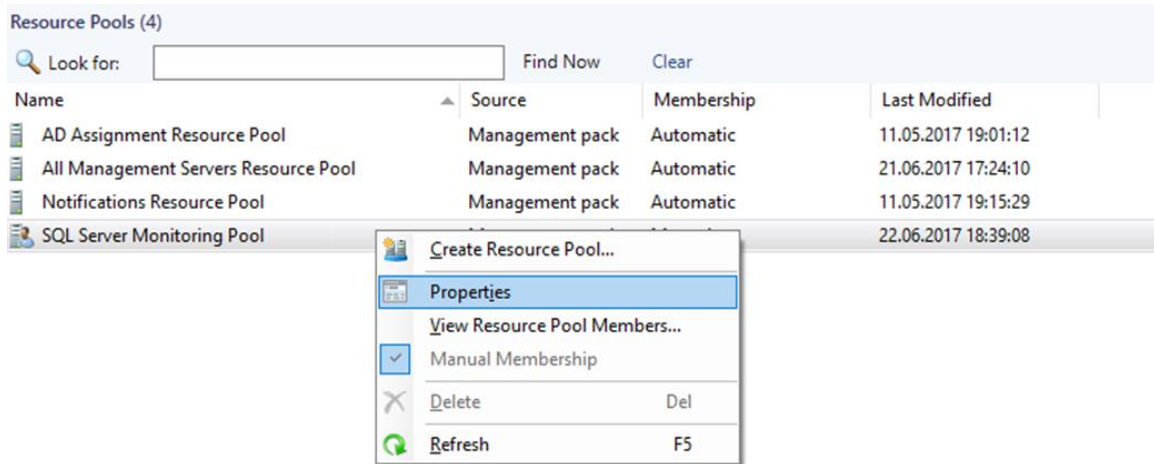
Select destination management pack:

<Select Management Pack> New...

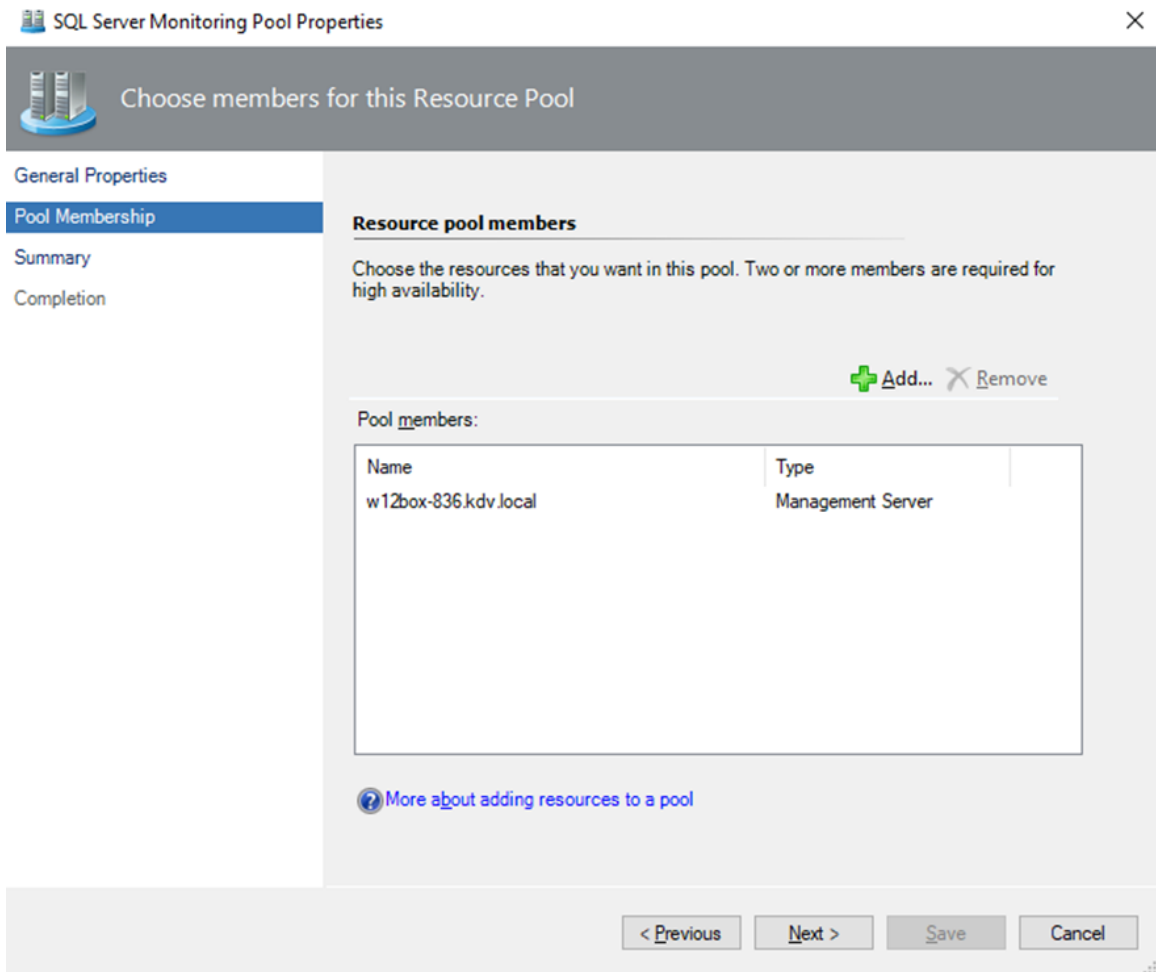
Help OK Apply Cancel

Configure SQL Server Monitoring Pool

The monitoring pool is available for configuration in the Operations Manager. To configure the monitoring pool, navigate to **Administration | Resource Pools**, right-click **SQL Server Monitoring Pool** in the list of Resource Pools and check **Manual Membership** option. Then, select **Properties** action.



As a result, **SQL Server Monitoring Pool Properties** window will be displayed. In this window, select **Pool Membership** tab. In this tab, click the **Add...** button to populate the monitoring pool.



You can configure SQL Server Monitoring Pool manually by adding custom Gateways or Management Servers.

Management Pack Purpose

In this section:

- [Monitoring Types](#)
- [Monitoring Scenarios](#)
- [How Health Rolls Up](#)



Note

For details on the discoveries, rules, and monitors contained in this management pack, see a separate “Microsoft SQL Server Management Pack Objects and Workflows” appendix to the guide.

Monitoring Types

This management pack provides three monitoring types described below.

Local Agent Monitoring

When the monitoring is provided via SCOM agent, the monitoring is performed according to the standard variant known as Local Agent Monitoring. There are no additional actions required to perform the monitoring via Local Agent; Microsoft Monitoring Agent installed on the server is enough. If it is necessary to perform Low-Privilege monitoring, configure the environment according to the instructions provided in the corresponding [article](#) of Low-Privilege Environments section.

Agentless Monitoring

Agentless Monitoring is performed from Management Servers or Gateway Servers mapped to the SQL Server Monitoring Pool (see [Configure SQL Server Monitoring Pool](#) section). At that, both SQL Server authentication and Windows authentication are supported. To configure agentless monitoring, follow to Authoring tab and choose Microsoft SQL Server 2017+ template. Add the instances by means of the [Add Monitoring Wizard](#). If it is necessary to perform Low-Privilege monitoring, configure the environment according to the instructions provided in the corresponding [article](#) of Low-Privilege Environments section.



Important

The following rules are not active if SQL Server on Windows instance is monitored agentlessly:

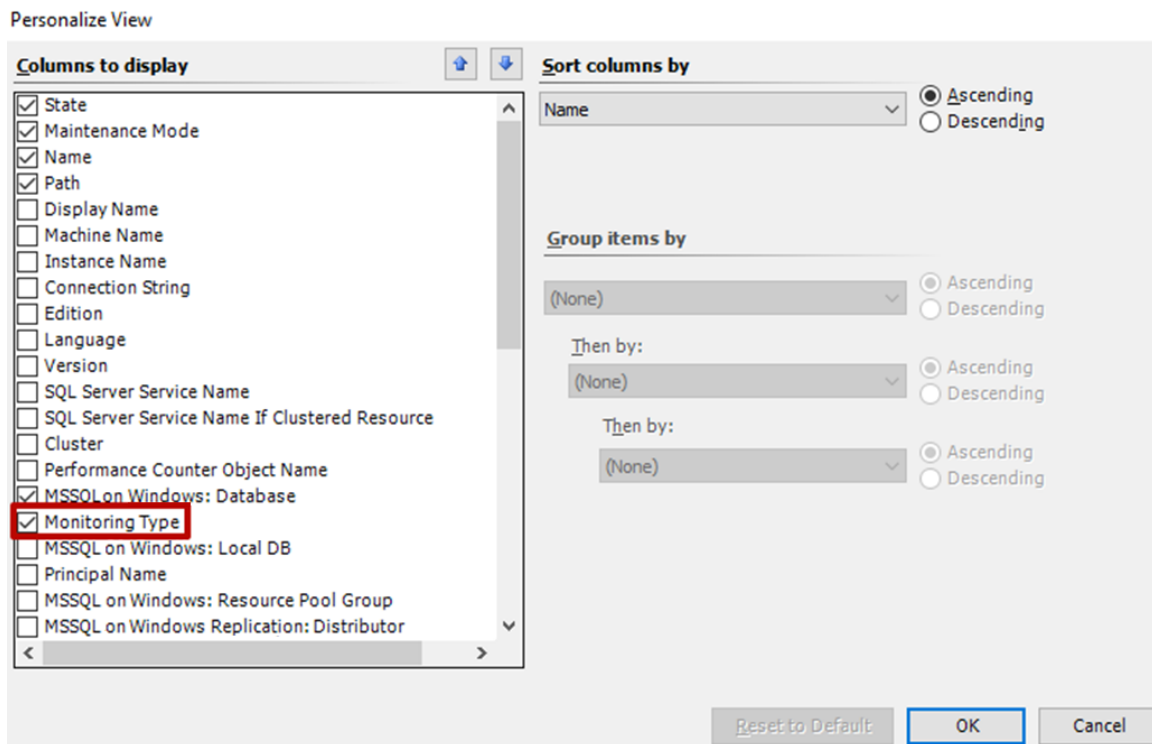
- MSSQL on Windows: Alert engine stopped due to unrecoverable local eventlog errors
- MSSQL on Windows: A SQL job failed to complete successfully
- MSSQL on Windows: Job step cannot be run because the subsystem failed to load
- MSSQL on Windows: The agent is suspect. No response within last minutes
- MSSQL on Windows: SQL Server Agent could not be started
- MSSQL on Windows: SQL Server Agent initiating self-termination
- MSSQL on Windows: Step of a job caused an exception in the subsystem
- MSSQL on Windows: SQL Server Agent is unable to connect to SQL Server
- MSSQL on Windows: Unable to re-open the local eventlog

Mixed Monitoring

Mixed Monitoring is intended for cases when you want to switch the monitoring from the agent to a SCOM pool. Such monitoring mode is quite similar to Agentless Monitoring, but in this case, you do not need to configure the connection strings manually. You can enable Mixed Monitoring by override (see the corresponding [section](#) for details). When you enable Mixed Monitoring, only SQL Server Seed is discovered locally by the SCOM agent. All other workflows are run from the pool. The details regarding the SQL Server Monitoring Pool discovery configuration are available in [Configure SQL Server Monitoring Pool](#) section. If it is necessary to perform Low-Privilege monitoring, configure the environment according to the instructions provided in the corresponding [article](#) of Low-Privilege Environments section.

View the Current Monitoring Types

To view the currently used monitoring types in the Operations Manager, go to **Database Engines** view, open **Personalize View** menu and enable **Monitoring Type** parameter:



Therefore, the complete **Database Engines** view will look as follows:

Database Engines (9)				
Look for: <input type="text"/> Find Now Clear				
State	Name	Path	MSSQL on Windows: Database	Monitoring Type
Critical	KDVCLUSTER.MSSQLSERVER		Healthy	Mixed
Healthy	SQLAON-037.MSSQLSERVER		Healthy	Mixed
Warning	SQLAON-038.MSSQLSERVER		Warning	Agentless
Critical	W16BOX-VNEXT1-3.MSSQLSERVER		Critical	Mixed
Healthy	W16BOX-VNEXT1-3.SQLEXPRESS_TRUE		Healthy	Local
Critical	W16BOX-VNEXT1-3.VNEXTCTP		Critical	Local
Critical	W16BOX-VNEXT1-4.MSSQLSERVER		Critical	Agentless
Critical	W16BOX-VNEXT1-4.SQLEXPRESS		Critical	Agentless
Critical	W16BOX-VNEXT1-4.VNEXTCTP		Critical	Agentless

Special Cases

There are several rules in SQL Server on Windows management pack, which raise alerts when the following events appear in Windows Application Event Log:

- MSSQL on Windows: Alert engine stopped due to unrecoverable local eventlog errors
- MSSQL on Windows: A SQL job failed to complete successfully
- MSSQL on Windows: Job step cannot be run because the subsystem failed to load
- MSSQL on Windows: The agent is suspect. No response within last minutes
- MSSQL on Windows: SQL Server Agent could not be started
- MSSQL on Windows: SQL Server Agent initiating self-termination
- MSSQL on Windows: Step of a job caused an exception in the subsystem
- MSSQL on Windows: SQL Server Agent is unable to connect to SQL Server
- MSSQL on Windows: Unable to re-open the local eventlog

These rules work in Local mode, but they do not work in Mixed mode by default. The Operations Manager does not alert on or collect events in the event log on an agent that are written from a computer other than the local computer. However, overriding "AllowProxying" parameter for these rules makes it possible.

Note: Enabling this option may cause remote code execution. Therefore, this flag is considered potentially harmful. Unless you make sure that your computer is secure, it is not recommended to override "AllowProxying" parameter.

Monitoring Scenarios

Discovery of SQL Server Database Engine Instances

When Agent Monitoring or Mixed Monitoring mode is used, the management pack automatically discovers stand-alone and clustered instances of SQL Server across all managed systems that run System Center Operations Manager agent service.

Database Discovery and State Monitoring

For each managed database engine, the databases are discovered and monitored by means of a number of rules and monitors. Please refer to a separate “Microsoft SQL Server Management Pack Objects and Workflows” appendix to the guide for the full list of rules and monitors targeted to databases.

You can apply overrides to the discovery to specify an “Exclude List” (in comma-delimited format) of database names that the discovery should not consider.

Data File and Transaction Log File Space Monitoring

The management pack collects a set of metrics to enable the space monitoring at File, Filegroup and Database levels. You may use reports to review this information for multiple databases and for long time intervals.

This feature supports following types of media:

- Local storage (both drive letters and mount points)
- Cluster Shared Volumes
- SMB Shares
- Azure BLOBs

By default, space monitoring is enabled for all levels. Therefore, an alert will be registered only when all files in the filegroup are unhealthy. If your environment is sensitive for any extra load, you may consider disabling monitoring on Filegroup and File level.

Many Databases on the Same Drive

Space monitoring introduced by this management pack may be noisy in environments where many databases share the same media and have the **autogrowth** setting enabled. In such cases,

an alert for each database is generated when the amount of free space on the hard drive reaches the threshold. To reduce the noise, turn off the space monitors for data and transaction log files, and use Operating System Management Pack to monitor space on the hard drive.

DB Storage Latency Monitoring

This management pack collects “DB Disk Read Latency (ms)” and “DB Disk Write Latency (ms)” performance metrics for each database. In addition, the management pack defines two associated monitors, which register alerts in case of significant performance degradation. These monitors are disabled by default. Enable these monitors only for specific DBs when necessary.

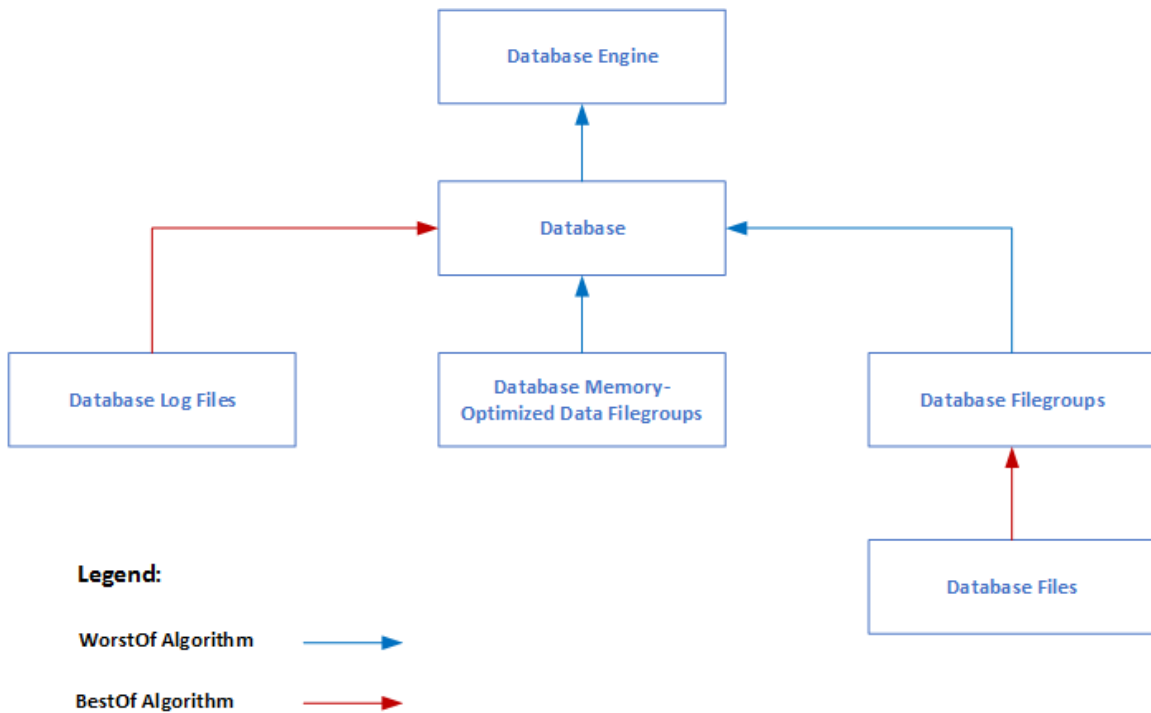
Blocked Sessions

The management pack defines the “**Blocking Sessions**” monitor, which is designed to query each database for the session, which is blocked during a significant period. If blocking is detected and it exceeds the given threshold, then the state is changed and an alert is raised.

You can apply an override to change the **WaitMinutes** parameter, which is used to determine if the blocked session should be considered as long running or not. The default value for this parameter is **one minute**.

How Health Rolls Up

The following diagram shows how the health states of objects roll up in this management pack.



Configure the Management Pack

This section provides guidance on configuring and tuning this management pack.

In this section:

- [Mandatory Configuration](#)
- [Best Practice: Create a Management Pack for Customizations](#)
- [How to Enable Agent Proxy Option](#)
- [How to Import a Management Pack](#)
- [Security Configuration](#)

Mandatory Configuration

To configure Management Pack for Microsoft SQL Server complete following steps:

- Review the “[Configure the Management Pack](#)” section of this guide.
- Grant the required permissions as described in “[Security Configuration](#)” section of this guide.

- Enable the Agent Proxy option on all agents that are installed on servers that are members of the cluster. It is not necessary to enable this option for standalone servers. For more information about enabling Agent Proxy option, see “[How to Enable Agent Proxy Option](#)” section of this guide.
- Import the Management Pack.
- Associate SQL Server Run As profiles with accounts that have appropriate permissions. For more information about configuring Run As profiles see “[How to Configure Run As Profiles](#)” section of this guide.
- To ensure correct operation of the management pack, check that SQL Server instances allow connections via TCP/IP protocol, and that SQL Server Browser service always runs on the host server. TCP/IP protocol is supported for all monitoring scenarios. Note that “Shared memory” protocol is supported for Local Agent monitoring scenarios only. “Named Pipes” protocol is not supported.

Best Practice: Create a Management Pack for Customizations

The Management Pack for Microsoft SQL Server is sealed so that you cannot change any of the original settings in the management pack file. However, you can create customizations, such as overrides or new monitoring objects, and save them to a different management pack. By default, the Operations Manager saves all customizations to the default management pack. As a best practice, you should instead create a separate management pack for each sealed management pack you want to customize.

Creating a new management pack for storing overrides has the following advantages:

- When you create a management pack to store customized settings for a sealed management pack, it is helpful to base the name of the new management pack on the name of the management pack that it is customizing, such as “Microsoft SQL Server Overrides”.
- Creating a new management pack for storing customizations of each sealed management pack makes it easier to export the customizations from a test environment to a production environment. It also makes it easier to delete a management pack, because you must delete any dependencies before you can delete a management pack. If customizations for all management packs are saved in the Default Management Pack and you need to delete a single management pack, you must first delete the Default Management Pack, which also deletes customizations to other management packs.

For more information about sealed and unsealed management packs, see [Management Pack Formats](#) article. For more information about management pack customizations and the default management pack, see [About Management Packs](#) article.

▶ How to Create a New Management Pack for Customizations

1. Open the Operations console, and then click the **Administration** button.
2. Right-click **Management Packs**, and then click **Create New Management Pack**.
3. Enter a name (for example, SQLMP Customizations), and then click **Next**.
4. Click **Create**.

How to Enable Agent Proxy Option

To enable **Agent Proxy option** complete following steps:

1. Open the Operations Console and click the **Administration** button.
2. In the Administrator pane, click **Agent Managed**.
3. Double-click an agent in the list.
4. On the Security tab, select **Allow this agent to act as a proxy and discover managed objects on other computers**.

How to Import a Management Pack

For the detailed information about importing a management pack, see [How to Import a Management Pack](#) article.

Security Configuration

This section provides guidance on configuring the security for this management pack.

In this section:

- [Run As Profiles](#)
- [Low-Privilege Environments](#)

Run As Profiles

The list of Run As profiles is as follows:

- Microsoft SQL Server Discovery Run As Profile – this profile is associated with all discoveries.
- Microsoft SQL Server Monitoring Run As Profile – this profile is associated with all monitors and rules.
- Microsoft SQL Server Run As Profile – this profile is associated with all tasks.
- Microsoft SQL Server SQL Credentials Run As Profile – this profile is used for authentication to Microsoft SQL Server Instances using SQL Credentials.

When Local Monitoring or Mixed Monitoring mode is used, all discoveries, monitors, and tasks defined in the SQL Server management packs use accounts defined in the “Default Action Account” Run As profile by default. If the default action account for a given system does not have the necessary permissions to discover or monitor the instance of SQL Server, then those systems can be bound to more specific credentials in the “Microsoft SQL Server ...” Run As profiles, which do have access.

How to Configure Run As Profiles in Local and Mixed Monitoring Modes

To configure Run As profiles, follow one of the scenarios described below:

1. SCOM Default Action Account is mapped to either Local System account, or any Domain User account, which is placed in the Local Administrators group on the operating system of the monitored machines. Note that the used account must be granted with SQL System Administrator rights (hereinafter - SA rights) in the monitored SQL Server instances (Domain User account can be granted with SA rights by granting SA to BUILTIN\Administrators local group in the SQL Server security access list). In this case, monitoring of SQL Server instances will work out of the box, except for some configurations described below. Please follow these steps to ensure that all requirements are met: If you store SQL Server databases on an SMB file share, make sure that Default Action Account has the rights described in the corresponding [Low-Privilege Environments](#) section.
2. SCOM Default Action Account is mapped to either Local System account or Domain User account as in the scenario described above, but SA rights cannot be granted to it, as long as the security policy prohibits granting SA rights to SCOM Default Action account. If the security policy permits to grant SA rights to a separate Domain User account, which will be used for launching SQL Server MP workflows only, perform the following steps:
 - a. Create a new Domain User account and add this account to Local Administrators group on each monitored server.
 - b. Grant SA rights to this account in SQL Server.

- c. Create a new Action account in SCOM and map it to the Domain User account created above.
 - d. Map the new Action account to all SQL Server MP Run As Profiles.
 - e. If you store SQL Server databases on an SMB file share, make sure that your Domain User account has the rights described in the corresponding [Low-Privilege Environments](#) section..
3. SCOM Default Action Account is mapped to Local System account, but SA rights cannot be granted thereto, as long as the security policy prohibits granting Local System with rights to access SQL Server. You can grant SA or Low Privilege rights to SCOM HealthService using its Service Security Identifier. For more details, refer to [SQL Server uses a service SID to provide service isolation](#) and [How to configure SQL Server 2012 to allow for System Center Advisor monitoring](#) articles. The configuration process is described in [How to Configure Monitoring by Means of a Service Security Identifier](#) section. Note that this method is intended for Local Monitoring Mode only.
 4. In case you need to grant the minimally required rights to SQL MP workflows, follow the instructions provided in [Low-Privilege Environments](#) section.

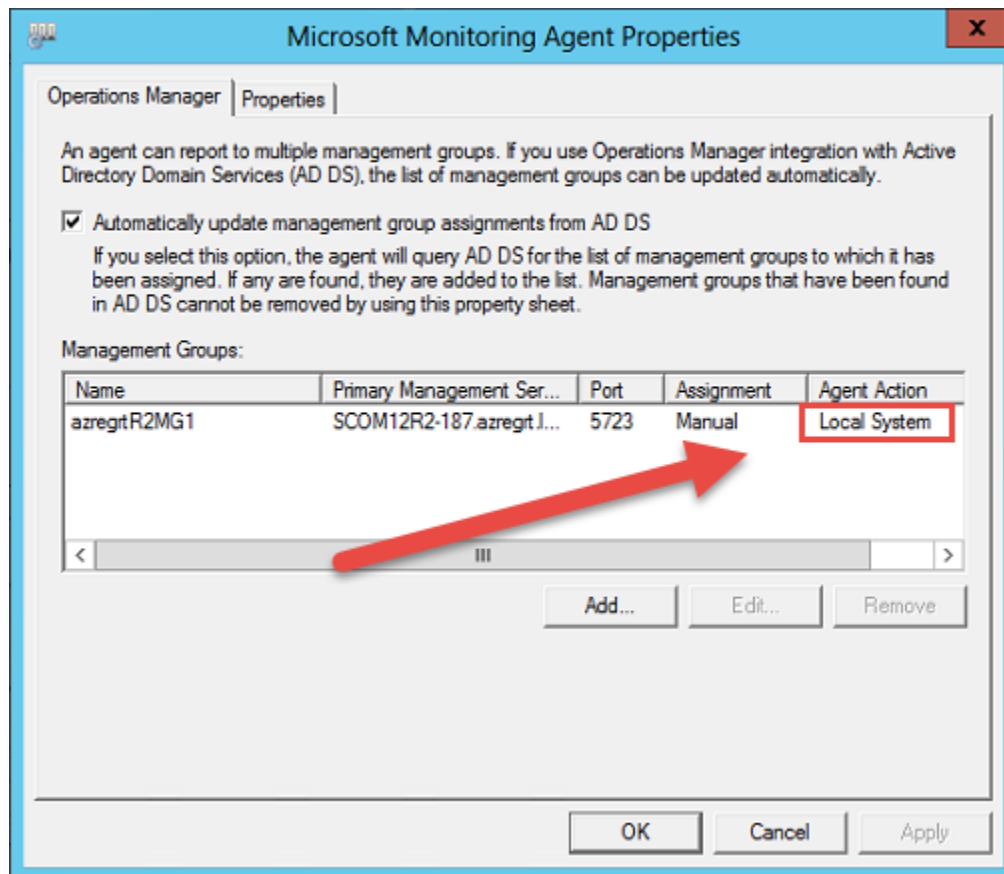
How to Configure Run As Profiles in Agentless Monitoring Mode

Create a login in SQL Server for monitoring purposes and grant it with SA rights or a set of Low Privilege permissions. You can use SQL Server authentication as well as Windows authentication for that login. Then, use this login in the Add Monitoring Wizard while adding an SQL Server instance to be monitored. Refer to [Configure Agentless Monitoring by Add Monitoring Wizard](#) section for the details on how to add an SQL Server instance to monitor it agentlessly and to [Configure Low-Privilege Agentless Monitoring by Add Monitoring Wizard](#) on how to configure Low Privilege monitoring in Agentless mode.

How to Configure Monitoring by Means of a Service Security Identifier

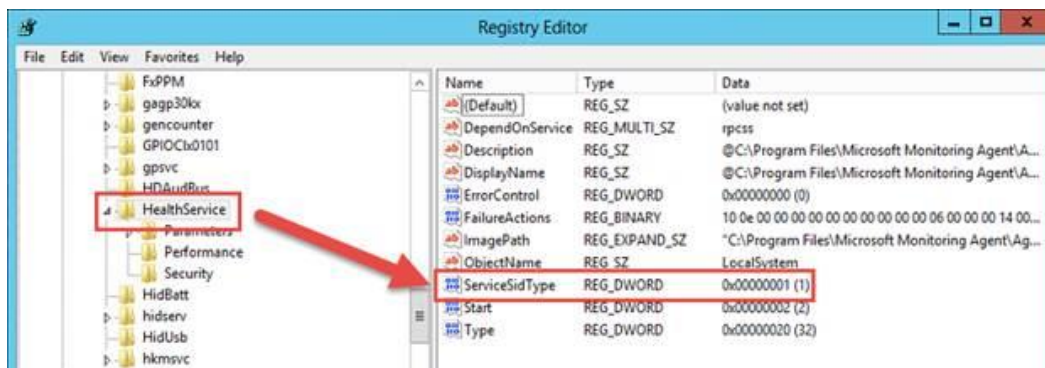
Below are the steps to configure monitoring via Service SIDs for SQL Server 2017+ on a Windows Server instance.

1. SCOM monitoring agent should use either “Local System” account (as in case illustrated below), or any other domain account without Administrator rights.



2. Open "Registry Editor" at the SCOM Agent managed computer. Add "ServiceSidType" REG_DWORD key with "1" value at "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HealthService".

This step is necessary to enable a service SID for Health Service.



3. Open Command Prompt as Administrator and run "*sc sidtype HealthService unrestricted*" query; then, restart "Health Service".

- Open Command Prompt as Administrator and run next query: "sc showsid HealthService". The service "STATUS" should be "Active":

```

Administrator: Command Prompt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\administrator.azregt>sc sidtype HealthService unrestricted
[SC] ChangeServiceConfig2 SUCCESS

C:\Users\administrator.azregt>sc showsid healthservice

NAME: healthservice
SERVICE_SID: S-1-5-80-3696737894-3623014651-202832235-645492566-13622391
STATUS: Active
C:\Users\administrator.azregt>sc showsid healthservice_

```

- Open SSMS and connect to the SQL Server 2017+ on Windows instance, which should be monitored using Health Service SID.
- Run the following query in the SSMS:

```

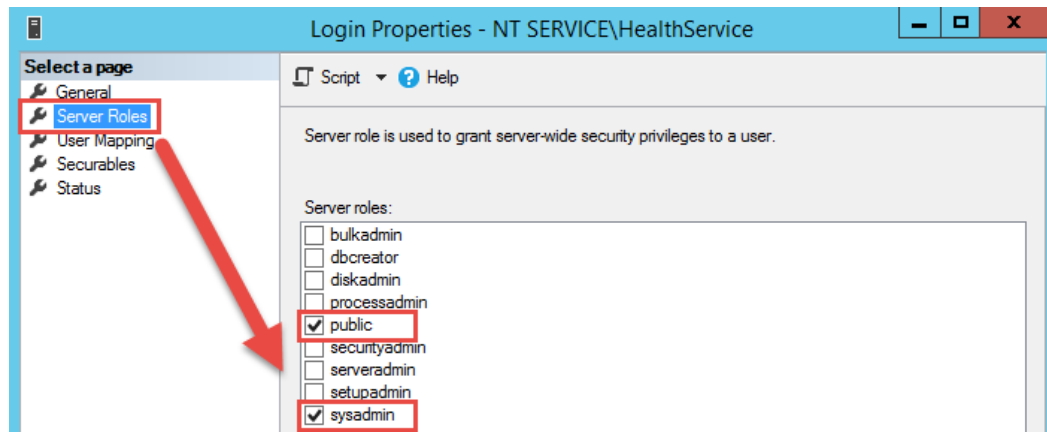
SET NOCOUNT ON;
DECLARE @accountname nvarchar(128);
DECLARE @command1 nvarchar(MAX);
DECLARE @command2 nvarchar(MAX);
DECLARE @command3 nvarchar(MAX);
SET @accountname = 'NT SERVICE\HealthService';
SET @command1 = 'USE [master];
CREATE LOGIN ['+@accountname+']
FROM WINDOWS WITH DEFAULT_DATABASE=[master];';
SET @command2 = '';
SELECT @command2 = @command2 + 'USE ['+db.name+'];
CREATE USER ['+@accountname+']
FOR LOGIN ['+@accountname+'];'
FROM sys.databases db
left join sys.dm_hadr_availability_replica_states hadrstate
on db.replica_id = hadrstate.replica_id
WHERE db.database_id <> 2
AND db.user_access = 0
AND db.state = 0
AND db.is_read_only = 0
AND (hadrstate.role = 1 or hadrstate.role is null);
SET @command3 = 'USE [master];
GRANT VIEW ANY DATABASE TO ['+@accountname+'];
GRANT VIEW ANY DEFINITION TO ['+@accountname+'];
GRANT VIEW SERVER STATE TO ['+@accountname+'];
GRANT SELECT on sys.database_mirroring_witnesses to ['+@accountname+'];
USE [msdb];

```

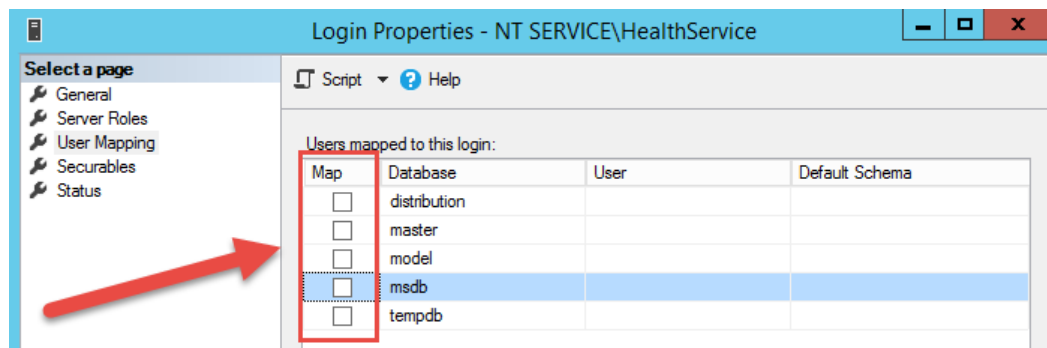


```
EXEC sp_addrolemember @rolename="PolicyAdministratorRole",
@membername="'+@accountname+'";
EXEC sp_addrolemember @rolename="SQLAgentReaderRole",
@membername="'+@accountname+'";
EXECUTE sp_executesql @command1;
EXECUTE sp_executesql @command2;
EXECUTE sp_executesql @command3;
```

- Grant “public” and “sysadmin” Server roles to “NT SERVICE\HealthService” login:



- Uncheck “Map” checkboxes for all databases at the “User Mapping” tab



Low-Privilege Environments

This section describes how to configure the Management Pack for Microsoft SQL Server for low-privilege access. All workflows (discoveries, rules, monitors, and actions) in this management pack are bound to Run As profiles described in “[Run As Profiles](#)” section. To enable low-privilege monitoring, appropriate permissions should be granted to Run As accounts and these accounts

should bound to respective Run As profiles. Subsections below describe how to grant permissions at both Operating System and SQL Server level for all [monitoring types](#).



Note

Please refer to “[Run As Profiles](#)” section for the detailed explanation of what Run As profiles are defined in Management Pack for Microsoft SQL Server.



Note

For more information about configuring Run As profiles, see “[How to Configure Run As Profiles](#)” section of this guide.

Local Agent Monitoring

To configure low-privilege environments for local agent monitoring, perform the steps described below.

Configure a Low-Privilege Environment in Active Directory

1. In Active Directory, create three domain users that will be commonly used for low-privilege access to all target SQL Server instances:
 - a. **SQLTaskAction**
 - b. **SQLDiscovery**
 - c. **SQLMonitor**
2. Create a domain group named **SQLMPLowPriv** and add the following domain users:
 - a. **SQLDiscovery**
 - b. **SQLMonitor**
3. Grant special permission: Read-only Domain Controllers – “Read Permission” to the **SQLMPLowPriv**

Configure a Low-Privilege Environment on the Agent Machine

1. Grant Read permission on “**HKLM:\Software\Microsoft\Microsoft SQL Server**” registry path for **SQLTaskAction** and **SQLMPLowPriv**.
2. Add the **SQLTaskAction** and **SQLMonitor** domain users to “EventLogReaders” local group.
3. Configure the “Allow log on locally” local security policy setting to allow the **SQLTaskAction** domain user and **SQLMPLowPriv** domain group users to log on locally.
4. Grant “Execute Methods”, “Enable Account”, “Remote Enable”, “Read Security” permissions to **SQLTaskAction** and **SQLMPLowPriv** for these WMI namespaces:
 - a. **root**
 - b. **root\cimv2**

- c. **root\default**
 - d. **root\Microsoft\SqlServer\ComputerManagement14**
5. Grant Read permission on “HKLM:\Software\Microsoft\Microsoft SQL Server*InstanceID*\MSSQLServer\Parameters” registry path for **SQLMPLowPriv** for each monitored instance.

Configure a Low-Privilege Environment on the Instance of SQL Server Database Engine

1. Open SQL Server Management Studio and connect to the instance of SQL Server Database Engine.
2. In SQL Server Management Studio, for each instance of SQL Server Database Engine running on a monitored server, create a login for “**SQLMPLowPriv**” and grant the following permissions:
 - a. VIEW SERVER STATE
 - b. VIEW ANY DATABASE
 - c. VIEW ANY DEFINITION
 - d. EXECUTE ON xp_readerrorlog

```
use msdb
go
GRANT VIEW server state to [SQLMPLowPriv]
GRANT VIEW any definition to [SQLMPLowPriv]
GRANT VIEW any database to [SQLMPLowPriv]
GRANT EXECUTE ON xp_readerrorlog TO [SQLMPLowPriv]
```

3. Create a **SQLMPLowPriv** user in each user database, master, msdb, and model. Link **SQLMPLowPriv** users to **SQLMPLowPriv** login. By adding user into the model database, you will automatically create a **SQLMPLowPriv** user in each future user-created database. You will need to provision the user manually for any database that will be attached or restored in future.

```
use msdb
go
CREATE USER [SQLMPLowPriv] FOR LOGIN [SQLMPLowPriv]
```

4. For msdb database, grant a **SQLMPLowPriv** user the following permissions:
 - a. EXECUTE ON msdb.dbo.sp_help_job
 - b. EXECUTE ON msdb.dbo.sp_help_jobactivity
 - c. SELECT ON sysjobs_view
 - d. SELECT ON sysschedules
 - e. SELECT ON sysjobschedules

- f. SELECT ON log_shipping_monitor_history_detail
- g. SELECT ON log_shipping_monitor_secondary
- h. SELECT ON log_shipping_secondary_databases
- i. SELECT ON log_shipping_monitor_primary
- j. SELECT ON log_shipping_primary_databases

```

use msdb
go
grant EXECUTE ON msdb.dbo.sp_help_job to [SQLMPLowPriv]
grant EXECUTE ON msdb.dbo.sp_help_jobactivity to [SQLMPLowPriv]
grant SELECT ON sysjobs_view to [SQLMPLowPriv]
grant SELECT ON sysschedules to [SQLMPLowPriv]
grant SELECT ON sysjobschedules to [SQLMPLowPriv]
grant SELECT ON log_shipping_monitor_history_detail to [SQLMPLowPriv]
grant SELECT ON log_shipping_monitor_secondary to [SQLMPLowPriv]
grant SELECT ON log_shipping_secondary_databases to [SQLMPLowPriv]
grant SELECT ON log_shipping_monitor_primary to [SQLMPLowPriv]
grant SELECT ON log_shipping_primary_databases to [SQLMPLowPriv]

```

5. For msdb database: add the **SQLMPLowPriv** user to the **SQLAgentReaderRole** database role.

```

use msdb
go
ALTER ROLE [SQLAgentReaderRole] ADD MEMBER [SQLMPLowPriv]

```

6. For msdb database: add the **SQLMPLowPriv** user to the **PolicyAdministratorRole** database role.

```

use msdb
go
ALTER ROLE [PolicyAdministratorRole] ADD MEMBER [SQLMPLowPriv]

```

Configure a Low-Privilege Environment on the Server, which Hosts an SMB Share Used by SQL Server Database Engine

1. Grant share permissions by opening share properties dialog for the share, which hosts SQL Server data files or SQL Server transaction log files.
2. Grant Read permissions to **SQLMPLowPriv**.

3. Grant NTFS permissions by opening the properties dialog for the shared folder and navigate to the “Security” tab.
4. Grant Read permissions to **SQLMPLowPriv**.

Configure Instances Low-Privilege Task Action Account on the Instance of SQL Server Database Engine

1. Open SQL Server Management Studio and connect to the instance of SQL Server Database Engine.
2. In SQL Server Management Studio, for each instance of SQL Server Database Engine running on a monitored server, create a login for **SQLTaskAction**.
3. Create a **SQLTaskAction** user in each user database, master, msdb, and model. Link **SQLTaskAction users to SQLTaskAction login**. By adding user into the model database, you will automatically create a **SQLTaskAction** user in each future user-created database. You will need to provision the user manually for any database that will be attached or restored in future.

```
use msdb
go
CREATE USER [SQLTaskAction] FOR LOGIN [SQLTaskAction]
```

Enable Execution of System Center Operations Manager Tasks for a Database Object

Some optional System Center Operations Manager tasks require a higher privilege on an agent machine and/or database to allow the task execution.

You should execute the following provisioning steps on the agent machine or the database only if you want to allow the System Center Operations Manager console operator to take remedial actions on that target.

1. If the task is related to starting or stopping an NT service (such as DB Engine Service, SQL Server Agent service, SQL Full Text Search Service, Integration Services): on the agent machine, grant the **SQLTaskAction** user permission to start or stop an NT service. This involves setting a service’s security descriptor. For more information, see [Sc sdset](#).

Read the existing privileges for a given service (using **sc sdshow**) and then grant additional privileges to the **SQLTaskAction** user for that server.

For example, suppose the results of the **SC sdshow** command for SQL Server service are as follows:

```
D:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;IU)(A;;CCLCSWLOCRRC;;;SU)S:(AU;FA;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)
```

In this case, the following command line grants sufficient access to **SQLTaskAction** for starting and stopping the SQL Server service (please replace colored strings with appropriate values and keep everything on a single line of text):

```
sc sdset SQLServerServiceName D:(A;;GRRPWP;;;SID for
SQLTaskAction)(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO
;;;BA)(A;;CCLCSWLOCRRC;;;IU)(A;;CCLCSWLOCRRC;;;SU)S:(AU;FA;CCDCLCSWRPWPDTLOCR
SDRCWDWO;;;WD)
```

2. In SQL Server Management Studio, add **SQLTaskAction** to db_owner database role for each database if the task is related to performing database checks:
 - a. "Check Catalog (DBCC)"
 - b. "Check Database (DBCC)"
 - c. "Check Disk (DBCC)" (invokes DBCC CHECKALLOC)

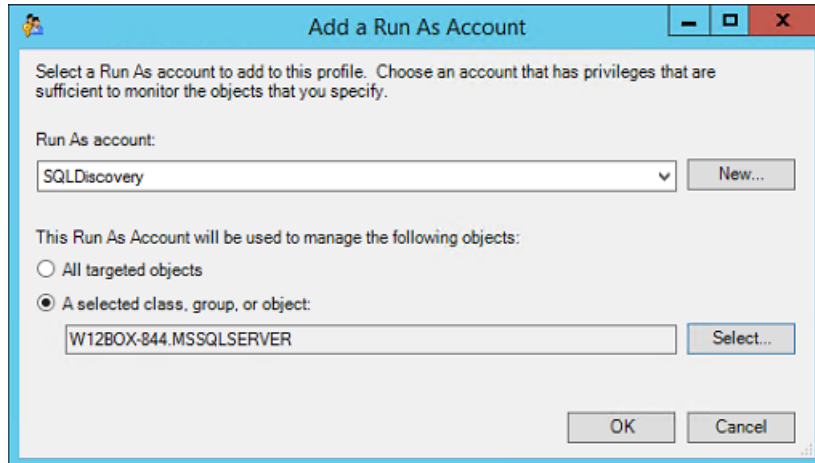
```
use msdb
go
ALTER ROLE [db_owner] ADD MEMBER [SQLTaskAction]
```

3. Grant the ALTER ANY DATABASE privilege to **SQLTaskAction** login to run the task if the task is related to changing the database state:
 - a. "Set Database Offline"
 - b. "Set Database Emergency State"
 - c. "Set Database Online"

Configure System Center Operations Manager

1. Import the SQL Server Management Pack if it has not been imported.
2. Create a **SQLTaskAction**, **SQLDiscovery** and **SQLMonitor** Run As accounts with "Windows" account type. For more information about how to create a Run As account, see [How to Create a Run As Account in Operations Manager 2007](#) or [How to Create Run As Account in Operations Manager 2012](#). For more information about various Run As Account types, see [Run As Accounts and Run As Profiles in Operations Manager 2007](#) or [Managing Run As Accounts and Profiles in Operations Manager 2012](#).
3. On the System Center Operations Manager console, configure the Run As profiles for the SQL Server Management Pack as following:
 - a. Set the "Microsoft SQL Server Task Run As Profile" Run As profile to use the **SQLTaskAction** Run As account.
 - b. Set the "Microsoft SQL Server Discovery Run As Profile" Run As profile to use the **SQLDiscovery** Run As account.
 - c. Set the "Microsoft SQL Server Monitoring Run As Profile" Run As profile to use the **SQLMonitor** Run As account.

4. To prevent problems with monitoring of SQL Server, the **SQLTaskAction**, **SQLDiscovery**, **SQLMonitor** Run As accounts should be used to manage the instances of SQL Server DB Engine:



Agentless Monitoring

To configure low-privilege environments for agentless monitoring, perform the steps described below.



Note

The steps below are suitable for SQL Server on both platforms: Windows and Linux.

Configure a Low-Privilege Environment on the Instance of SQL Server Database Engine

1. Open SQL Server Management Studio and connect to the instance of SQL Server Database Engine.
2. In SQL Server Management Studio, for each instance of SQL Server Database Engine running on a monitored server, create an SQL login for monitoring and grant the following permissions:

```
use msdb
go
GRANT VIEW server state to [SQLMPLowPriv]
GRANT VIEW any definition to [SQLMPLowPriv]
GRANT VIEW any database to [SQLMPLowPriv]
```

Go to User Mapping in setting for created SQL Login and add role **db_datareader** for database master:

```
use msdb
go
ALTER ROLE [db_datareader] ADD MEMBER [SQLMPLowPriv]
go
```

and grant the following permissions:

```
use msdb
go
GRANT EXECUTE ON xp_readerrorlog to [SQLMPLowPriv]
```

3. Create a user in each user database, master, msdb, and model. Link **created users to the login**. By adding a user into the model database, you will automatically create a user in each future user-created database. You will need to provision the user for any database that will be attached or restored in future manually:

```
use [yourdatabase]
go
CREATE USER [SQLMPLowPriv] FOR LOGIN [SQLMPLowPriv]
go
use [master]
go
CREATE USER [SQLMPLowPriv] FOR LOGIN [SQLMPLowPriv]
go
use [msdb]
go
CREATE USER [SQLMPLowPriv] FOR LOGIN [SQLMPLowPriv]
go
use [model]
go
CREATE USER [SQLMPLowPriv] FOR LOGIN [SQLMPLowPriv]
go
```

4. For msdb database grant the user the following permissions:

```
use msdb
go
GRANT EXECUTE ON msdb.dbo.sp_help_job to [SQLMPLowPriv]
GRANT EXECUTE ON msdb.dbo.sp_help_jobactivity to [SQLMPLowPriv]
```



```
GRANT SELECT ON sysjobs_view to [SQLMPLowPriv]
GRANT SELECT ON syschedules to [SQLMPLowPriv]
GRANT SELECT ON sysjobschedules to [SQLMPLowPriv]
GRANT SELECT ON log_shipping_monitor_history_detail to [SQLMPLowPriv]
GRANT SELECT ON log_shipping_monitor_secondary to [SQLMPLowPriv]
GRANT SELECT ON log_shipping_secondary_databases to [SQLMPLowPriv]
GRANT SELECT ON log_shipping_monitor_primary to [SQLMPLowPriv]
GRANT SELECT ON log_shipping_primary_databases to [SQLMPLowPriv]
```

Enable Execution of System Center Operations Manager Tasks for a Database Object

Some optional System Center Operations Manager tasks require a higher privilege on an agent machine and/or database to allow the task execution.

You should execute the following provisioning steps on the database only if you want to allow the System Center Operations Manager console operator to take remedial actions on that target.

1. In SQL Server Management Studio, add SQL Login (**SQLMPLowPriv**) to **db_owner** database role for each database if the task is related to performing database checks:
 - a. "Check Catalog (DBCC)"
 - b. "Check Database (DBCC)"
 - c. "Check Disk (DBCC)" (invokes DBCC CHECKALLOC)

```
use [yourdatabase]
go
ALTER ROLE [db_owner] ADD MEMBER [SQLMPLowPriv]
go
```

2. Grant the ALTER ANY DATABASE privilege to SQL Login (**SQLMPLowPriv**) to performing database tasks:
 - a. "Set Database Online"
 - b. "Set Database Offline"
 - c. "Set Database to Emergency State"

```
use msdb
go
GRANT ALTER ANY DATABASE to [SQLMPLowPriv]
```

3. For msdb database: add the **SQLMPLowPriv** user to the **SQLAgentReaderRole** and **PolicyAdministratorRole** database roles:

```
use [msdb]
```

```

go
ALTER ROLE [PolicyAdministratorRole] ADD MEMBER [SQLMPLowPriv]
go
use [msdb]
go
ALTER ROLE [SQLAgentReaderRole] ADD MEMBER [SQLMPLowPriv]
go

```

Configure Low-Privilege Agentless Monitoring by Add Monitoring Wizard

Perform the actions in accordance with the steps specified in "[Configure Agentless Monitoring by Add Monitoring Wizard](#)" section but with a few changes as follows:

Click the corresponding button to **Add Instances** for monitoring.

In this window, select a common Run As Account with the appropriate SQL Low-Privilege Login. Then, enter the data sources and (or) connection strings. Please, follow the instructions provided in this window to avoid errors.

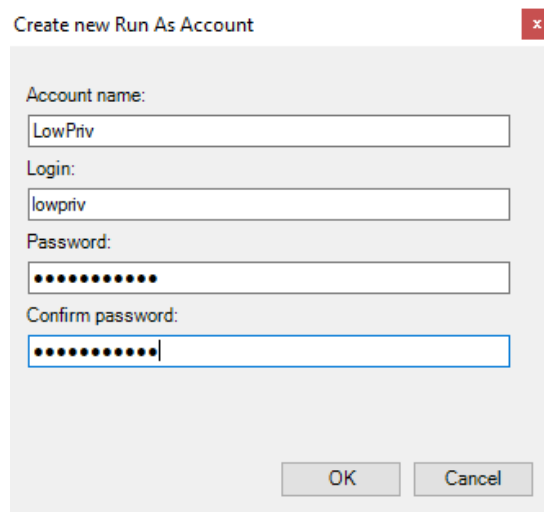
The data is to be entered in the format provided in the examples below:

172.31.2.133;MachineName="W12BOX-839";InstanceName="MSSQLSERVER";Platform="Windows"

172.31.2.133,50626;MachineName="W12BOX-839";InstanceName="SQLEXPRESS";Platform="Windows"

172.17.5.115;MachineName="ubuntu";InstanceName="MSSQLSERVER";Platform="Linux"

You can also create a new Run As account by clicking the **New...** button.



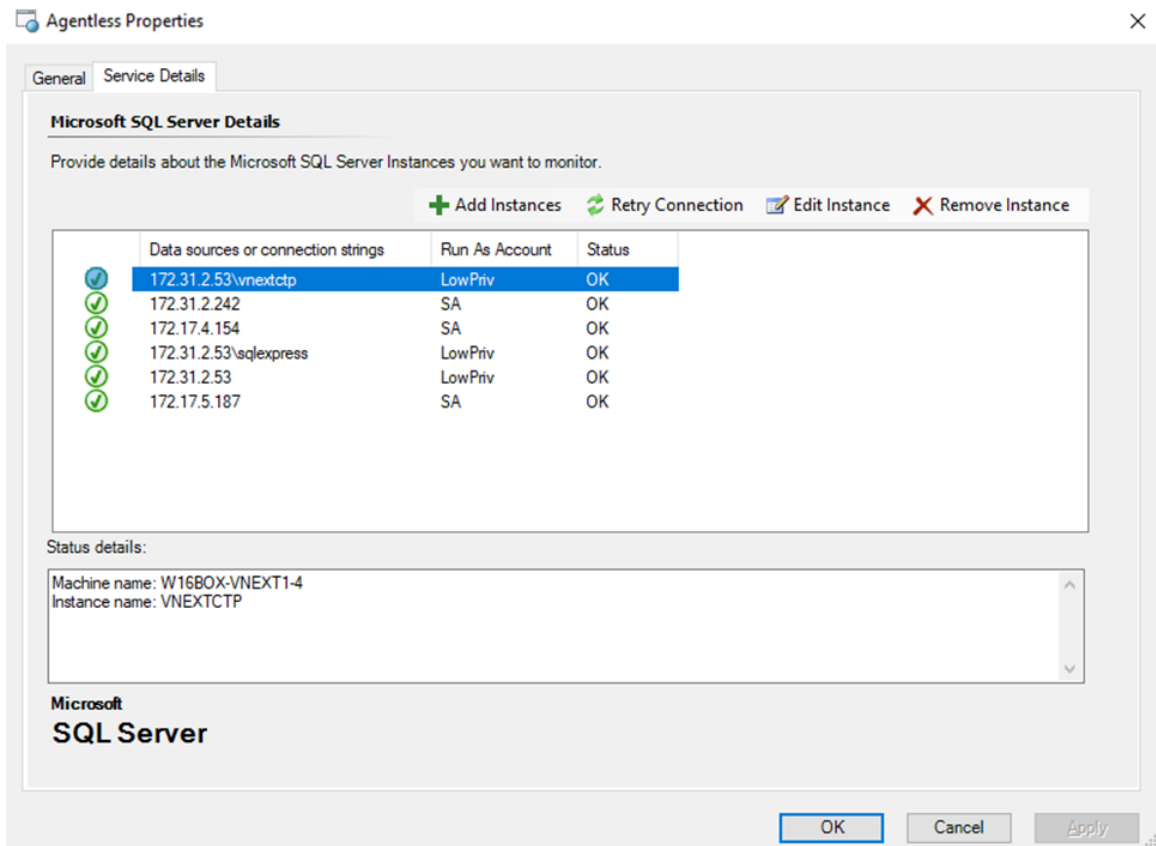
The screenshot shows a dialog box titled "Create new Run As Account". It contains the following fields and values:

- Account name: LowPriv
- Login: lowpriv
- Password: [Masked]
- Confirm password: [Masked]

At the bottom of the dialog are "OK" and "Cancel" buttons.

In the corresponding window, enter your new Run As Account name and credentials of the SQL server you want to monitor.

After clicking the **OK** button in **Add Instances** window, testing of the connection to the selected instance will be performed.



When the connection testing is completed, you can view and edit properties of the added instance. To do that, select the instance and click the **Edit Instance** button.

After that, your monitoring template will be successfully created.

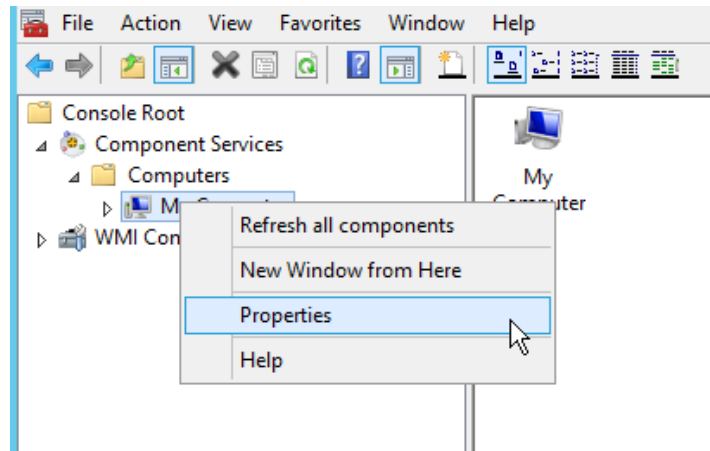
Mixed Monitoring

To configure low-privilege environments for mixed monitoring, perform the steps described in [Local Agent Monitoring](#) section; after that, perform the steps presented below:

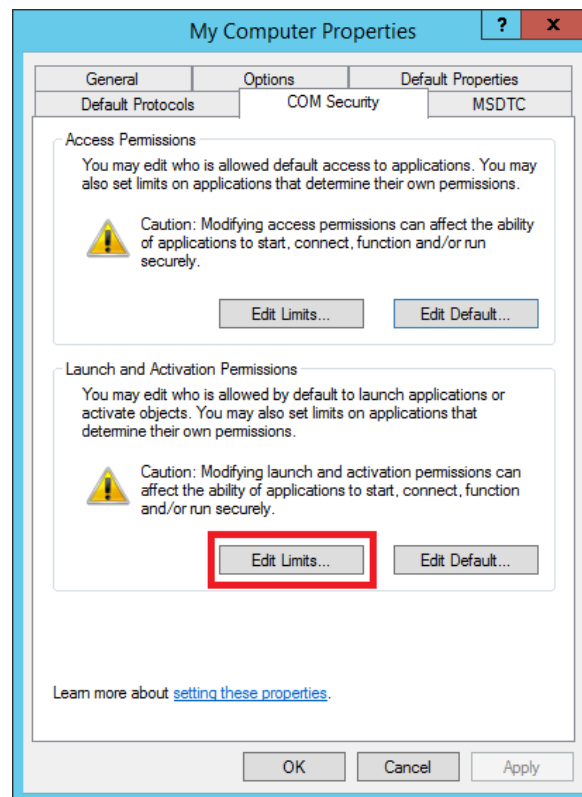
Manage Remote Access to the WMI

Below are the steps to configure security for configurations with Low Privilege account. On each mixed mode monitoring server, perform the following steps:

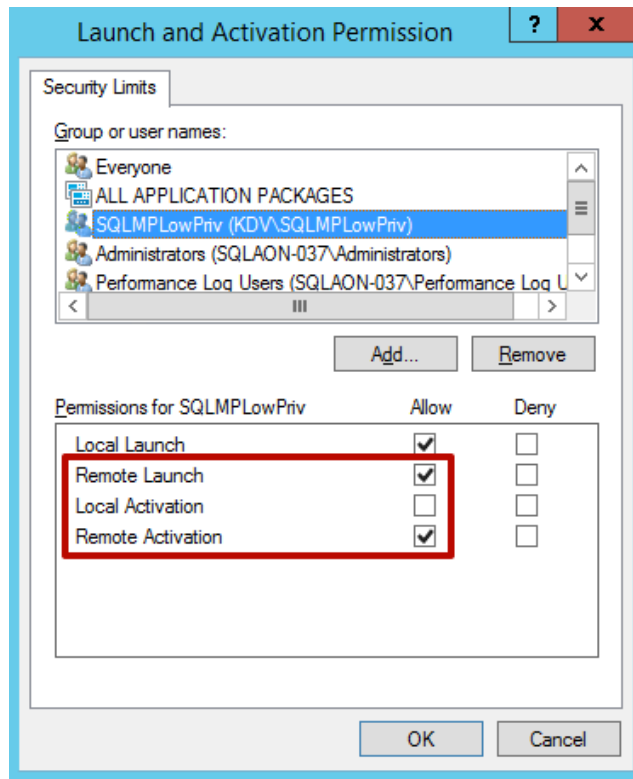
1. Launch mmc.exe and add two Snap-Ins:
 - **Component Services**
 - **WMI Control** (for local computer)
2. Expand **Component Services**, right-click **My Computer** and click **Properties**; the corresponding dialog menu will be displayed:



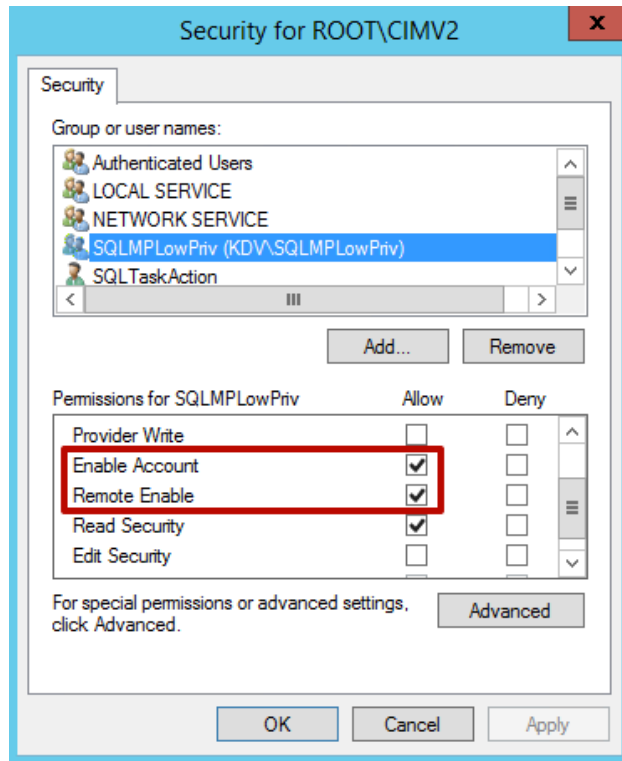
3. In this dialog menu, go to **Security** tab.
4. Click the **Edit Limits** button in **Launch and Activation Permissions** section; the corresponding dialog menu will be displayed:



5. In this dialog menu, set the following permissions for the remote machine's account:
 - **Remote Launch**
 - **Remote Activation**



6. Go to **WMI Control** snap-In and call its properties; the corresponding dialog menu will be displayed.
7. In this dialog menu, go to **Security** tab, select **Root\CIMV2**, **Root\Microsoft\SqlServer**, and **Root\Microsoft\SqlServer\ComputerManagement14** namespaces and click the **Security** button.
8. Add the following permissions for the target computer:
 - **Enable Account**
 - **Remote Enable**



9. Click the **Advanced** button; the corresponding dialog menu will be displayed.
10. In this dialog menu, select the target account and click the **Edit** button.
11. In the following dialog menu, make sure that **Applies to the** parameter is set to **This namespace only** value, and the following permissions are set:
 - **Enable Account**
 - **Remote Enable**

Grant permissions to get information about services

1. Retrieve the user SID.

From the Windows command prompt, type **PowerShell** and click Enter to open the PowerShell.

Run the following command to retrieve the user SID of the *Spotlight User*.

Replace **domainName** and **userName** with the domain name and user name for the *Spotlight User* account:

```
function GetSidByName($userName){
    $objUser = New-Object System.Security.Principal.NTAccount($userName)
    $strSID = $objUser.Translate([System.Security.Principal.SecurityIdentifier])
    return $strSID.Value
}
GetSidByName 'domainName\userName'
```

```
PS C:\Users\administrator.AP-LAB> function GetSidByName($userName){
    $objUser = New-Object System.Security.Principal.NTAccount($userName)
    $strSID = $objUser.Translate([System.Security.Principal.SecurityIdentifier])
    return $strSID.Value
}
GetSidByName 'ap-lab\sa16-062$'
S-1-5-21-228093553-1527544583-567409058-1125
```

2. Retrieve the current SDDL for the Services Control Manager.

From the Windows command prompt, run the following command to retrieve the current SDDL for the Services Control Manager. The SDDL is saved in the file called *file.txt*.

```
sc sdshow scmanager > file.txt
```

The SDDL looks something like this. For more information, see [Microsoft KB914392](#).

```
D:(A;;CC;;;AU)(A;;CCLCRPRC;;;IU)(A;;CCLCRPRC;;;SU)(A;;CCLCRPWPRC;;;SY)(A;;KA;;;BA)S:
(AU;FA;KA;;;WD)(AU;OIIOFA;GA;;;WD)
```

3. Modify the SDDL.

Copy the section of the SDDL that ends in IU (Interactive Users). This section is one complete bracketed clause, i.e. (A;;CCLCRPRC;;;IU). Paste this clause directly after the clause you copied.

In the following text, replace *IU* with the user SID of the *Spotlight User*.

The new SDDL looks something like the following:

```
D:(A;;CC;;;AU)(A;;CCLCRPRC;;;IU) (A;;CCLCRPRC;;;S-1-5-21-214A909598-1293495619-
13Z157935-75714)(A;;CCLCRPRC;;;SU)(A;;CCLCRPWPRC;;;SY)(A;;KA;;;BA)
S:(AU;FA;KA;;;WD)(AU;OIIOFA;GA;;;WD)
```

4. Set the security credentials for accessing the Service Control Manager.

The *sdset* command on *sc* sets the security credentials for accessing the Service Control Manager (*scmanager*). Note the permissions on *scmanager* are being replaced. Setting security credentials is not additive. That is why we needed to copy the existing permissions.

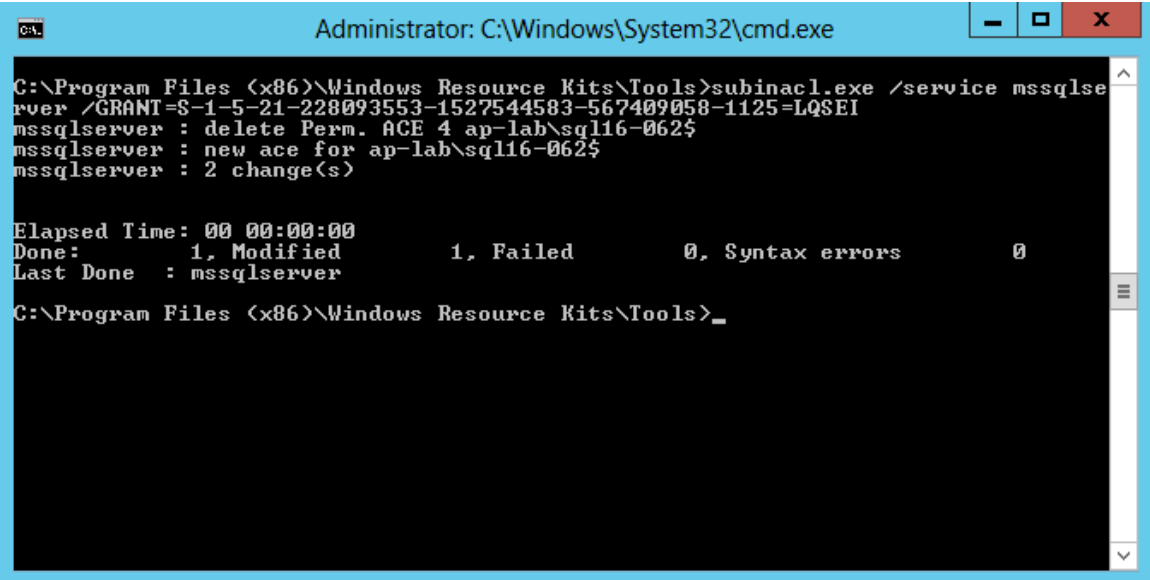
```
sc sdset scmanager
"D:(A;;CC;;;AU)(A;;CCLCRPRC;;;IU)(A;;CCLCRPRC;;;SU)(A;;CCLCRPWPRC;;;SY)(A;;KA;;;BA)(
A;;CCLCRPRC;;;S-1-5-21-214A909598-1293495619-13Z157935-
75714)S:(AU;FA;KA;;;WD)(AU;OIIOFA;GA;;;WD)"
```


5. Set the rights on the SQL server, SQL agent and SQL Full-text Filter Daemon Launcher services by using the [Command-Line Tool SubInACL](#) utility for the user SID of the *Spotlight User*. Run the utility with the following options:

```
subinacl.exe /service mssqlserver /GRANT= S-1-5-21-214A909598-1293495619-132157935-75714=LQSEI
```

```
subinacl.exe /service sqlserveragent /GRANT= S-1-5-21-214A909598-1293495619-132157935-75714=LQSEI
```

```
subinacl.exe /service mssqlfdlauncher /GRANT= S-1-5-21-214A909598-1293495619-132157935-75714=LQSEI
```



```
Administrator: C:\Windows\System32\cmd.exe
C:\Program Files (x86)\Windows Resource Kits\Tools>subinacl.exe /service mssqlserver /GRANT=S-1-5-21-228093553-1527544583-567409058-1125=LQSEI
mssqlserver : delete Perm. ACE 4 ap-lab\sql16-062$
mssqlserver : new ace for ap-lab\sql16-062$
mssqlserver : 2 change(s)

Elapsed Time: 00 00:00:00
Done:         1, Modified          1, Failed          0, Syntax errors          0
Last Done : mssqlserver

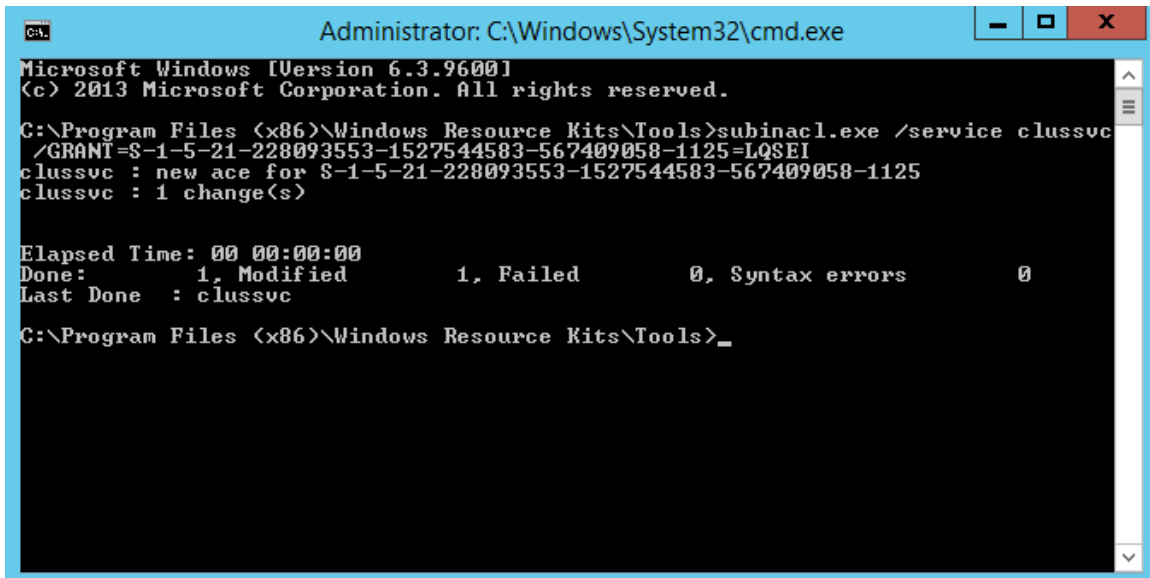
C:\Program Files (x86)\Windows Resource Kits\Tools>_
```

The following rights have the following meaning:

- L: Read control
- Q: Query Service Configuration
- S: Query Service Status
- E: Enumerate Dependent Services
- I: Interrogate Service

6. Set the rights on the ClusSvc (Cluster Service) by using the [Command-Line Tool SubInACL](#) utility for the user SID of the *Spotlight User*. Run the utility with the following options:

```
subinacl.exe /service clussvc /GRANT= S-1-5-21-214A909598-1293495619-13Z157935-75714=LQSEI
```



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\Windows Resource Kits\Tools>subinacl.exe /service clussvc
/GRANT=S-1-5-21-228093553-1527544583-567409058-1125=LQSEI
clussvc : new ace for S-1-5-21-228093553-1527544583-567409058-1125
clussvc : 1 change(s)

Elapsed Time: 00 00:00:00
Done:         1, Modified         1, Failed         0, Syntax errors         0
Last Done   : clussvc

C:\Program Files (x86)\Windows Resource Kits\Tools>_
```

Use a Registry Key to manage Remote Access to the Registry

Create a registry key to manage remote access to the registry. If you need to create the key to restrict access to the registry, follow these steps:

1. Start Registry Editor (Regedt32.exe), and then locate the following key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control
2. In the Edit menu, click Add Key, and then enter the following values:
Key Name: SecurePipeServers
Class: REG_SZ
3. Locate the following key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ SecurePipeServers
4. In the Edit menu, click Add Key, and then enter the following values:
Key Name: winreg
Class: REG_SZ
5. Locate the following key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ SecurePipeServers\winreg
6. In the Edit menu, click Add Value, and then enter the following values:
Value Name: Description


















Data Type: REG_SZ
String: Registry Server

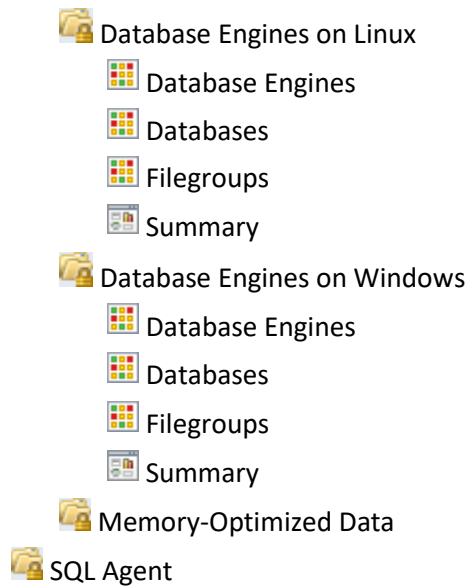
7. Locate the following key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg
8. Right-click winreg, click Permissions, and then edit the current permissions or add the users or groups to whom you want to grant access.
9. Quit Registry Editor, and then restart Windows.

View Information in the Operations Manager Console

Version-Independent (Generic) Views and Dashboards

This management pack introduces common folder structure, which will be used by future releases of management packs for different components of SQL Server. The following views and dashboards are version-independent and show information about all versions of SQL Server:

-  Microsoft SQL Server 2017+
 -  Active Alerts
 -  Computers
 -  SQL Server Roles
 -  Summary
 -  Task Status
 -  Integration Services
 -  SQL Server Database Engines
 -  Active Alerts
 -  All Performance Data
 -  Summary
 -  Task Status
 -  Always On High Availability
 -  Database Engines
 -  Database Engines
 -  Databases
 -  Filegroups



 **Note**

The “Computers” view displays the computers on which the agents are installed and the management pack discovery is running. Note that this view does not display computers configured for agentless monitoring.

“SQL Server Roles” dashboard provides an information about all instances of SQL Server Database Engine, SQL Server Reporting Services, SQL Server Analysis Services and SQL Server Integration Services:

SQL Server Roles

Instances (55)

Icon	Health	Maintenance Mode	Display Name	Path	Instance Type
			MSSQLSERVER	SQL12-051LONGNAME.KDV.local	Reporting Services
			MSSQLSERVER	SQL12-051LONGNAME.KDV.local	DB Engine
			MSSQLSERVER	SQL12-051LONGNAME.KDV.local	Analysis Services
			MSSQLSERVER	SQL14-093LONGNAME.KDV.local	DB Engine
			MSSQLSERVER	SQL12-048.KDV.local	DB Engine
			MSSQLSERVER	SQL12-048.KDV.local	Analysis Services
			MSSQLSERVER	SQL14-089.KDV.local	DB Engine
			MSSQLSERVER	SQL12-048.KDV.local	Reporting Services
			MSSQLSERVER	SQL2016RTM.KDV.local	DB Engine
			SQL2012EXPRESS	SQL12-048.KDV.local	DB Engine
			SQL2014EXPRESS	SQL14-089.KDV.local	DB Engine
			SQL2014EXPRESS	SQL14-093LONGNAME.KDV.local	DB Engine
			SQL2012EXPRESS	SQL12-051LONGNAME.KDV.local	DB Engine
			SQLEXPRESS	SQL2K8R2-046.KDV.local	DB Engine
			SQLEXPRESS	SQL2016RTM.KDV.local	DB Engine

For more information, see the guide to Microsoft SQL Server dashboards.

SQL Server Views

The Management Pack for Microsoft SQL Server introduces the comprehensive set of state, performance and alert view, which can be found in the dedicated folder:

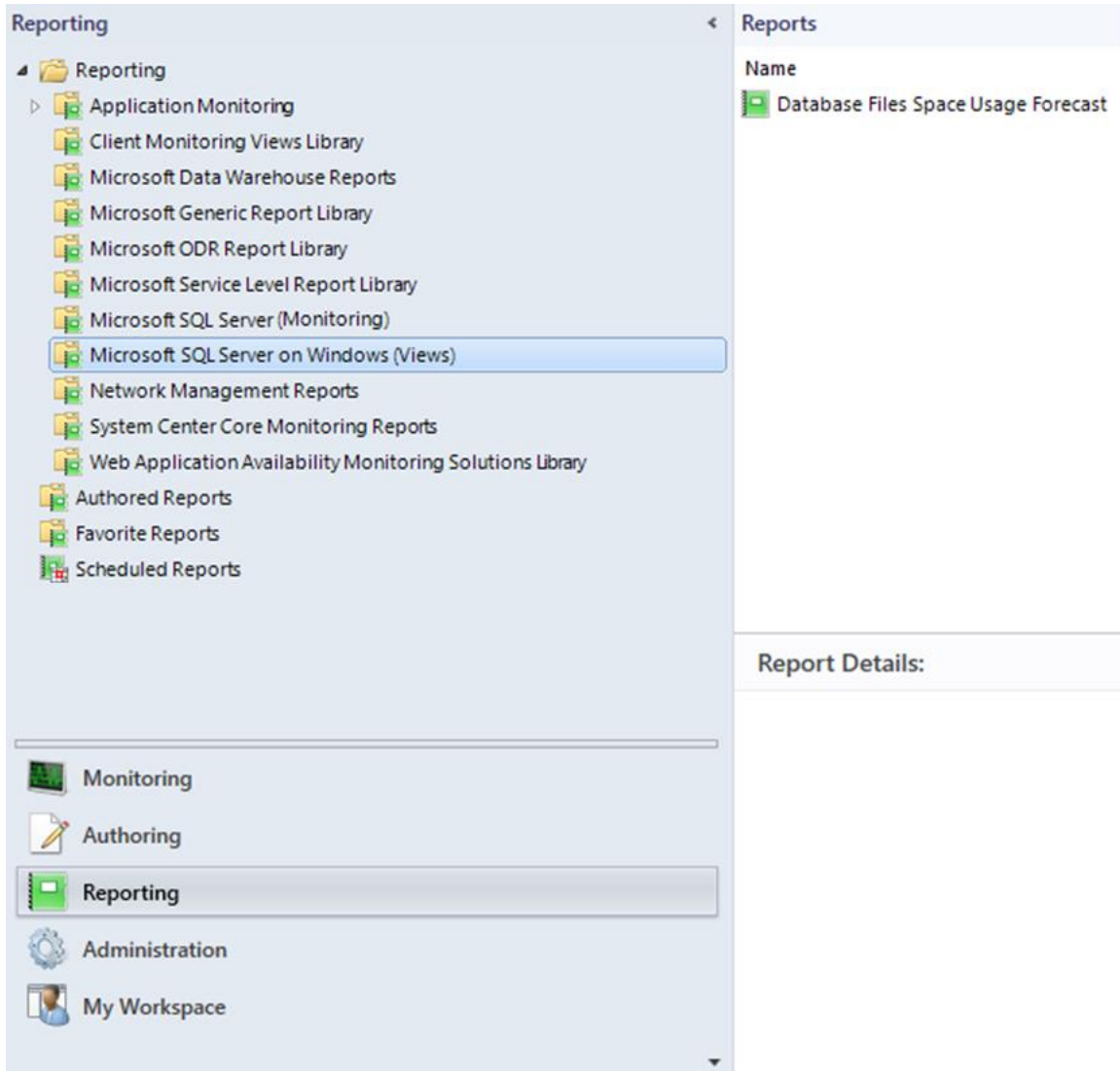
- Monitoring
 - Microsoft SQL Server 2017+
 - SQL Server Database Engines

Note

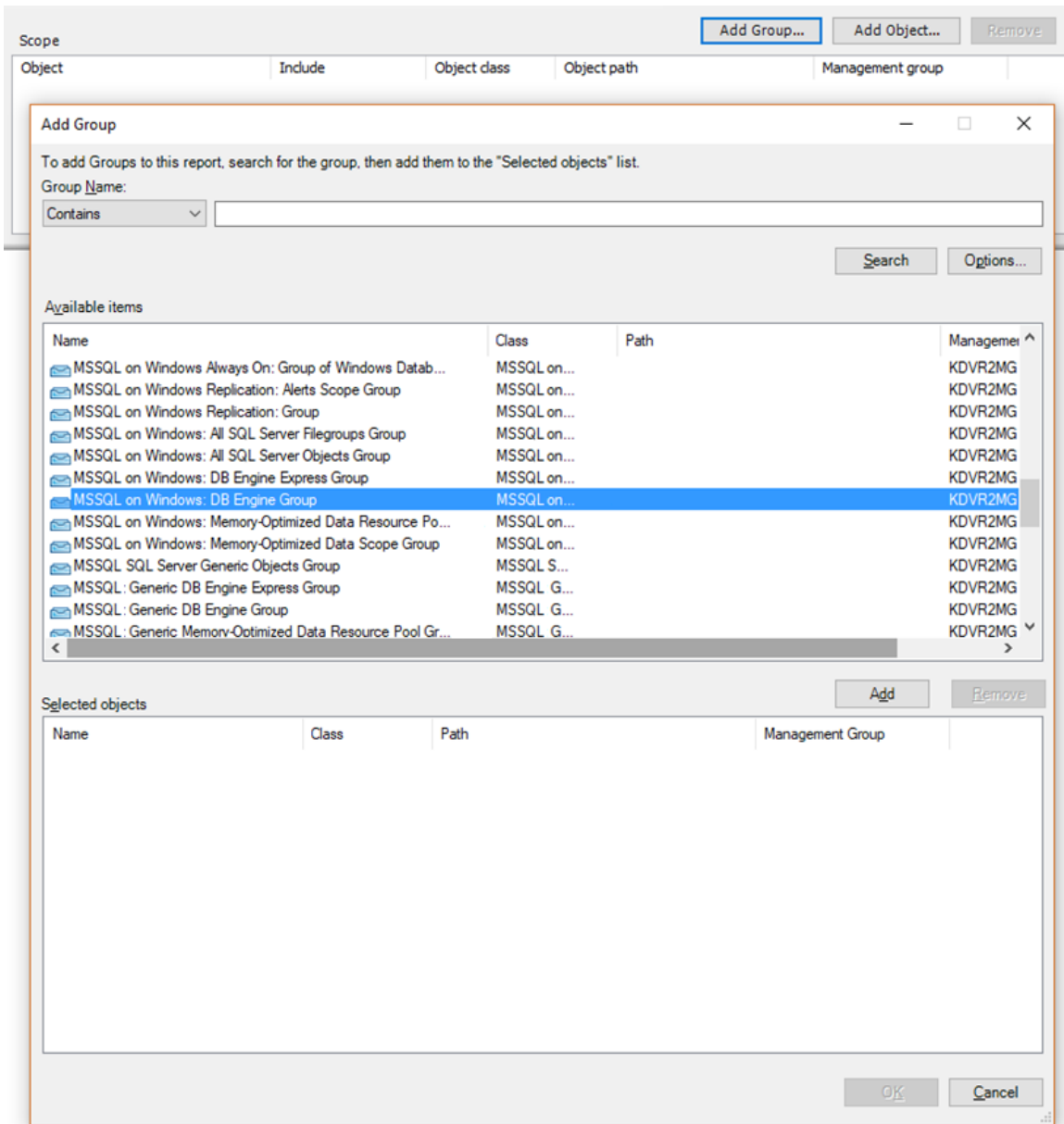
Some views may contain a very long list of objects or metrics. To find a specific object or group of objects, you can use the **Scope**, **Search**, and **Find** buttons on the Operations Manager toolbar. For more information, see “[Finding Data and Objects in the Operations Manager consoles](#)” article in the Operations Manager Help.

SQL Server Reporting

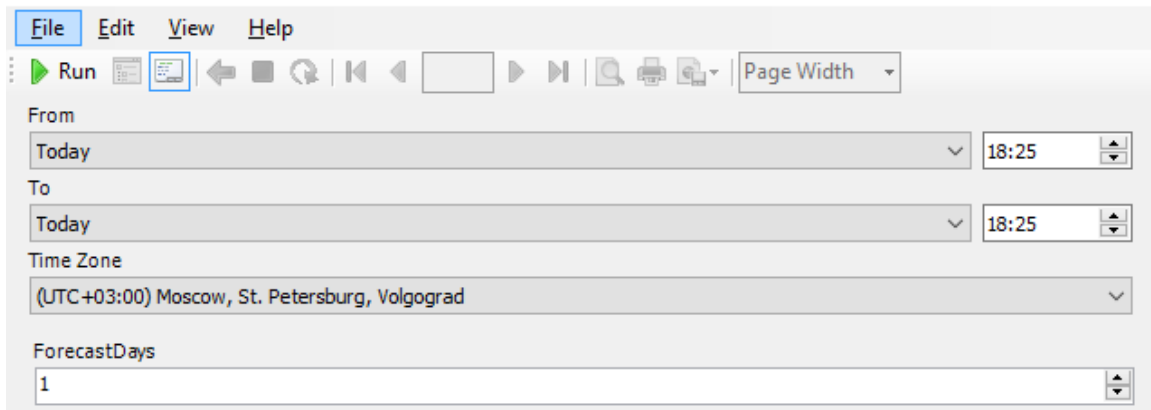
The Management Pack for Microsoft SQL Server introduces Database Files Space Usage Forecast report available in the corresponding section of the Operations Manager:



To open the report menu, double-click the report. In this menu, you must add an object (or a group of objects) to the report:



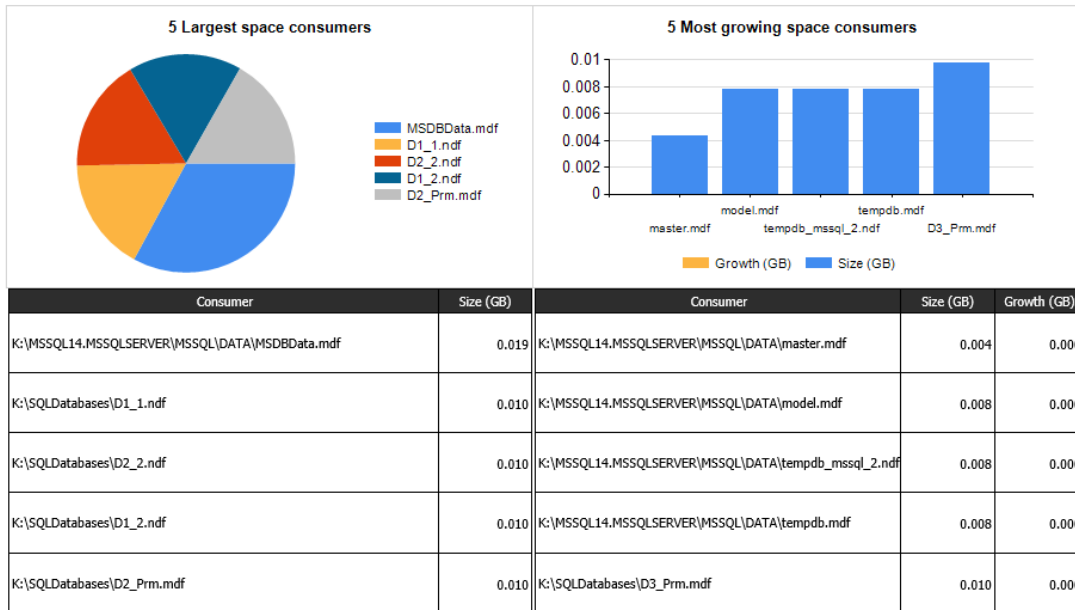
Then, select the period and the corresponding time zone for the report, and select the number of days for the file space consumption forecast:



Click the **Run** button to create the report. The report will display several charts with the following performance items:

- Initially consumed file space (GB)
- Finally consumed file space (GB)
- Initial average free file space (%)
- Final average free file space (%)
- File space consumption forecast (GB)

The report displays a separate chart for every selected object or a group of objects.



You can view the corresponding space usage forecast in a separate table:

Consumer	Initial Size (GB)	Growth (GB)	Final Size (GB)	Size forecast (GB)
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\imoltp.mdf	0.008	0.000	0.008	0.008
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\master.mdf	0.004	0.000	0.004	0.004
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\model.mdf	0.008	0.000	0.008	0.008
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\MSDBData.mdf	0.019	0.000	0.019	0.019
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\ServiceBrokerTest.mdf	0.008	0.000	0.008	0.008
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\tempdb.mdf	0.008	0.000	0.008	0.008
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\tempdb_mssql_2.ndf	0.008	0.000	0.008	0.008
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\TestDB_onCluster.mdf	0.008	0.000	0.008	0.008
K:\SQLDatabases\D1_1.ndf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D1_2.ndf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D1_Primary.mdf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D2_1.ndf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D2_2.ndf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D2_Primary.mdf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D3_1.ndf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D3_2.ndf	0.010	0.000	0.010	0.010
K:\SQLDatabases\D3_Primary.mdf	0.010	0.000	0.010	0.010

Note that this report works with Windows objects only.

Appendix: Known Issues and Troubleshooting

Rules and monitors may provide incorrect data if the default interval override values are changed

Issue: If the value of Interval (seconds) overridable parameter is set lower than the default value, rules and monitors may provide incorrect data.

Resolution: Make sure that Interval (seconds) overridable parameter is set no lower than the default value.

Seed discovery of a deleted platform pack may be still working on the pool nodes

Issue: An error may occur when a platform pack was deleted, but its seed discovery is still working on the pool nodes.

Resolution: Upon deletion of a platform pack, delete the corresponding seed discovery manually.

“Database Status” monitor is constantly changing its status

Issue: If “Auto Close” parameter for the database is set to “True”, “Database Status” monitor is constantly changing its status from “Healthy” to “Recovering/Restoring” and vice versa according to the timeout set in the override parameters.

Resolution: In view of the monitoring operation specifics, no resolution is required.

Enabling of “Auto Close” database parameter blocks collection of the performance metrics

Issue: If “Auto Close” parameter for the database is set to “True”, all performance rules return empty values.

Resolution: Set “Auto Close” database parameter back to “False”.

If a machine containing a monitored agentless instance is not available, multiple errors occur in the watcher node event log

Issue: If a machine containing a monitored agentless instance is not available, multiple SQL Server Monitoring MP Windows and SQL Server Discovery MP Windows errors occur in the watcher node event log. The errors will keep coming until the machine is available.

Resolution: No resolution available.

Some issues may occur upon installation of the management pack

Issue: The log reader may begin scanning the whole log of the SQL Server, which may lead to triggering of all alerts found. At that, RepeatCount property may contain excess number of events.

Resolution: No resolution available.

Double quotes in a database name may cause database console tasks failures

Issue: Database console tasks take database names enclosed in double quotes as one of their arguments. A database name may contain any symbol including double quotes. If it does, the console tasks for this database will not work.

Resolution: No resolution.

When an instance is not available, Module.Monitoring.Performance.MSSQLLogReaderEventTrigger exception is received in the event log

Issue: When an instance is not available, Module.Monitoring.Performance.MSSQLLogReaderEventTrigger exception is received in the event log. This exception will keep coming until the instance is available. The interval of this exception coming is equal to the lowest interval set for the alert rules.

Resolution: No resolution.

Odd behavior of the monitors' operational states

Issue: If the resource pool contains more than one management server, the operational states of all the monitors will be changing according to the failover settings of the resource pool.

Resolution: No resolution.

SQL Server on Docker: multiple errors occur after reboot of the Docker

Issue: Multiple errors occur after reboot of the SQL Server on Docker because Docker-ID (MachineName) is changed after the reboot.

Resolution: In SCOM, go to the monitoring template properties, open Service Details tab and click "Retry Connection".

Monitoring errors 40 and 121 may occur

Issue: Monitoring errors 40 and 121 may sporadically occur in the event log.

Resolution: No resolution.

Extended discovery intervals

Issue: in case of using a resource pool with several watcher nodes, the discovery intervals may be significantly extended.

Resolution: No resolution.

None of the event rules works on localized SQL DB Engines

Issue: None of the event rules works on localized SQL DB Engines. In current implementation, these rules work with English version only.

Resolution: No resolution.

Cluster instances maintenance mode is not available

Issue: In current implementation, maintenance mode does not work for 2017 cluster instances.

Resolution: No resolution.

Deleted policies are displayed in the Operations Manager.

Issue: SQL Server on Windows/Linux Policies deleted in SQL Server Management Studio are still displayed in the Operations Manager.

Resolution: No resolution.

Console tasks for Availability Group objects with names containing double-quota character do not work

Issue: Double-quota character may not be used in names of Availability Group and Databases in the Availability Group (for Always On). Therefore, console tasks for objects with such names do not work.

Resolution: No resolution.

Connection fails when IP address is specified as a connection string for a Linux-based instance.

Issue: When adding a Linux-based instance ("Add Instances" step of the Add Monitoring Wizard), the connection test fails if IP address is specified as a connection string and the authentication type is "Windows AD credentials".

Resolution: Specify the name of the machine as a connection string and use correct authentication type.

SCOM issue: Configuration Service may be frozen after Management Pack re-installation.

Issue: Configuration Service may be frozen after Management Pack re-installation. This appears to be a SCOM issue.

Resolution: No resolution.

"Database Critical Policies Availability" monitor may not change its status to "Critical" state.

Issue: "Database Critical Policies Availability" monitor may not change its status to "Critical" state when an issue with availability of Database Critical Policies hosted on this Database occurs.

Resolution: No resolution.

"Out of memory" errors are received in the Operations Manager

Issue: "Out of memory" errors are regularly received in the Operations Manager while the server has plenty of memory, and the instances are part of an Availability Group.

Resolution: Isolate the SQL Server WMI provider and increase the UploadTimeout.

To isolate the provider in its own host, follow the steps below from an elevated PowerShell:

```
$a =  
[WMI]'Root\Microsoft\SqlServer\ComputerManagement14:__Win32Provider.name="MSSQL_ManagementProvider"  
$a.HostingModel = "NetworkServiceHost:SQL"  
$a.put()
```

To revert the change:

```
$a =  
[WMI]'Root\Microsoft\SqlServer\ComputerManagement14:__Win32Provider.name="MSSQL_ManagementProvider"  
$a.HostingModel = "NetworkServiceHost"  
$a.put()
```

To increase the unload timeout to 30 minutes, follow these steps:

- Open WBEMTEST.
- Click the "Connect" button.
- In the "Namespace", enter *Root\Microsoft\SqlServer\ComputerManagement14*, and then click the "Connect" button.
- Click the "Query" button.
- Enter *select * from __win32provider where name = 'MSSQL_ManagementProvider'*, then click the "Apply" button.
- Double-click the resulting row.
- Double-click the "UnloadTimeout" value.
- Select "Not NULL" level, enter *0000000003000.000000:000*, and then click the "Save Property" button.
- Click the "Save Object" button.
- Click the "Close" button.

Errors occur in workflows related to Memory-Optimized Data databases

Issue: The following errors occur in workflows related to Memory-Optimized Data for databases with the “AutoClose” parameter set to “True”:

"Database is being recovered. Waiting until recovery is finished."

Resolution: No resolution.