Guide to Microsoft System Center Management Pack for SQL Server

Published in August 2020 by Microsoft Corporation.

This guide is based on version 7.0.24.0 (RTM) of the Management Pack for Microsoft SQL Server.

The Operations Manager team encourages you to provide 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.

Table of Contents

Changes History

August 2020 - 7.0.24.0 RTM

- **What's New**
    - Added a new "Securables Configuration Status" monitor targeted to SQL Server databases
    - Updated the "Product Version Compliance" monitor with the most recent versions of public updates for SQL Server
    - Updated the "Securables Configuration Status" monitor targeted to the DB Engine when a SQL Server instance participates in Availability Groups
    - Removed the "Securables Configuration Status" monitor targeted to the Availability Replica as non-useful
    - Updated the "SQL Server Database Engines" discovery; the “Netbios Computer Name” property is now uppercased.
    - Updated display strings
- **Bug Fixes**
    - Fixed the Alerting Rules data source to avoid an alert storm after exiting maintenance mode
    - Fixed the SQL Log Reader data source to support changing of the SQL Authentication method
    - Fixed the Performance Reader data source to support changing of the SQL Authentication method

June 2020 - 7.0.23.0 CTP

- **What's New**
    - Added reports from version-specific management packs for SQL Server
    - Updated monitor “Job Duration” to add current job run's duration to its alert description
    - Updated Web Console version of SQL MP Dashboards to support SCOM 2019 UR1
    - Updated monitor “Product Version Compliance” with versions of most recent public updates to SQL Server
    - Updated data source of alerting rules to avoid alert storm after exiting maintenance mode
    - Updated alert description of monitor “Securables Configuration Status”
    - Added “CheckStartupType” property to SSIS Health Status monitor

  - Revised columns of SQL Agent and SQL Agent Jobs state views
  - Updated display strings

- **Bug Fixes**

  - Fixed issue in data source of SPN Status monitor that may lead to memory leak
  - Fixed error “Unsupported path format” in workflows targeting Filegroups
  - Fixed discovery error on non-readable availability replicas
  - Fixed wrong Run As profile in SSIS Seed Discovery
  - Fixed issue that caused rule “Disable Discovery of Selected DB Engines” to fail
  - Fixed discovery issue for databases in recovering state
  - Fixed issue in monitor “Securables Configuration Status” when it went critical on Shared-Memory-only SQL Servers

December 2019 - 7.0.20.0 RTM

Including changes made in the prior preview release — v.7.0.18, November 2019.

- **What's New**

  - Updated MP to support SQL Server 2019 RTM
  - Added filter by edition to “Local DB Engine Discovery”
  - Redesigned DB Space monitoring to improve performance: Enabled by default monitors and performance rules targeting Database which watch for disk space consumption by ROWS Filegroups and Logfiles
  - Redesigned DB Space monitoring: Added two monitors and two performance rules targeting Database to watch for disk space consumption by In-Memory and FILESTREAM data
  - Redesigned DB Space monitoring: Read-only filegroups now count as well
  - Redesigned DB Space monitoring: Disabled by default all workflows targeting Filegroups, Files, Logfiles
  - Redesigned XTP performance counters to make them completely version-agnostic
  - Added attribute “TCP Port” to “SQL DB Engine Class” and updated “DB Engine Discovery” to populate the new property
  - Added summary dashboard for SCOM 2019 Web Console (HTML5)
  - Added support for cluster nodes with disjoined namespaces
  - Added sampling to algorithm of monitor “WMI Health State” in order to eliminate false alerting on cluster SQL Server instances
  - Updated alert descriptions of monitors “Availability Database,” “Availability Replica,” and “Availability Group” (generating alerts still disabled by default)
  - Updated monitor “Product Version Compliance” with versions of most recent public updates to SQL Server
  - Disabled by default monitor “Buffer Cache Hit Ratio” and changed its threshold from 0% to 90%
  - Disabled by default monitor “Page Life Expectancy”
  - Removed monitors “Availability Database Join State” and “Availability Replica Join State” as not useful
  - Updated display strings
  - Revised columns on DB Engine state views

- **Bug Fixes**

- Fixed: monitor “Service Principal Name Configuration Status” raises false alerts because of case-sensitive comparison
- Fixed: “Local DB Engine Discovery” crashes when Windows has Turkish locale
- Fixed issue that caused performance degradation in workflows “General Always On Discovery,” “Database Replica Discovery,” and “Always On System Policy Monitoring”
- Fixed: “General Always On Discovery” throws errors on environments with several Distributed Availability Groups
- Fixed monitoring issue in case of Database is replicated by Always On Availability Group
- Fixed empty property bag when Availability Group has cluster type NONE
- Fixed wrong target in alerting rule “DB Backup Failed to Complete”
- Fixed rule “MSSQL Integration Services on Windows: The package restarted from checkpoint file” and its alert
- Fixed rule “OS Error occurred while performing I/O on pages“ and its alert
- Fixed: "DB Disk Write Latency" and "DB Disk Read Latency" monitors and performance rules get wrong performance metric
- Fixed alert description of monitor “WMI Health State”

Management Pack Scope and Supported Configurations

This management pack is version-agnostic, which means that it supports discovery and monitoring of SQL Server 2012 through 2019 and up, including SQL on Linux with SQL Server 2017 and up.

The management pack discovers and monitors SQL Server right out of the box when there are no version-specific management packs for SQL Server previously installed. If this management pack is installed in addition to the version-specific management packs for SQL Server 2008 and 2012, 2014, 2016, see Configuration with old management packs for SQL Server to get more information on such a configuration.

This section explains what SQL Server features are covered by this management pack, what configurations are supported, what monitoring features the management pack offers, and what prerequisites should be met to begin with this management pack.

Notes to Release

- **Former generation of the management packs for SQL Server 2012, 2014, and 2016 reached the end of support** This management pack is virtually a new version of the version-agnostic management pack for SQL Server 2017 and up, whose last version, 7.0.7.0, was released in July 2018. All the following versions of the management pack, including the current one, are for the monitoring of SQL Server 2012, 2014, and 2016 in addition to previously supported 2017 and up. The former generation of management packs for SQL Server 2008—2016 reached the end of support with the first public release of the management pack for SQL Server 2012 and up (April, 2019).

- **Upgradability issues of path SQL Server 2017+ MP v.7.0.7.0 → any version of the current management pack** This version has a great deal of significant changes in comparison to the last released version of SQL Server 2017+ MP. Some of these changes are so severe that lead to upgradability issues, however they all are necessary for this management pack to provide solid monitoring.

  After importing this management pack over the SQL Server 2017+ Management Pack, all already-discovered instances of SQL Server 2017 will be re-discovered. This will cause SCOM to “forget” historical data for these instances, which affects reporting.

Management pack for SQL Server 2017+ Integration Services cannot be upgraded and has to be removed before importing this update. Remove the following management pack (this file is part of the delivery of SQL Server 2017+ MP): *Microsoft SQL Server 2017+ Integration Services on Windows*.

- **Localization for SQL Server 2017+ MP cannot be imported over the current version of SQL Server MP**

  This management pack cannot be localized with the localization packs initially made for SQL Server 2017+ MP. If you already have SQL Server 2017+ MP (7.0.0.0 or 7.0.7.0) imported and localized, then you do not need to remove the localization pack before importing this management pack.

SQL Server Configurations and Features

Operating Systems and Platforms

List of supported operating systems/platforms is as following:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019
- Ubuntu 16.04 and 18.04
- Red Hat Enterprise Linux 7.3 and 7.4
- SUSE Linux Enterprise Server v12 SP2
- Docker Engine 1.8+
- Azure Kubernetes Service (AKS)

Localized versions of Windows Server are supported by Management Pack.

SQL Server Features

The management pack works with any edition of SQL Server, from Express to Enterprise. Here is the list of SQL Server features and configurations that the management pack supports (unsupported features and configurations are also on this list flagged as "Not supported").

- SQL Server Database Engine
- SQL Server Database, including filegroups, data files, transaction log files, FILESTREAM and Memory-Optimized Data containers. Different storage options for databases supported:
  - Local storage (both drive letters and mount points)
  - Cluster Shared Volumes
  - SMB Shares
  - Azure BLOBs
- SQL Server Agent and Jobs
- SQL Server Memory-Optimized Data (In-Memory OLTP)
- SQL Server High Availability Features
  - Single-domain Availability Groups, including availability replicas and database replicas
  - Distributed Availability Groups

    - Failover Clustering
    - Log Shipping
    - Replication — **Not supported**; use dedicated management packs for SQL Server Replication to monitor this feature.
    - Mirroring — **Not supported**
    - Domain-independent Availability Groups — **Not supported**
    - Workgroup Cluster Availability Groups — **Not supported**
- Authentication Mode — both SQL Server Authentication and Windows Authentication are supported.
- Localized versions of SQL Server — **Not supported**; Management Pack can only work with the English-language version of SQL Server.

SCOM Configurations

This management pack offers three modes of monitoring: Agent Monitoring, Agentless Monitoring, and Mixed Monitoring. See Monitoring Modes for more information on each of them. Agent Monitoring and Mixed Monitoring modes work with SQL on Windows only. Agentless Monitoring was initially designed to enable monitoring of SQL on Linux but it works with SQL on Windows as well. The management pack does not require a dedicated management group and can work in virtual environments. List of supported versions of SCOM is as following:

- System Center Operations Manager 2012 R2
- System Center Operations Manager 2016
- System Center Operations Manager 1801
- System Center Operations Manager 1807
- System Center Operations Manager 2019

Prerequisites

- **.NET Framework 4.5+**

  Installation of .NET Framework 4.5 or newer is required

- **Management Pack for Windows Server Operating System & Management Pack for UNIX and Linux Operating Systems**

  As a best practice, you should import the Windows or Linux Server Management Pack for the operating system you are using. The 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.

- **Removal of overrides for “SQL on Windows: Discover Installation Source (seed)”**

  In case of upgrading the management pack for SQL Server 2017+ to the current one, remove the overrides set for the “SQL on Windows: Discover Installation Source (seed)” discovery beforehand because this discovery has Interval (seconds) overridable parameter instead of Frequency (seconds) one.

- **“Allow log on locally” security policy for domain monitoring account is enabled**

If a domain account is used as the action account for this management pack, make sure to enable the “Allow log on locally” policy for it. See Enabling “Allow Log On Locally” Security Policy for more information.

- **Each agent has the Agent Proxy option enabled**

  Enable the Agent Proxy option for all agents that will use this management pack to monitor SQL Server. Agent Proxy setting allows an agent to forward data to the management server on behalf of another entity and it should be set enabled if agent workflow scenarios discover any non-hosted objects (the management pack creates a non-hosted object for every SQL Server instance).

- **SQL Server Connection Protocols**

  For the agent monitoring mode, all three protocols TCP/IP, "Named Pipes", and "Shared Memory" are supported. Keeping SQL Server Browser running is not a prerequisite for this monitoring mode. For the agentless monitoring mode, both protocols TCP/IP and "Named Pipes" are supported. SQL Server Browser should be enabled and running. For the mixed monitoring mode, only TCP/IP protocol is supported, and keeping SQL Server Browser running is not a prerequisite for this monitoring mode. See Monitoring Modes for more information on the monitoring modes provided by this management pack.

- **Removal of management pack “Microsoft SQL Server 2017+ Integration Services on Windows” before importing this management pack**

  Management pack for SQL Server 2017+ Integration Services cannot be upgraded and has to be removed before importing this management pack. Remove the following management pack (this file is part of the delivery of SQL Server 2017+ MP): Microsoft SQL Server 2017+ Integration Services on Windows

- ***Author* set of privileges on SCOM SDK**

  This management pack needs the Author set of privileges on the SCOM SDK to be able to create a management pack and store overrides in it. If the default action account on SCOM does not have these permissions, make sure to have an account granted with them and map this account to the Microsoft SQL Server SCOM SDK Run As Profile.

Management Pack Delivery

Management Pack delivers as a download on microsoft.com and is also available on the SCOM Online Catalog. The download provides the next files:

- SQLServerMP.Windows.msi — set of .MP and .MPB files to start monitoring SQL on Windows.
- SQLServerMP.Linux.msi — set of .MP and .MPB files to start monitoring SQL on Linux.
- SQLServerMPGuide.pdf — this operations guide.
- SQLServerDashboardsGuide.pdf — operations guide to SQL MP Dashboards.
- SQLServerMPWorkflowList.pdf — complete list of SQL Server MP workflows with descriptions and parameters.

Management Pack for Microsoft SQL Server includes the following files:

- Microsoft.SQLServer.Core.Library.mpb
- Microsoft.SQLServer.Core.Views.mp
- Microsoft.SQLServer.Core.WebDashboards.mp

- Microsoft.SQLServer.IS.Windows.mpb
- Microsoft.SQLServer.IS.Windows.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

> ⚠ Do not import Microsoft.SQLServer.Core.WebDashboards.mp into SCOM before 2019. This file contains the SQL Server MP Dashboards for the new Operations Manager Web Console introduced with SCOM 2019.

Importing Management Pack

For more information on how to import a management pack, see How to Import a Management Pack.

Disabled Space Monitoring Workflows for SQL on Linux

The following workflows are disabled because they are not provided with the necessary data by the SQL Server on Linux. We do not recommend enabling them.

- 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

Monitoring Configuration

When either Agent Monitoring or Mixed Monitoring mode is used, the management pack automatically discovers stand-alone and cluster instances of SQL Server across all managed systems that run the System Center Operations Manager Agent service. Agentless Monitoring mode requires manual configuration for each SQL Server instance to be monitored. These three monitoring modes are described in detail below in this guide. As a rule of thumb, choose Agent Monitoring mode for SQL Server on Windows and Agentless Monitoring mode for SQL Server on Linux.

You can customize the management pack by means of overriding the defaults. Your customizations cannot be stored in the SQL Server MP, so SCOM saves all the overrides in the Default management pack until you choose another management pack. It is a best practice to store all overrides for a management pack you want to customize in a dedicated management pack. We recommend to create at least one management pack named, for example, "Microsoft SQL Server Customizations" and save all the customizations for the SQL Server MP in it.

Coexistence of SQL Server 2008/2012/2014/2016 MPs and SQL Server MP

The management pack discovers and monitors SQL Server 2012 and up right out of the box when there are no version-specific management packs for SQL Server 2012, 2014, and/or 2016 previously installed (the old management packs). In the case when there is any of the old management packs detected during an import of the version-agnostic management pack, the latter disables the discovery and monitoring for those SQL Server versions already covered by the old management packs. It was made to avoid double monitoring.

How Disabling of Monitoring Works

The version-agnostic management pack runs the “MSSQL on Windows: Automatic setup of DB Engine discovery filter” action rule after importing. The rule does the following actions:

- Search for instances of SQL Server 2012, 2014 and 2016 discovered by the old management packs. If there is at least one instance of any of those SQL Server versions, this version will be disabled for discovery by the version-agnostic management pack.
- Overrides the “MSSQL on Windows: Discover SQL Server Database Engines (Local)” discovery by filling out the “SQL Server versions to be excluded” parameter with versions to disable and saves this override in a new management pack called “Microsoft SQLServer overrides.” See Disabling Monitoring of Specified SQL Server Versions to get more information.
- Disables itself and saves this override in the same management pack. Do not remove this management pack in order to keep this rule disabled.

If the old management packs are imported after importing the version-agnostic management pack, the monitoring provided by the latter will not be disabled.

How to Enable Monitoring of SQL Server 2012/2014/2016 by SQL Server MP

When you are ready to start off the monitoring of SQL Server 2012/2014/2016 by the version-agnostic management pack, remove or edit the override for the “MSSQL on Windows: Discover SQL Server Database Engines (Local)” discovery described in How Disabling of Monitoring Works. Removal of management pack “Microsoft SQLServer overrides” while still having the old management packs makes the rule to re-create this management pack, which will disable the SQL Server MP to discovery for SQL Server 2012/2014/2016.

Monitoring Modes

The management pack offers three monitoring modes:

- **Agent Monitoring Mode**—the monitoring is carried out by the SCOM Agent. This is the most common kind of monitoring; the old management packs for SQL Server, as well as the vast majority of all SCOM management packs, provide this kind of monitoring. It supports SQL on Windows only.
- **Agentless Monitoring Mode** is the monitoring mode that was originally designed to monitor SQL on Linux, however it supports both SQL on Linux and SQL on Windows. In this monitoring mode, the management pack's workflows run on management servers and gateway servers mapped to the SQL Server Monitoring Pool or All Management Servers Pool, if the first is not configured. This monitoring mode does not provide the automatical discovery of SQL Server instances and requires all the instances to be manually added to the monitoring. See Configuring Agentless Monitoring Mode for more information.
- **Mixed Monitoring Mode** is the hybrid of the Agent and Agentless modes. In this monitoring mode, the management pack places its seed on all computers where there is the SCOM Agent and uses this seed to automatically discover all SQL Server on Windows instances but the entire monitoring is carried out like in Agentless Monitoring mode—from Management Servers and Gateway Servers that are members of the SQL Server Monitoring Pool. It supports SQL on Windows only. See Configuring Mixed Monitoring Mode for more information.

All monitoring modes support both the SQL Server and the Windows authentication.

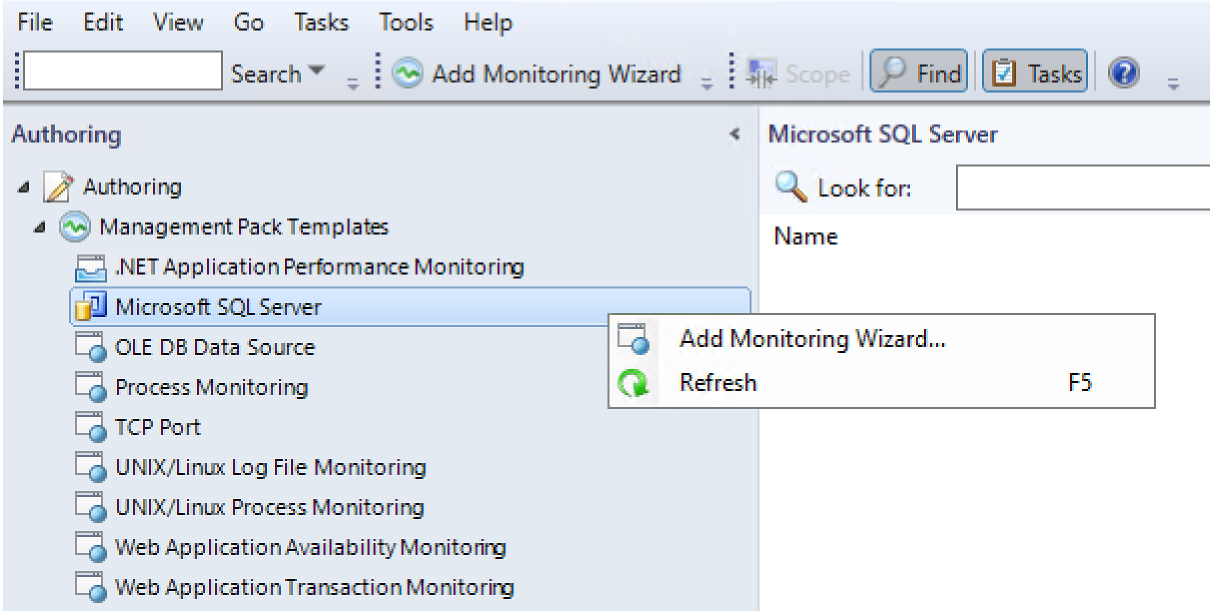
When Agent Monitoring or Mixed Monitoring mode is used, the management pack automatically discovers stand-alone and clustered instances of SQL Server across all managed systems that run the System Center Operations Manager agent service. For more information on how to discover Database Engine Instances in Agentless mode, see Configuring Agentless Monitoring Mode.

Because management pack database engine objects are unhosted, the **Path** column in the dashboard view will remain empty for all discovered objects in any Monitoring mode. Both the **Name** and the **Machine Name** columns are used to replace **Path** properties in the management pack. The **Name** column is used to display the Machine NetBIOS name and the SQL Server instance name in the following format: < Machine NetBIOS name\SQL Instance Name >. A **Machine Name** will display FQDN of the machine in case the computer joined a domain or the NetBIOS name for non-joined domain computers (e.g. Linux machines).

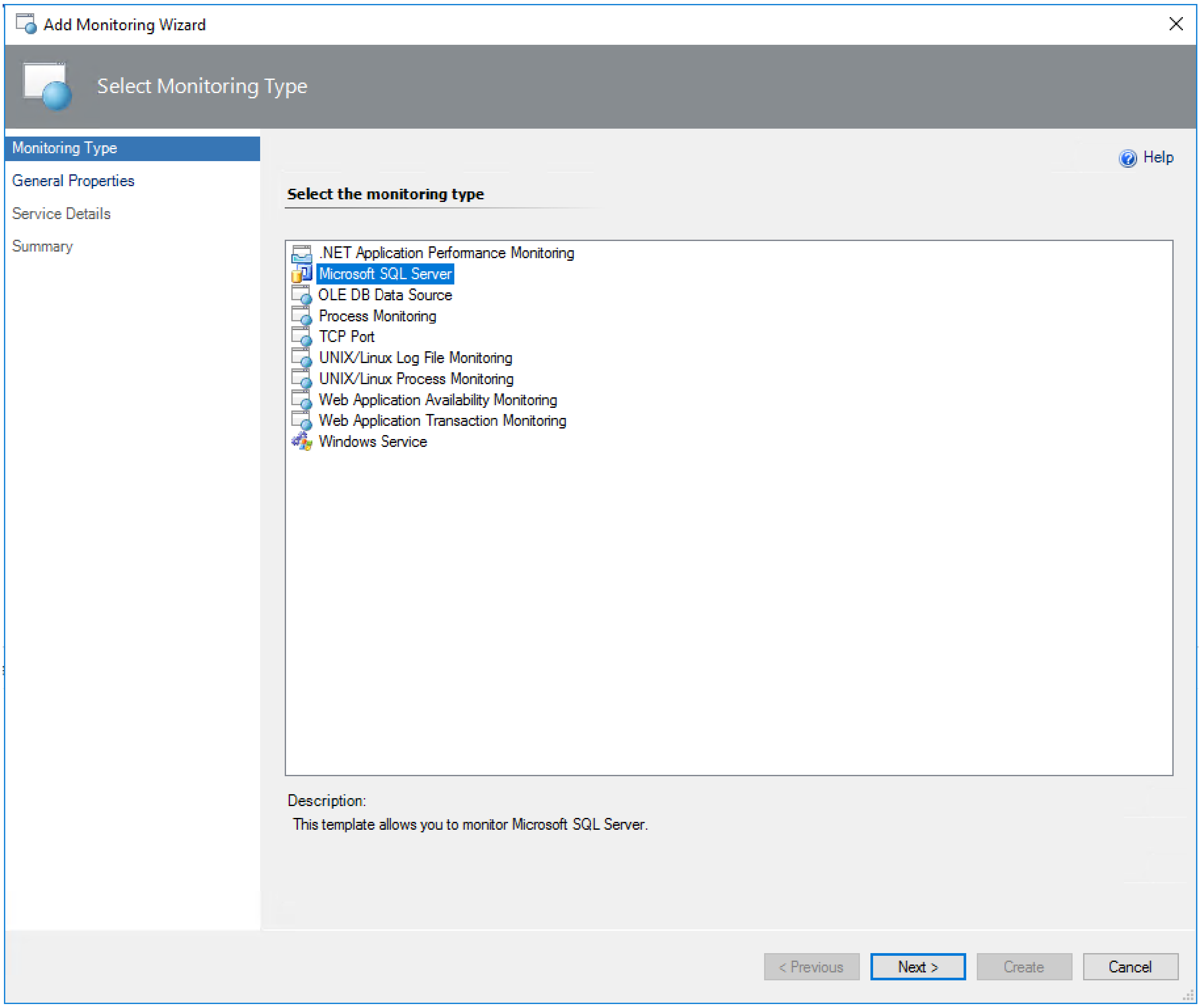
As Agentless Monitoring and Mixed Monitoring modes make the management pack workflows run on management servers, we recommend you not use them without prior test in a non-production environment to avoid unexpected overload of the servers. Having management servers dedicated to the monitoring of SQL Server is recommended as well. See Configuring SQL Server Monitoring Pool for more information.

Configuring Agentless Monitoring Mode

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

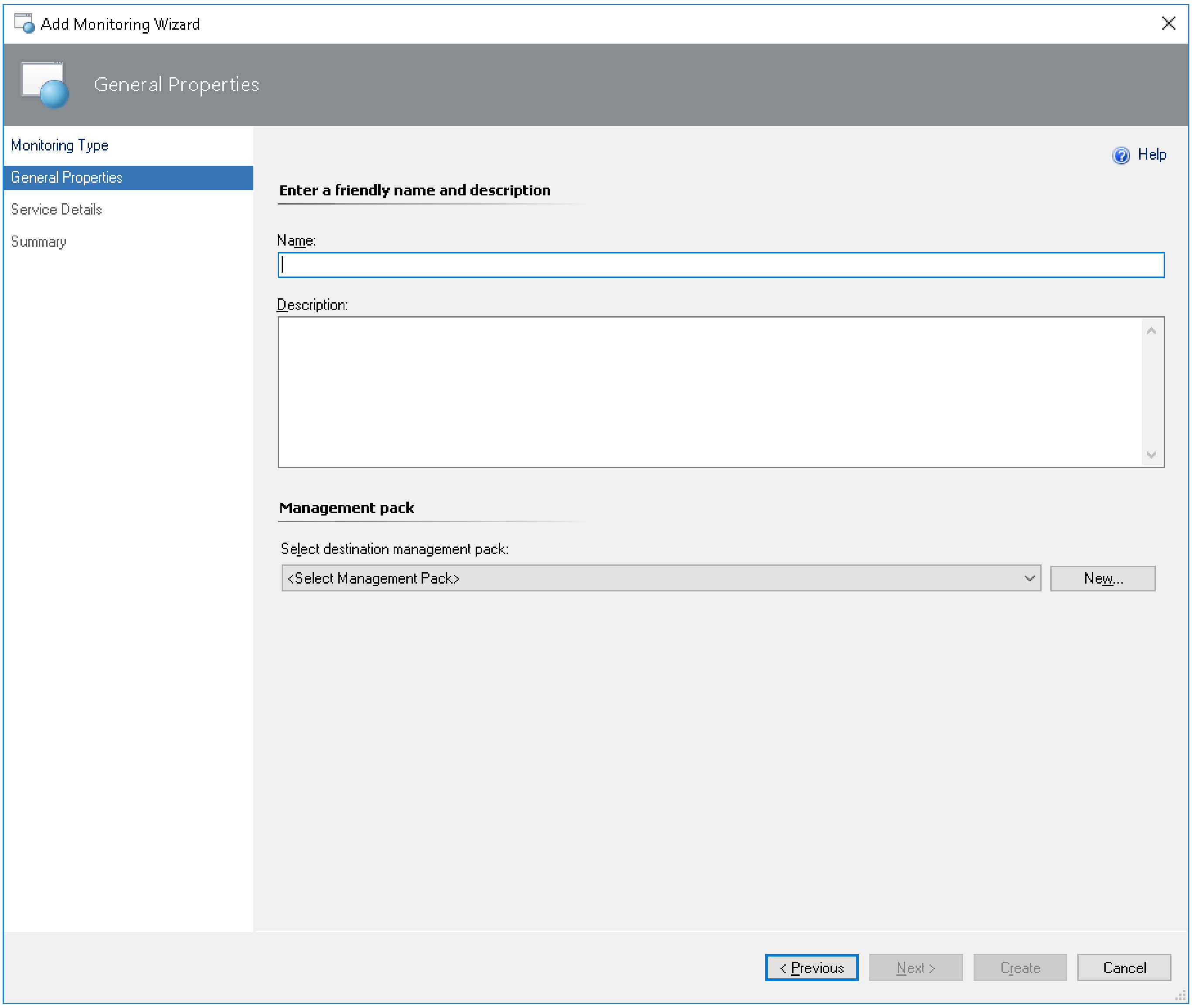

File Edit View Go Tasks Tools Help
Search
Add Monitoring Wizard
Scope
Find
Tasks
Authoring
Authoring
Management Pack Templates
.NET Application Performance Monitoring
Microsoft SQL Server
OLE DB Data Source
Process Monitoring
TCP Port
UNIX/Linux Log File Monitoring
UNIX/Linux Process Monitoring
Web Application Availability Monitoring
Web Application Transaction Monitoring
Microsoft SQL Server
Look for:
Name
Add Monitoring Wizard...
Refresh F5

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

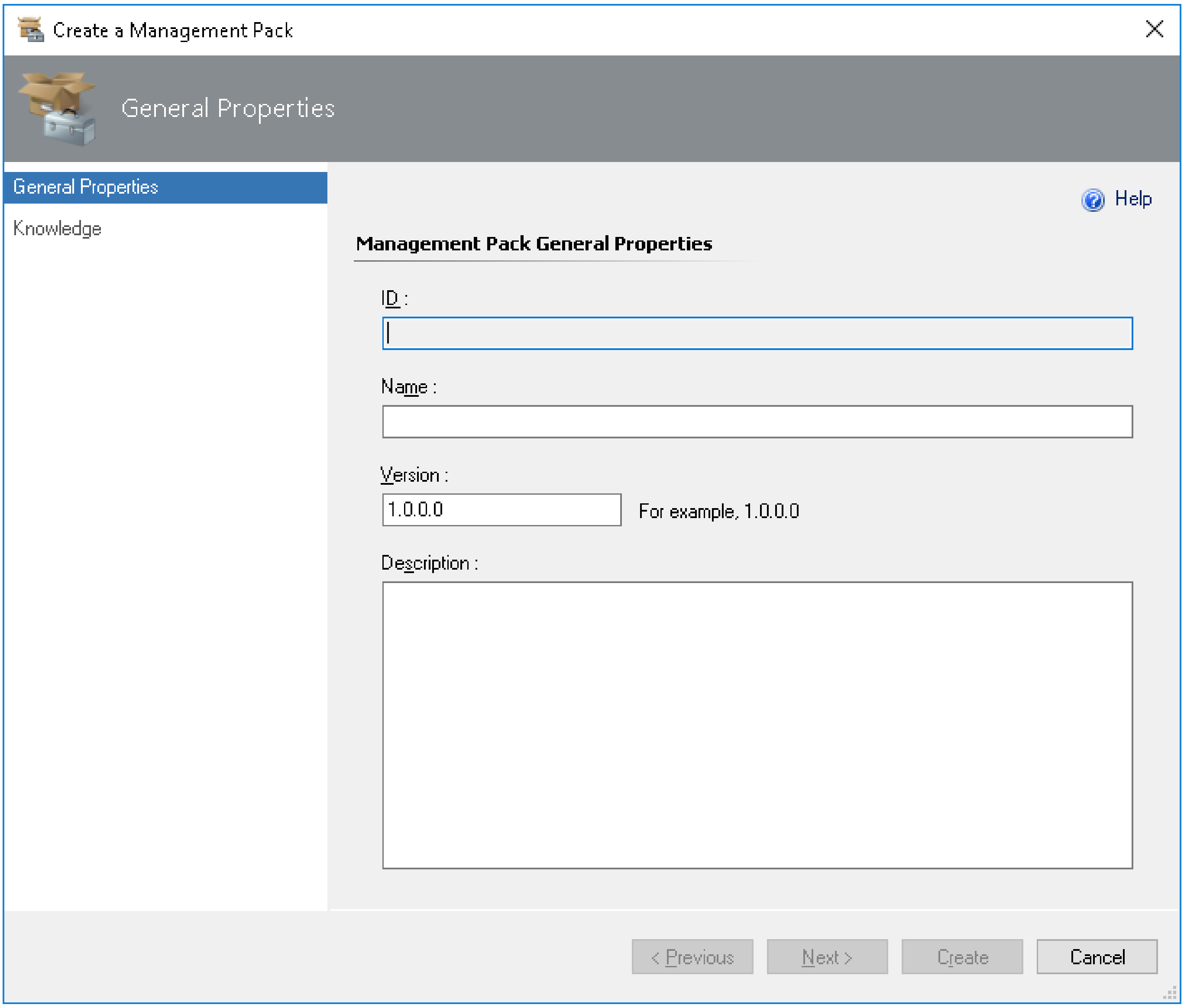


Description:
This template allows you to monitor Microsoft SQL Server.

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.



Management Pack General Properties

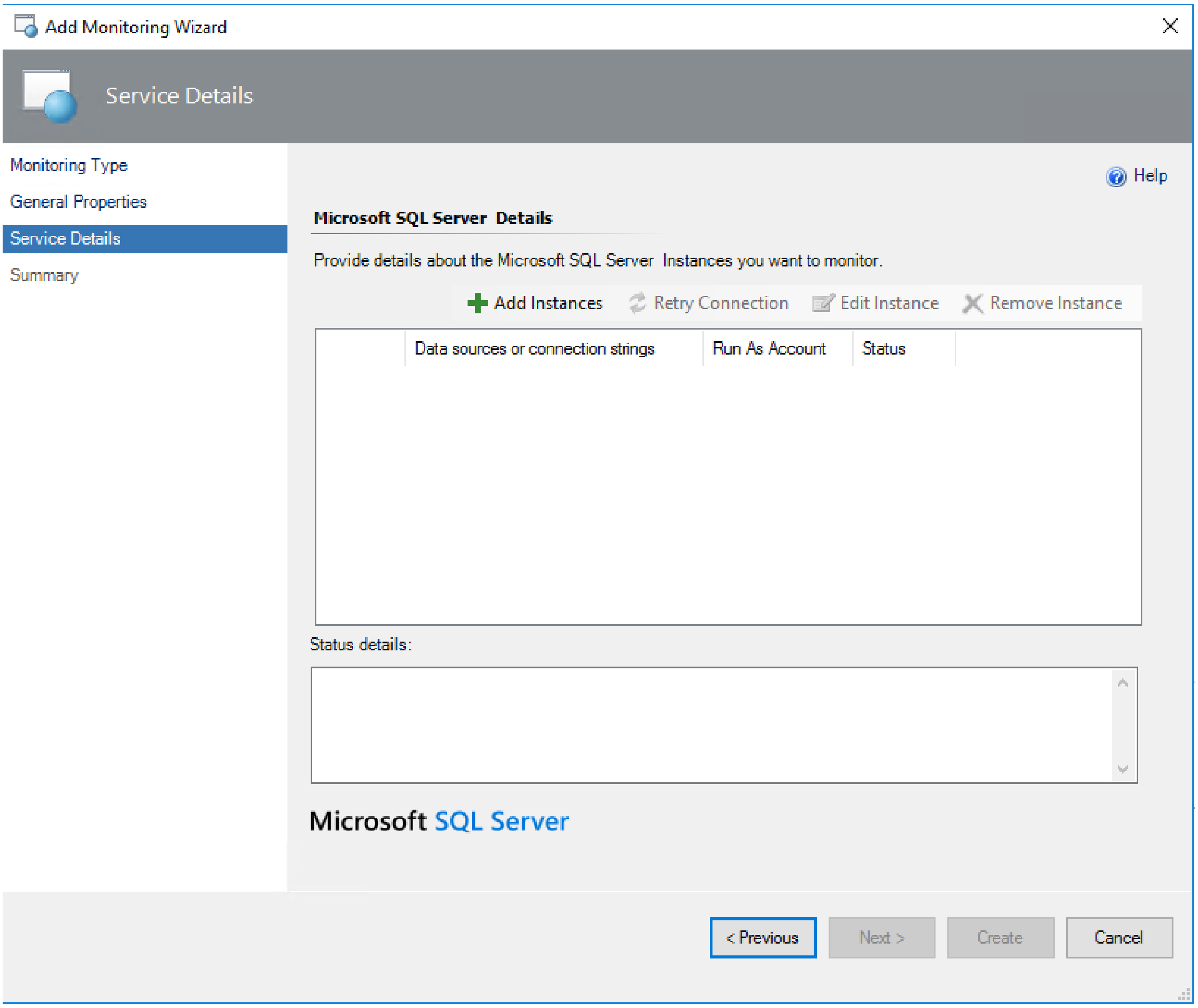
ID :

Name :

Version : 1.0.0.0 For example, 1.0.0.0

Description :

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



Add Monitoring Wizard

Service Details

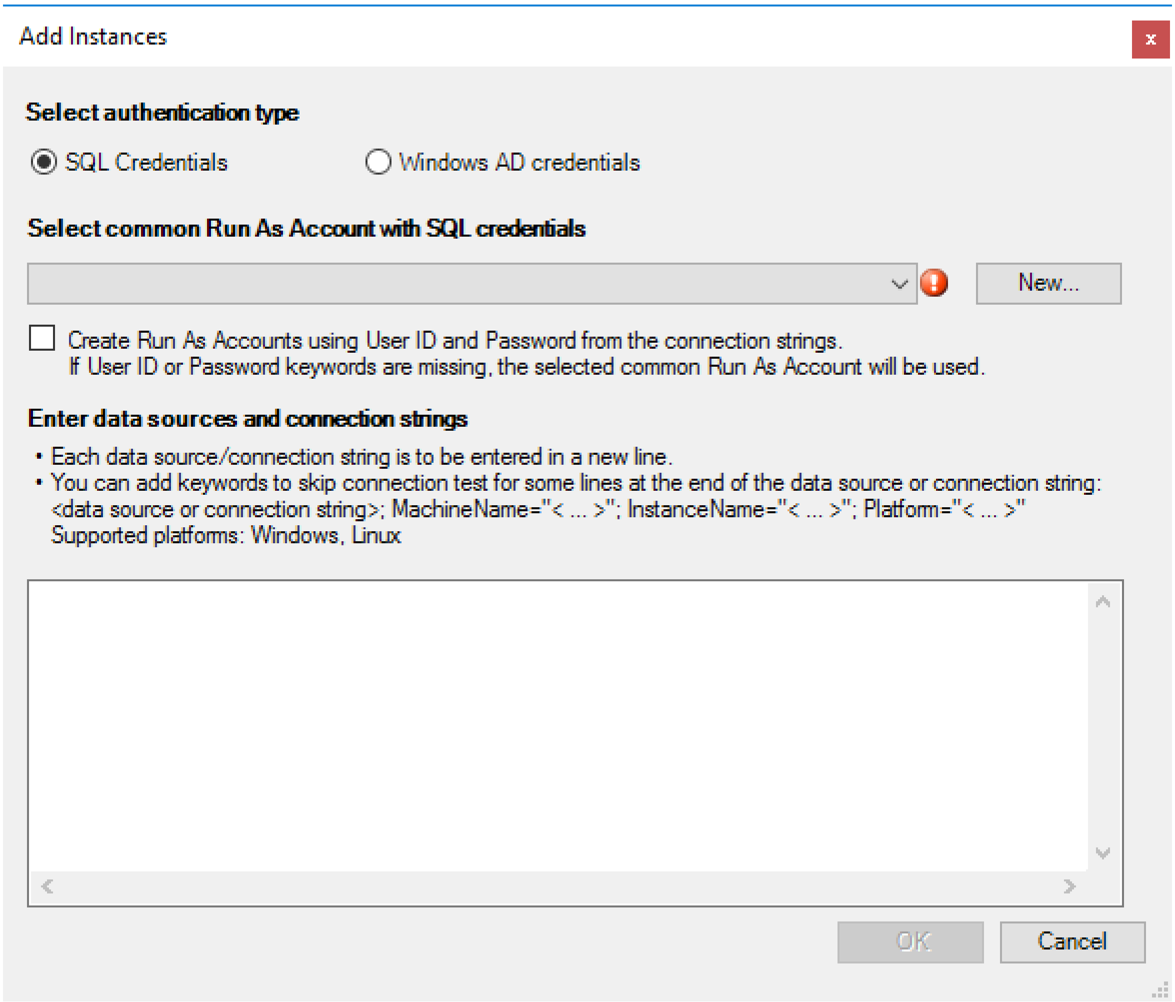
Microsoft SQL Server Details

Provide details about the Microsoft SQL Server Instances you want to monitor.

	Data sources or connection strings	Run As Account	Status

Status details:

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



Add Instances

Select authentication type

● SQL Credentials ○ Windows AD credentials

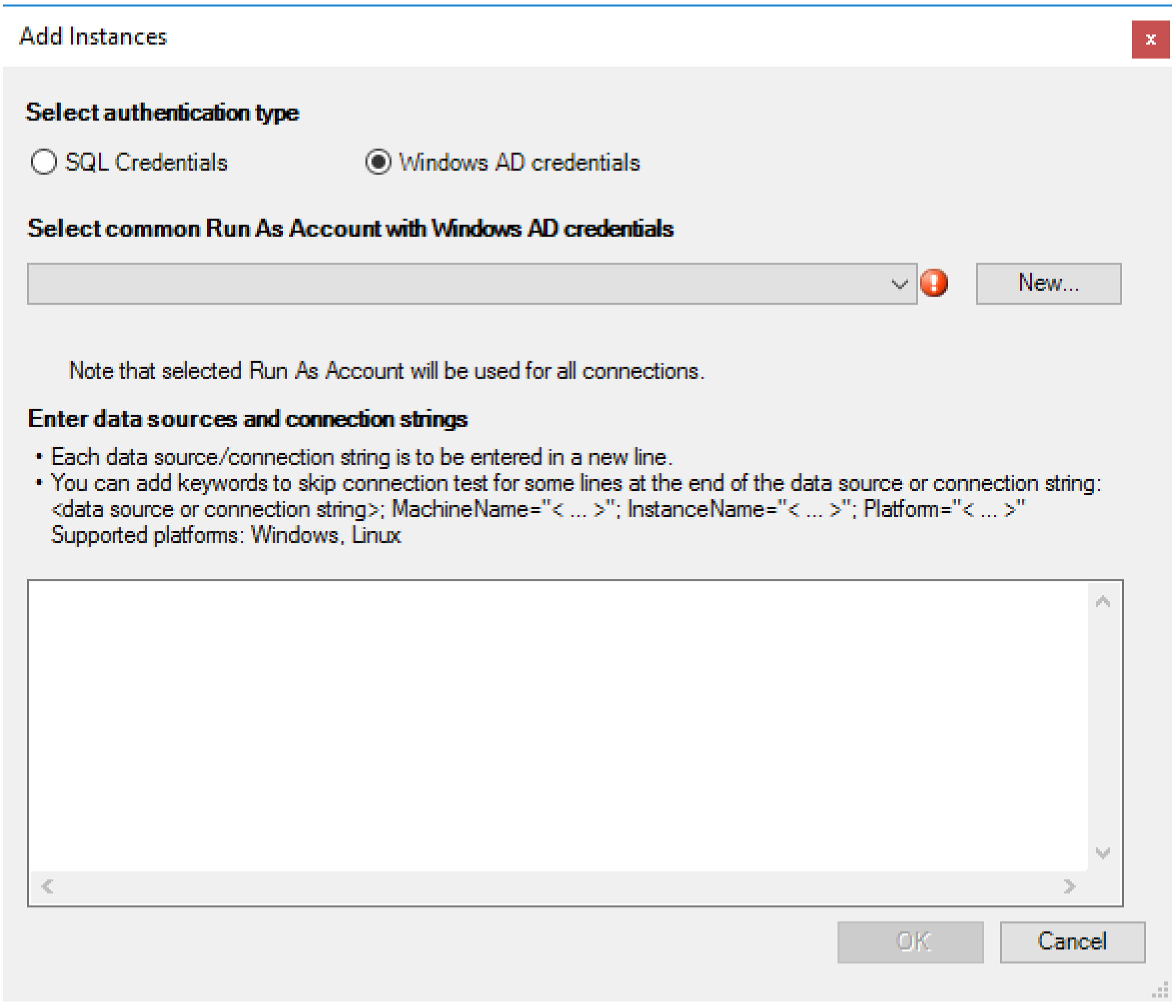
Select common Run As Account with SQL credentials

□ Create Run As Accounts using User ID and Password from the connection strings.
If User ID or Password keywords are missing, the selected common Run As Account will be used.

Enter data sources and connection strings

- Each data source/connection string is to be entered in a new line.
- You can add keywords to skip connection test for some lines at the end of the data source or connection string:
  <data source or connection string>; MachineName="< ... >"; InstanceName="< ... >"; Platform="< ... >"
  Supported platforms: Windows, Linux

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


Add Instances

Select authentication type

○ SQL Credentials ● Windows AD credentials

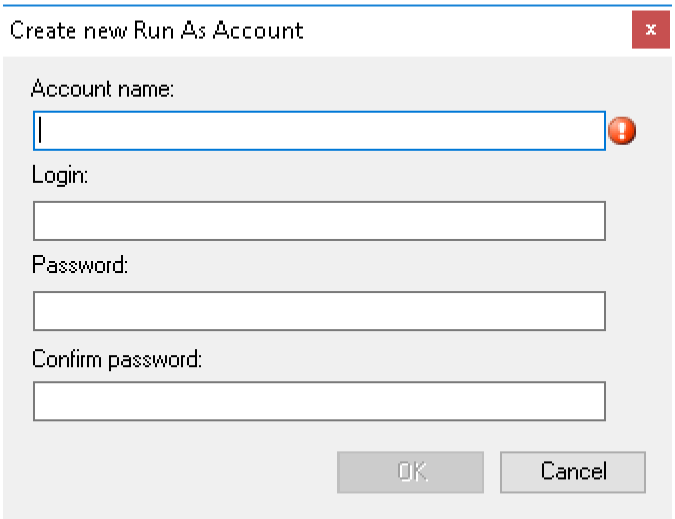
Select common Run As Account with Windows AD credentials

Note that selected Run As Account will be used for all connections.

Enter data sources and connection strings

- Each data source/connection string is to be entered in a new line.
- You can add keywords to skip connection test for some lines at the end of the data source or connection string: <data source or connection string>; MachineName="< ... >"; InstanceName="< ... >"; Platform="< ... >" Supported platforms: Windows, Linux

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.


Create new Run As Account

Account name:

Login:

Password:

Confirm password:

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.

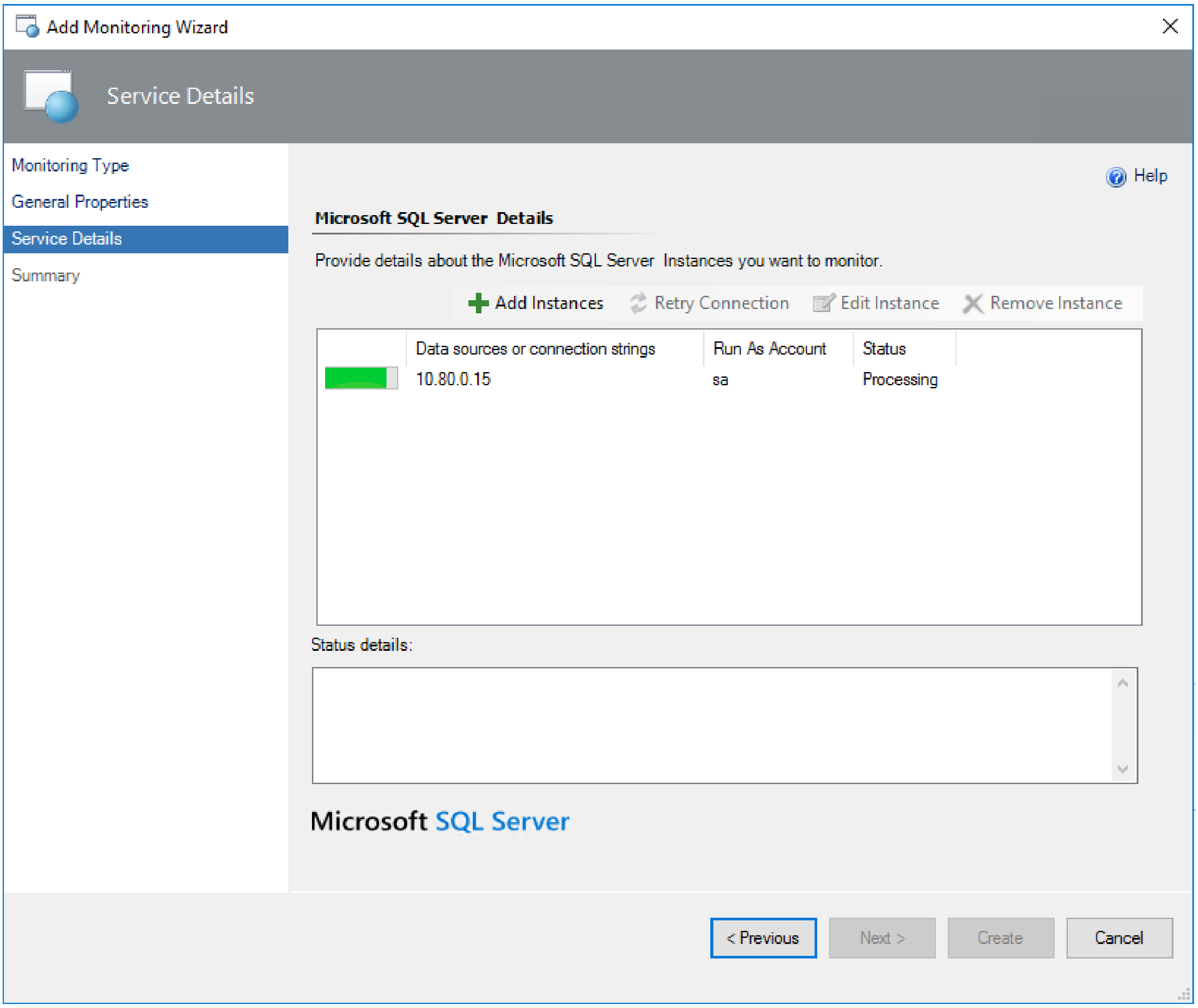
The data is to be entered in the format provided in the examples below:

- 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"

> ⚠ When adding a Linux-based instance, the connection test fails if an 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.

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



Microsoft SQL Server Details

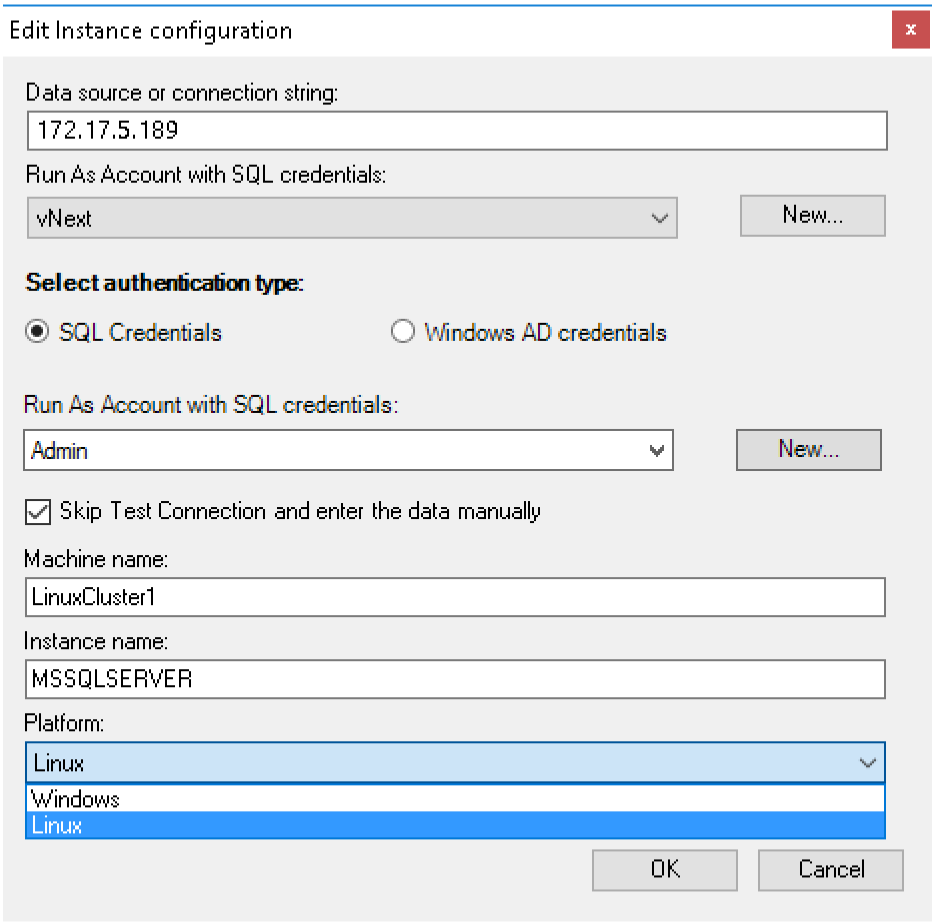
Provide details about the Microsoft SQL Server Instances you want to monitor.

	Data sources or connection strings	Run As Account	Status
🟩	10.80.0.15	sa	Processing

Status details:

> ⚠ The Monitoring Template Wizard may show the following error while checking connection: "An error occurred discovery: A connection was successfully established with the server, but then an error occurred during the login process". See Login fails when adding a new instance using the Add Monitoring Wizard to work this issue out.

When the connection testing is completed, you can view and edit the 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: 172.17.5.189

Run As Account with SQL credentials: vNext

Select authentication type:

● SQL Credentials ○ Windows AD credentials

Run As Account with SQL credentials: Admin

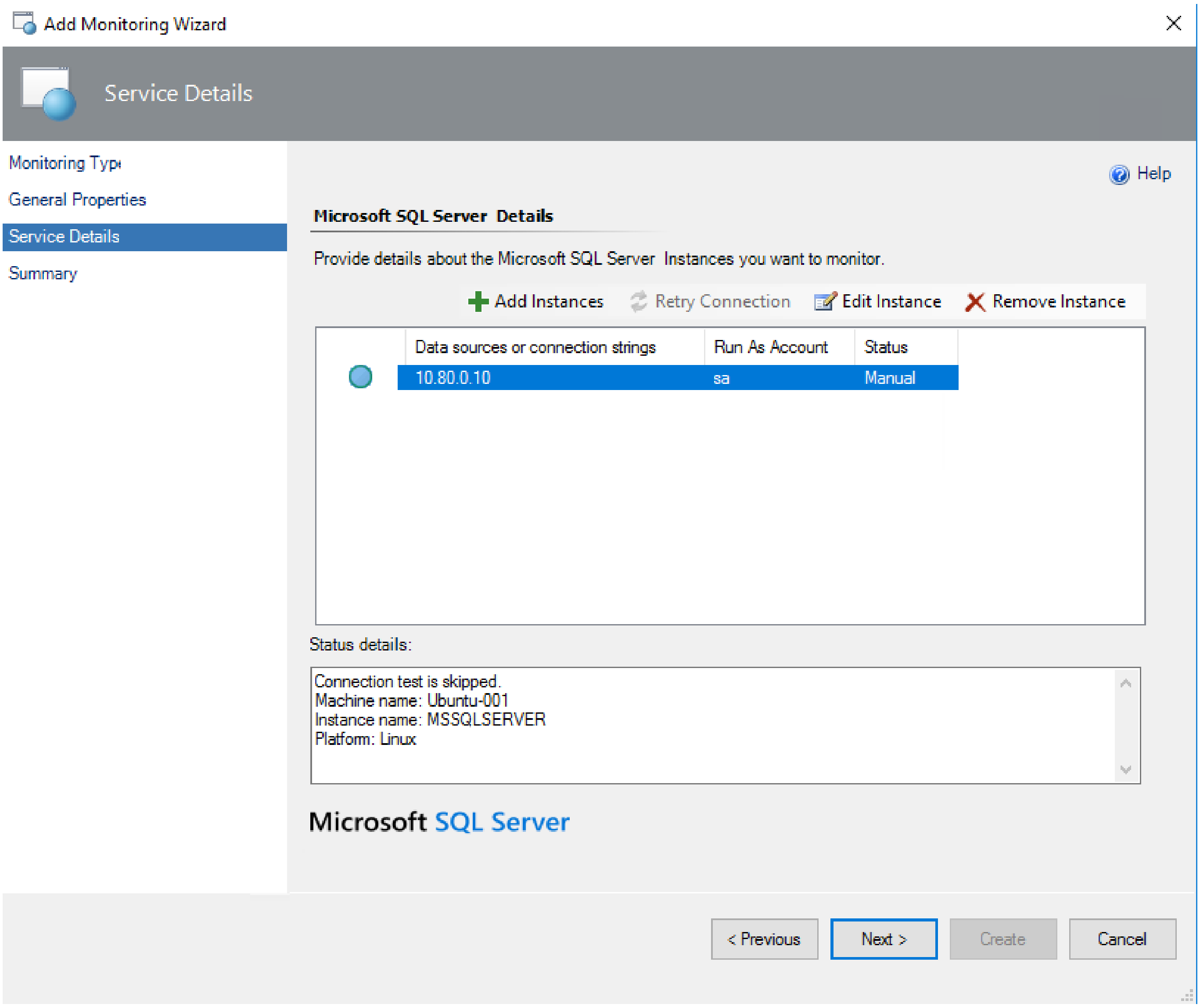
☑ Skip Test Connection and enter the data manually

Machine name: LinuxCluster1

Instance name: MSSQLSERVER

Platform: Linux

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”:



Microsoft SQL Server Details

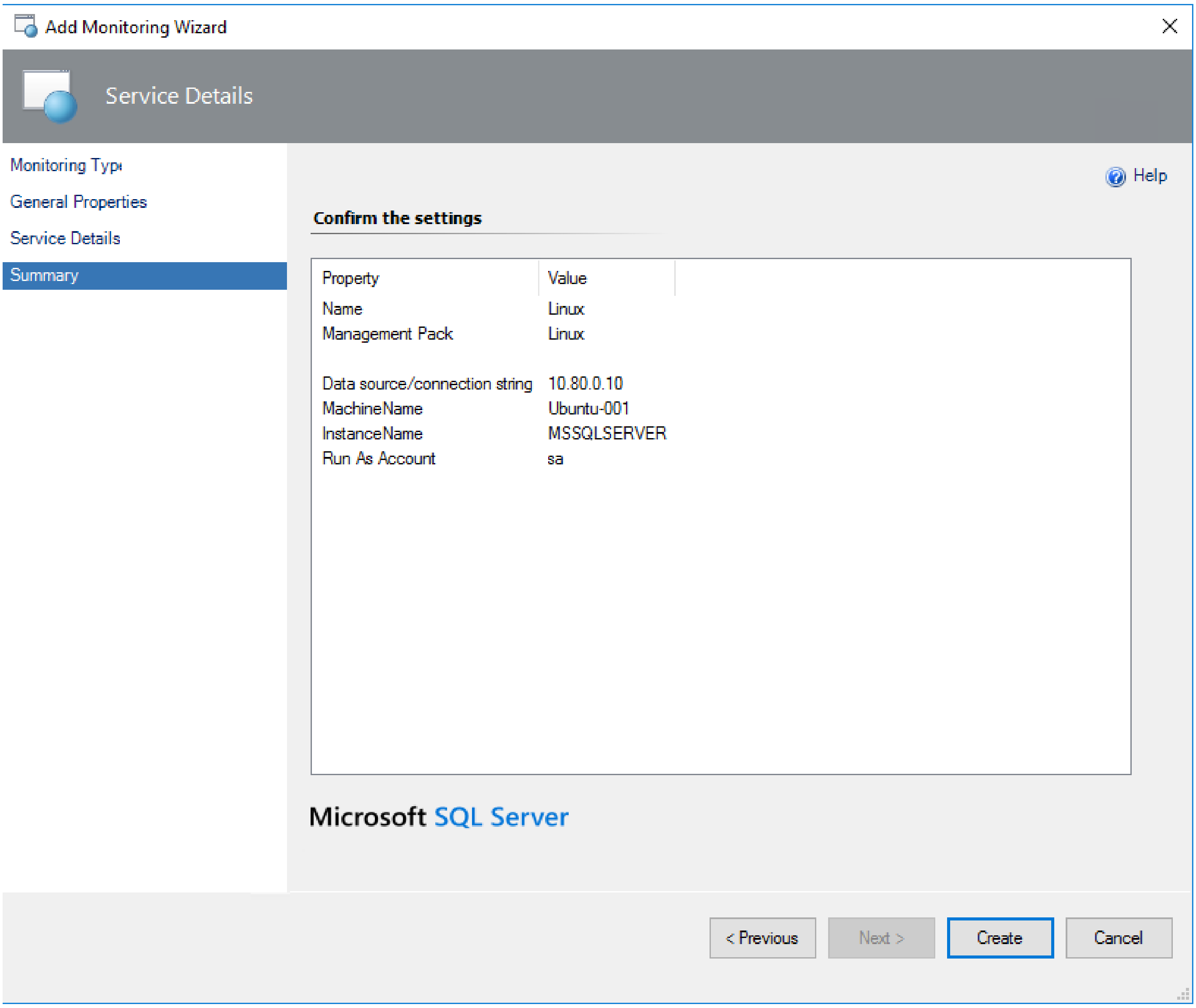
Provide details about the Microsoft SQL Server Instances you want to monitor.

Data sources or connection strings	Run As Account	Status
10.80.0.10	sa	Manual

Status details:

```
Connection test is skipped.
Machine name: Ubuntu-001
Instance name: MSSQLSERVER
Platform: Linux
```

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



Confirm the settings

Property	Value
Name	Linux
Management Pack	Linux
Data source/connection string	10.80.0.10
MachineName	Ubuntu-001
InstanceName	MSSQLSERVER
Run As Account	sa

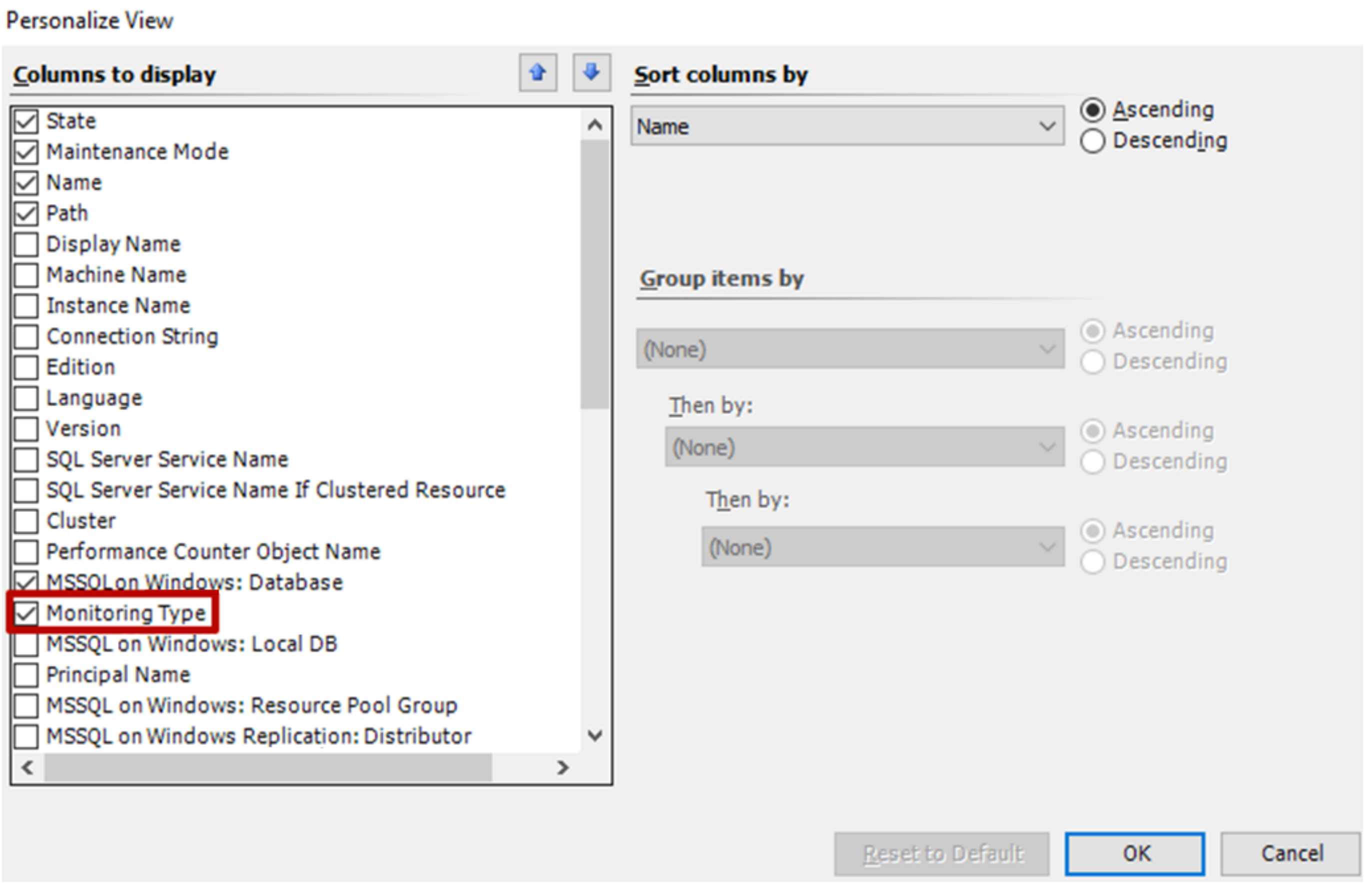
After that, your monitoring template will be successfully created.

Configuring Mixed Monitoring Mode

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 the override.

When you enable Mixed Monitoring, only SQL Server Seed is discovered locally by the SCOM agent. All other workflows are run from the dedicated management server pool. The details regarding the SQL Server Monitoring Pool discovery configuration are available in Configuring SQL Server Monitoring Pool section.

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:



Personalize View

Columns to display

- ☑ State
- ☑ Maintenance Mode
- ☑ Name
- ☑ Path
- □ Display Name
- □ Machine Name
- □ Instance Name
- □ Connection String
- □ Edition
- □ Language
- □ Version
- □ SQL Server Service Name
- □ SQL Server Service Name If Clustered Resource
- □ Cluster
- □ Performance Counter Object Name
- ☑ MSSQL on Windows: Database
- ☑ Monitoring Type
- □ MSSQL on Windows: Local DB
- □ Principal Name
- □ MSSQL on Windows: Resource Pool Group
- □ MSSQL on Windows Replication: Distributor

Sort columns by

Name ● Ascending ○ Descending

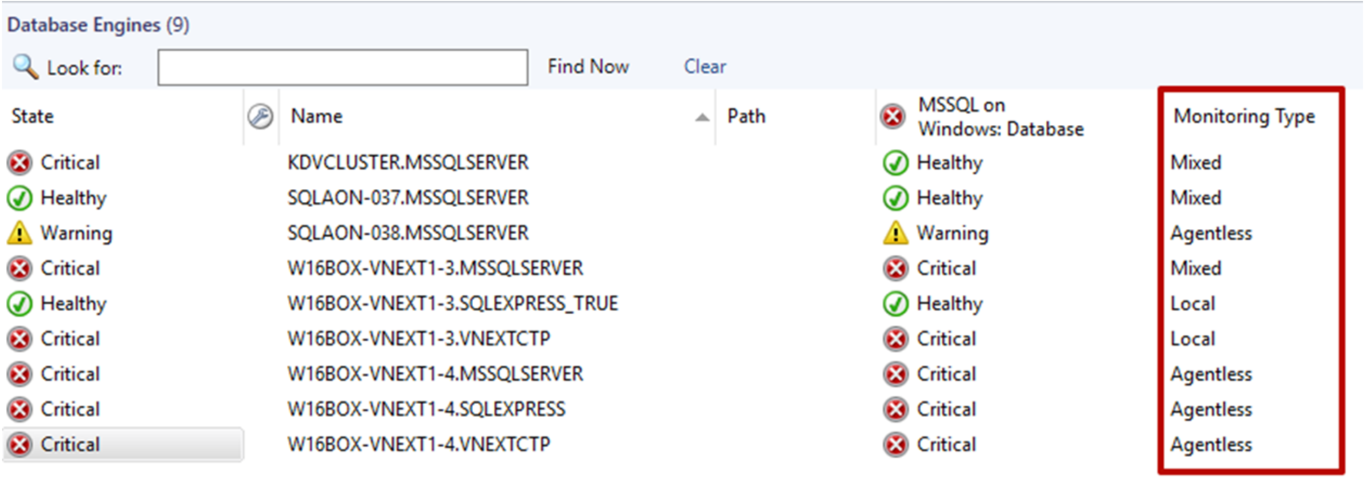
Group items by

(None) ● Ascending ○ Descending

Then by: (None) ● Ascending ○ Descending

Then by: (None) ● Ascending ○ Descending

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

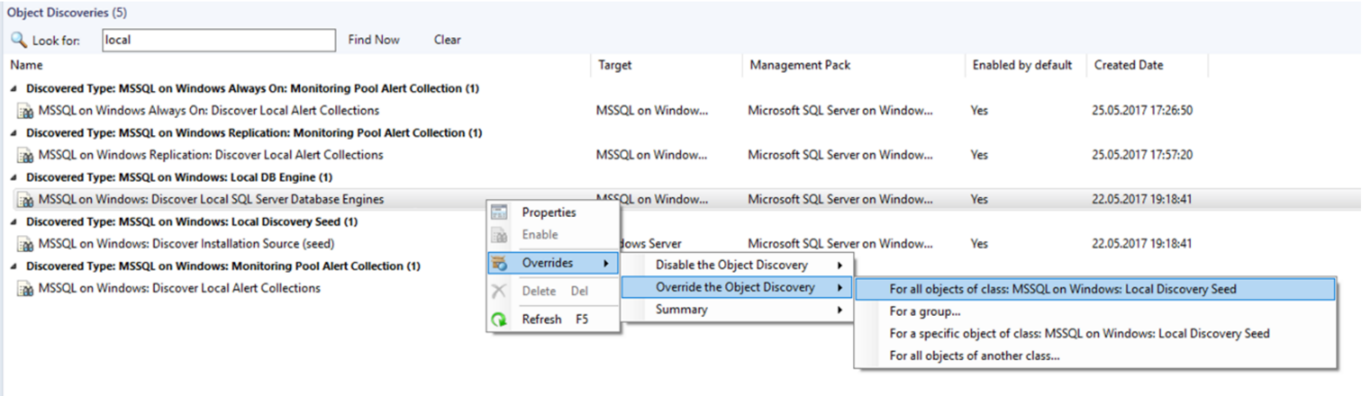


Database Engines (9)

Look for: 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

To enable Mixed Monitoring Mode, in the Operations Manager console, 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**.



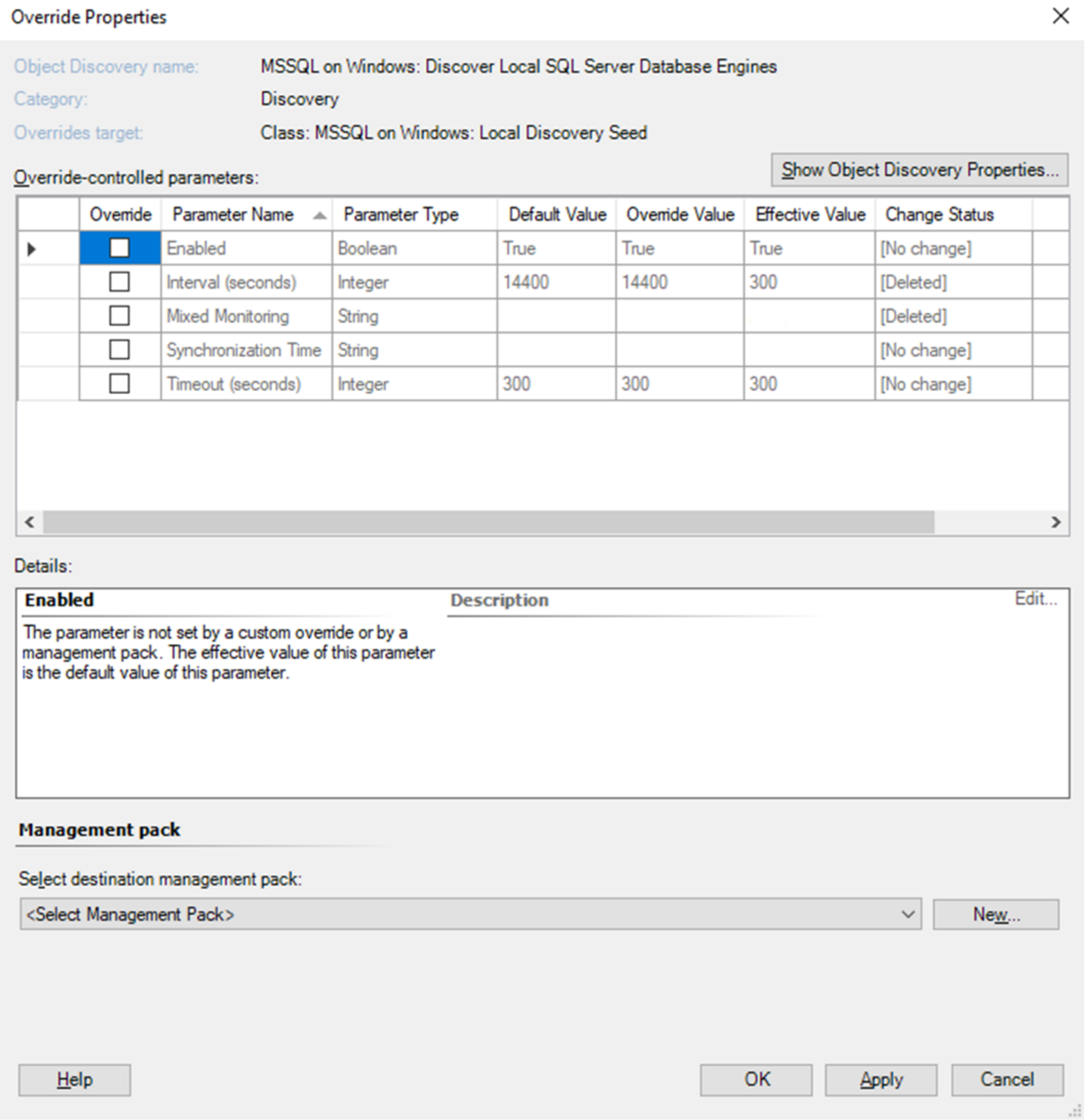
Object Discoveries (5)

Look for: local

Name	Target	Management Pack	Enabled by default	Created Date
Discovered Type: MSSQL on Windows Always On: Monitoring Pool Alert Collection (1)				
MSSQL on Windows Always On: Discover Local Alert Collections	MSSQL on Window...	Microsoft SQL Server on Window...	Yes	25.05.2017 17:26:50
Discovered Type: MSSQL on Windows Replication: Monitoring Pool Alert Collection (1)				
MSSQL on Windows Replication: Discover Local Alert Collections	MSSQL on Window...	Microsoft SQL Server on Window...	Yes	25.05.2017 17:57:20
Discovered Type: MSSQL on Windows: Local DB Engine (1)				
MSSQL on Windows: Discover Local SQL Server Database Engines	MSSQL on Window...	Microsoft SQL Server on Window...	Yes	22.05.2017 19:18:41
Discovered Type: MSSQL on Windows: Local Discovery Seed (1)				
MSSQL on Windows: Discover Installation Source (seed)	lows Server	Microsoft SQL Server on Window...	Yes	22.05.2017 19:18:41
Discovered Type: MSSQL on Windows: Monitoring Pool Alert Collection (1)				
MSSQL on Windows: Discover Local Alert Collections				

Find Now
Clear
Properties
Enable
Overrides
Delete Del
Refresh F5
Disable the Object Discovery
Override the Object Discovery
Summary
For all objects of class: MSSQL on Windows: Local Discovery Seed
For a group...
For a specific object of class: MSSQL on Windows: Local Discovery Seed
For all objects of another class...

As a result, the **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 an 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:

Override	Parameter Name	Parameter Type	Default Value	Override Value	Effective Value	Change Status
□	Enabled	Boolean	True	True	True	[No change]
□	Interval (seconds)	Integer	14400	14400	300	[Deleted]
□	Mixed Monitoring	String				[Deleted]
□	Synchronization Time	String				[No change]
□	Timeout (seconds)	Integer	300	300	300	[No change]

Details:

Enabled

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>

Agentless and Mixed Modes Performance

When the monitoring is configured via Agentless or Mixed Monitoring mode, the management pack's workflows that run on management servers are mapped to either the SQL Server Monitoring Pool or All Management Servers Pool (if SQL Server Monitoring Pool does not have members). In this case, a management server or servers experience a higher load than in the Local Agent Monitoring mode.

The following monitoring configuration was validated for both Agentless and Mixed Monitoring modes:

- SCOM Server 1 – Azure size: Standard DS12_v2, 4 vcpus, 28 GB memory, 12800 IOPS, Windows Server 2012R2, SCOM 2012R2
- SCOM Server 2 – a server dedicated to monitoring SQL Server. The only member of the SQL Server Monitoring Pool. Azure size: Standard DS12_v2, 4 vcpus, 28 GB memory, 12800 IOPS, Windows Server

2012R2, SCOM 2012R2.

- 12 VMs with SQL Server (2012, 2014, 2016, 2017) – 600 databases per instance, ~40000 SQL Server MP objects in total.

SCOM Server 2 that monitored SQL Server had more than half of its CPU and RAM resources available during the performance testing session.

SQL Server Agent Alerting Rules: Specifics of Configuration

Management Pack includes nine alerting rules for SQL Server Agent-related errors. These rules are enabled by default in Agent Monitoring Mode but disabled in Mixed Monitoring Mode. In Agentless Monitoring Mode, these rules cannot work (therefore they are not available for SQL on Linux at all).

- 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

They are disabled in Mixed Monitoring Mode because, by default, Operations Manager does not allow to collect events from the event log on remote computers. But overriding these rules by enabling the "AllowProxying" option makes it possible.

> ⚠ Note that 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 enable the "AllowProxying" option.

Always On Alert Rules

This management pack has two event rules for alerting when the following events appear in the Windows Application log:

- Event ID 1480, Database Replica role is changed
- Event ID 19406, Availability Replica role changed

SQL Server may not fire these events in the Application Log by default. To enable them, execute the following T-SQL scripts:

- sp_altermessage 1480, 'with_log', 'true'
- sp_altermessage 19406, 'with_log', 'true'

Always On Policies Monitoring

This management pack collects the health for all available Always On objects on the target instance of SQL Server by reading the state of the PBM (Policy-Based Management) policies state for each of the objects.

Beside system policies, this management pack provides the ability to monitor Custom User Policies defined for these facets:

- Availability Group
- Availability Replica
- Database Replica

For each facet, the management pack introduces two monitors for Custom User Policy:

- Two-state monitor with the 'Warning' state. This monitor reflects the state of Custom User Policy, which has one of the predefined warning categories as Policy Category.
- Two-state monitor with the 'Error' state. This monitor reflects the state of Custom User Policy, which has one of the predefined error categories as Policy Category.

Data File and Transaction Log File Space Monitoring

This 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 the 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 at the Filegroup and File levels.

Many Databases on the Same Drive

Space monitoring introduced in 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 the 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 and performance rules are disabled by default. Enable them only for specific DBs when necessary.

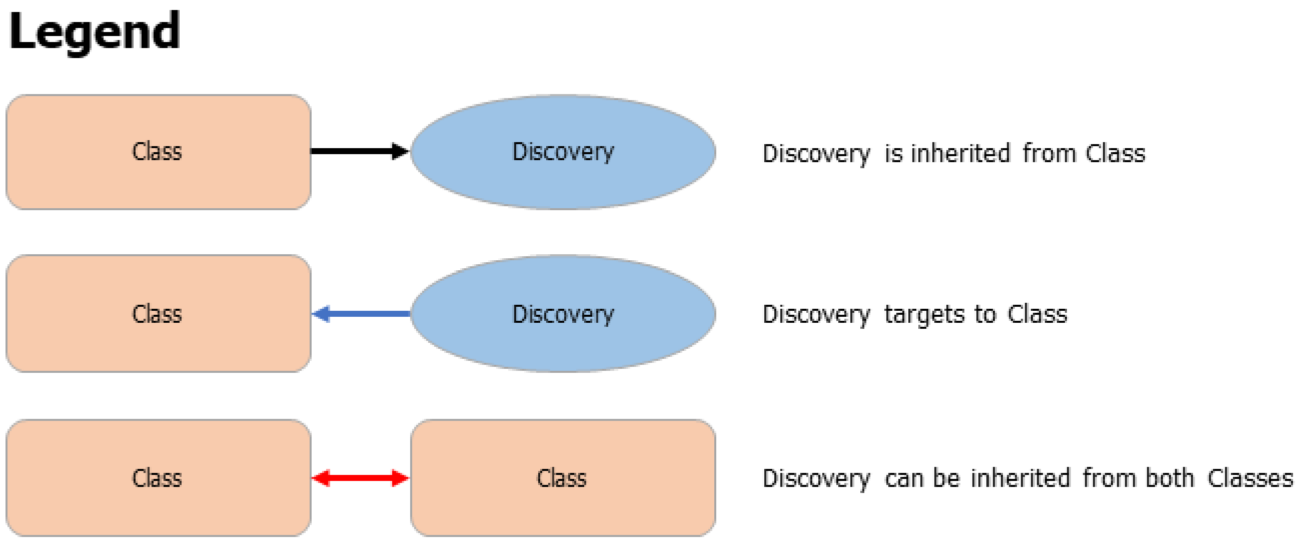
Blocked Sessions

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

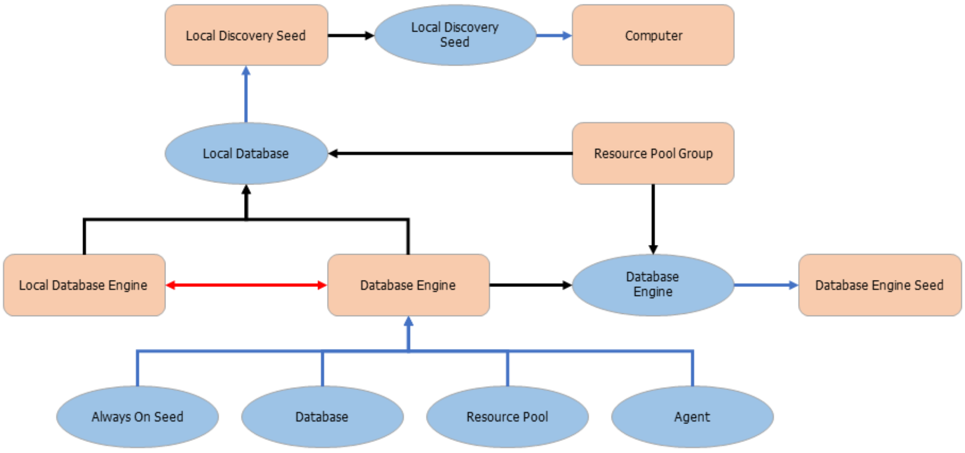
You can apply an override to change the **WaitMinutes** parameter, which is used to determine if the blocked session should be considered as long-running or not. The default value for this parameter is one minute.

How Health Rolls Up

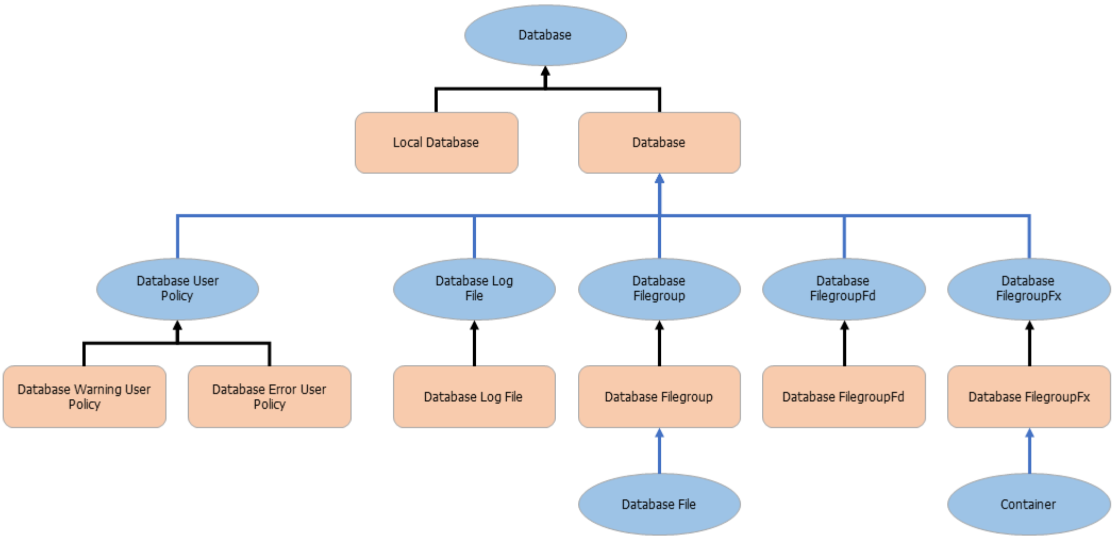
The following diagrams show how the health states of objects roll up for the SQL Server on Windows management pack.


Legend
Class
Discovery
Discovery is inherited from Class
Class
Discovery
Discovery targets to Class
Class
Class
Discovery can be inherited from both Classes

Health Rollup Diagram

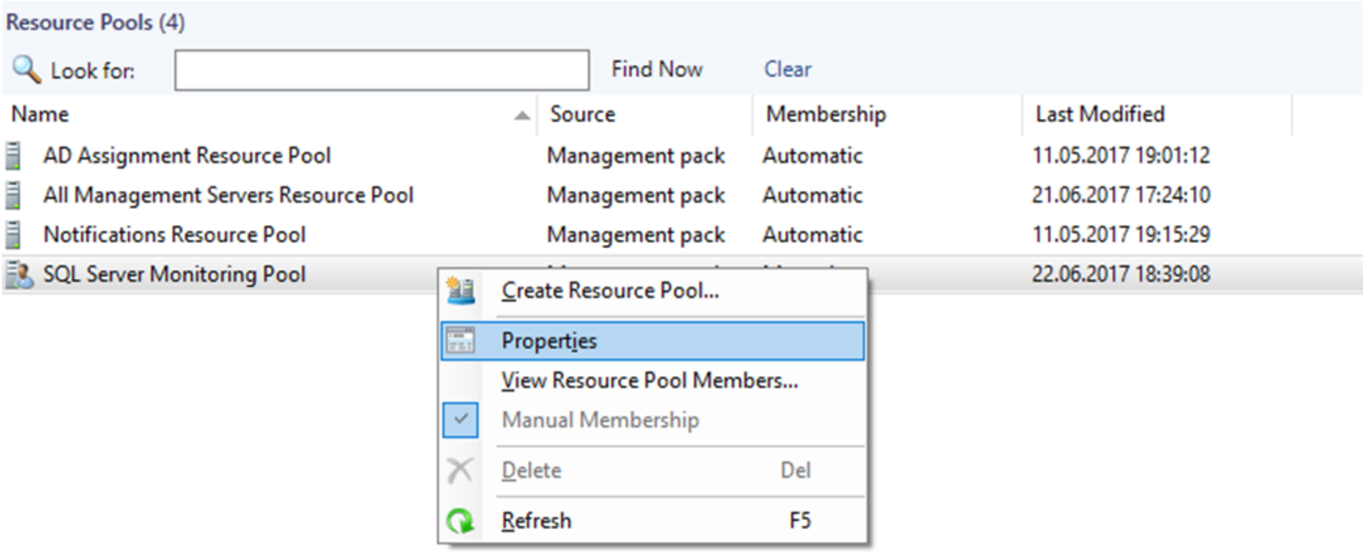


Database Health Rollup Diagram



Configuring SQL Server Monitoring Pool

The monitoring pool is available for configuration in the Operations Manager console. 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.

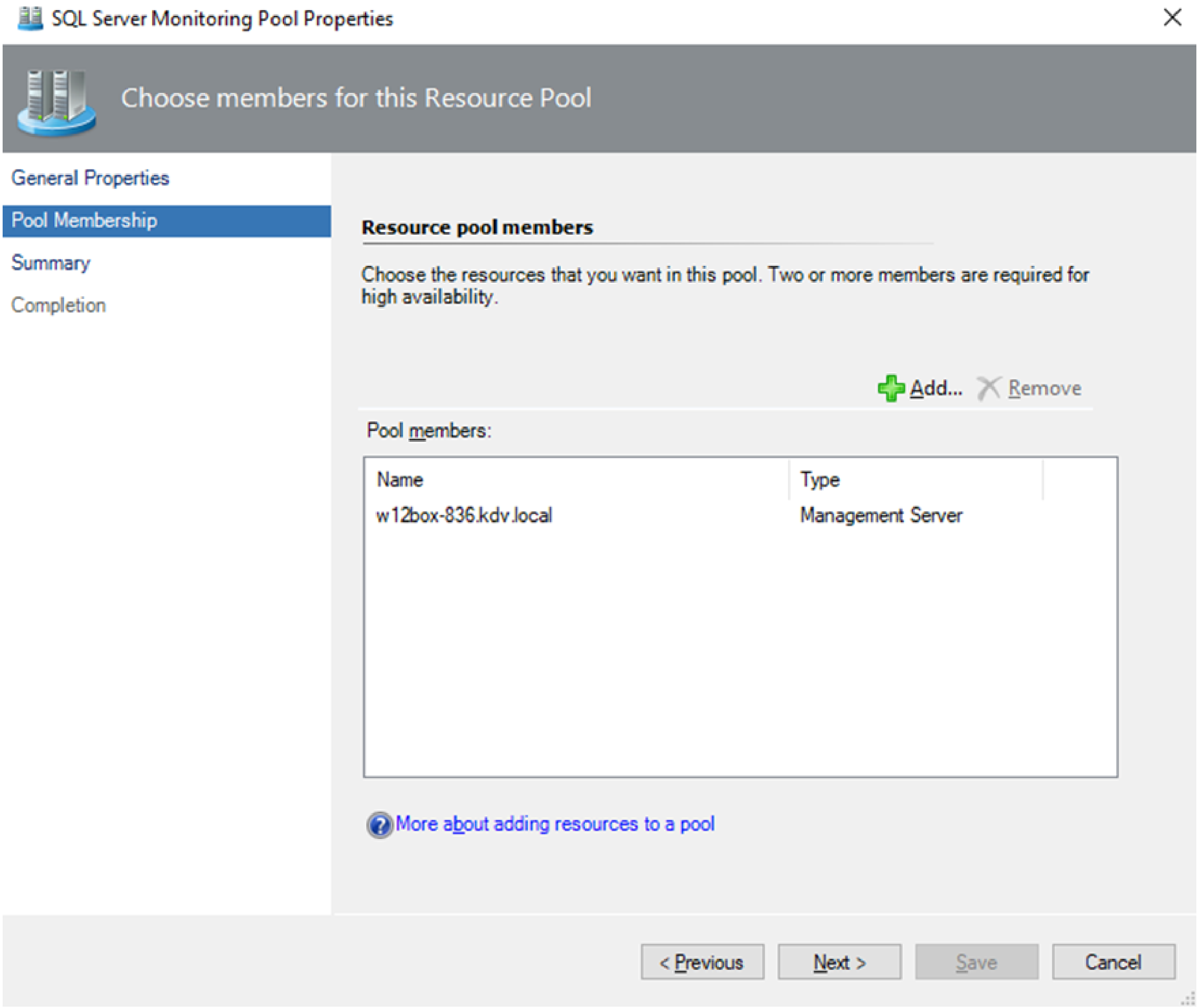


Resource Pools (4)

Look for:

Name	Source	Membership	Last Modified
AD Assignment Resource Pool	Management pack	Automatic	11.05.2017 19:01:12
All Management Servers Resource Pool	Management pack	Automatic	21.06.2017 17:24:10
Notifications Resource Pool	Management pack	Automatic	11.05.2017 19:15:29
SQL Server Monitoring Pool			22.06.2017 18:39:08

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



SQL Server Monitoring Pool Properties
Choose members for this Resource Pool
General Properties
Pool Membership
Summary
Completion
Add...
Remove
More about adding resources to a pool
< Previous
Next >
Save
Cancel

Resource pool members

Choose the resources that you want in this pool. Two or more members are required for high availability.

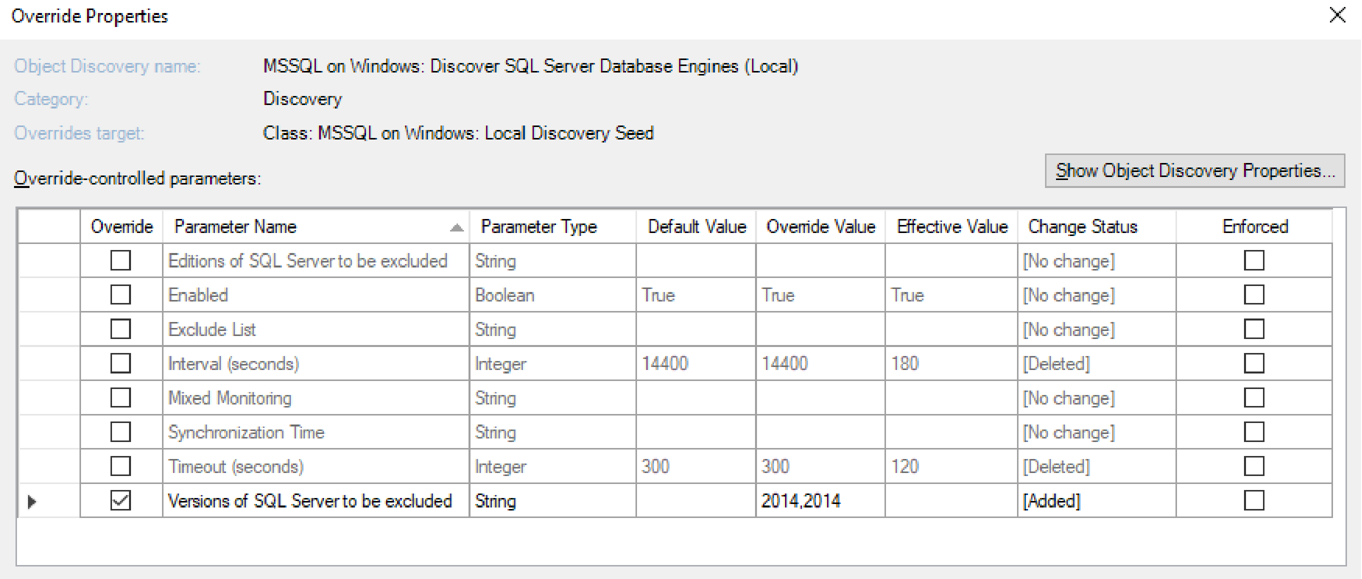
Pool members:

Name	Type
w12box-836.kdv.local	Management Server

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

Disabling Monitoring of Specified SQL Server Versions

You can exclude instances of SQL Server from the monitoring by SQL Server version. Create an override for the “Versions of SQL Server to be excluded” parameter of the “MSSQL on Windows: Discover SQL Server Database Engines (Local)” discovery and list versions separating them with commas. For example, the override “2014,2012” makes the management pack remove all previously discovered instances of SQL Server 2012 and 2014 and disable further discovery of such instances.



Override Properties

Object Discovery name: MSSQL on Windows: Discover SQL Server Database Engines (Local)

Category: Discovery

Overrides target: Class: MSSQL on Windows: Local Discovery Seed

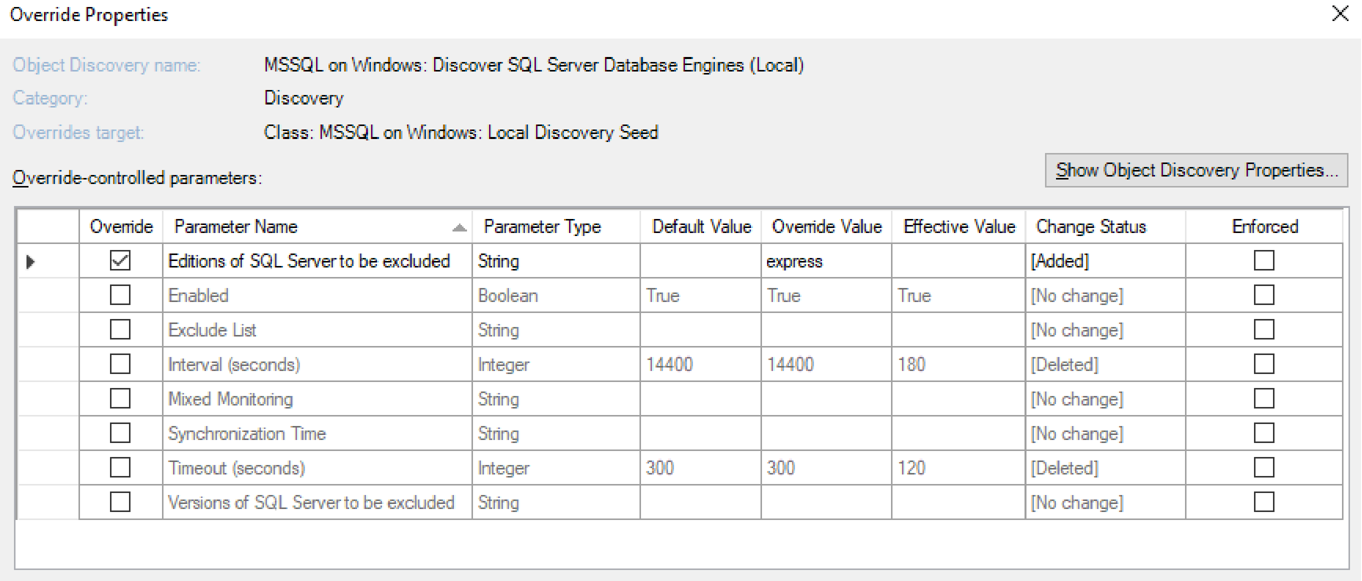
Override-controlled parameters:

	Override	Parameter Name	Parameter Type	Default Value	Override Value	Effective Value	Change Status	Enforced
	□	Editions of SQL Server to be excluded	String				[No change]	□
	□	Enabled	Boolean	True	True	True	[No change]	□
	□	Exclude List	String				[No change]	□
	□	Interval (seconds)	Integer	14400	14400	180	[Deleted]	□
	□	Mixed Monitoring	String				[No change]	□
	□	Synchronization Time	String				[No change]	□
	□	Timeout (seconds)	Integer	300	300	120	[Deleted]	□
▶	☑	Versions of SQL Server to be excluded	String		2014,2014		[Added]	□

Disabling Monitoring of Specified SQL Server Editions

You can exclude instances of SQL Server from the monitoring by SQL Server edition. Create an override for the “Editions of SQL Server to be excluded” parameter of the “MSSQL on Windows: Discover SQL Server Database Engines (Local)” discovery and list editions separating them with commas. Use the matching table below to figure out what short names of the editions to use in the parameter.

Short Name	Covered Editions
Enterprise	Enterprise Edition, Enterprise Edition: Core-based Licensing, Enterprise Evaluation Edition
Standard	Standard Edition, Business Intelligence Edition
Web	Web Edition
Developer	Developer Edition
Express	Express Edition, Express Edition with Advanced Services



Override Properties

Object Discovery name: MSSQL on Windows: Discover SQL Server Database Engines (Local)

Category: Discovery

Overrides target: Class: MSSQL on Windows: Local Discovery Seed

Override-controlled parameters:

	Override	Parameter Name	Parameter Type	Default Value	Override Value	Effective Value	Change Status	Enforced
▶	☑	Editions of SQL Server to be excluded	String		express		[Added]	□
	□	Enabled	Boolean	True	True	True	[No change]	□
	□	Exclude List	String				[No change]	□
	□	Interval (seconds)	Integer	14400	14400	180	[Deleted]	□
	□	Mixed Monitoring	String				[No change]	□
	□	Synchronization Time	String				[No change]	□
	□	Timeout (seconds)	Integer	300	300	120	[Deleted]	□
	□	Versions of SQL Server to be excluded	String				[No change]	□

Disabling Monitoring of Specified Databases by Name

You can stop discovery and monitoring of databases by specifying their names in the "Exclude list" property of both discoveries "MSSQL on Windows: Discover SQL Server Databases for a Database Engine" and "MSSQL on Linux: Discover SQL Server Databases for a Database Engine." Use commas to separate database names on the list and asterisks to replace one or more characters. For example, setting the parameter to `dev*,*test*,*stage,dbnotmon` causes the monitoring configuration as in the table below.

DB Name	Monitored/Not monitored
dev	Not monitored
dev_sales	Not monitored
sales_dev	Monitored
test	Not monitored
test_sales	Not monitored
sales_test	Not monitored
stage	Not monitored
stage_dev	Monitored
dev_stage	Not monitored
dbnotmon	Not monitored
dbnotmon_sales	Monitored
sales_dbnotmon	Monitored

If you have `*` on the list as a database name (e.g., `*temp*,*,*dev*` or `*temp,*`), it disables monitoring of any database.

Monitor "Securables Configuration Status"

This monitor checks if each of the required SQL Server securables is accessible under the configured run-as account.

The following is a complete list of securables that are checked by the monitor targeted to the SQL Server DB Engine:

- Server-Level permissions
  - VIEW SERVER STATE
  - VIEW ANY DEFINITION
  - VIEW ANY DATABASE
  - ALTER ANY DATABASE
- SELECT permission on dynamic management views
  - master.sys.dm_hadr_availability_group_states
  - master.sys.dm_hadr_availability_replica_states
  - master.sys.dm_hadr_database_replica_cluster_states

    - master.sys.dm_hadr_database_replica_states
    - sys.dm_os_performance_counters
    - sys.dm_tran_active_transactions
    - sys.dm_tran_session_transactions
    - sys.dm_tran_active_transactions
    - sys.dm_tran_session_transactions
    - sys.dm_exec_sessions
    - sys.dm_exec_requests
    - sys.dm_exec_connections
    - sys.dm_os_sys_info
    - sys.dm_os_ring_buffers
    - sys.dm_os_volume_stats
    - sys.dm_hadr_database_replica_states
    - sys.dm_server_services
    - sys.dm_db_xtp_checkpoint_files
    - sys.dm_db_xtp_table_memory_stats
    - sys.dm_resource_governor_resource_pools
    - sys.dm_db_xtp_hash_index_stats
    - sys.dm_os_threads

- SELECT permission on catalog views

    - msdb.dbo.syspolicy_object_sets
    - msdb.dbo.syspolicy_policy_categories
    - msdb.dbo.syspolicy_target_sets
    - msdb.dbo.syspolicy_target_set_levels
    - sys.dm_os_host_info
    - msdb.dbo.syspolicy_policies
    - msdb.dbo.syspolicy_conditions
    - msdb.dbo.syspolicy_policy_execution_history
    - msdb.dbo.syspolicy_configuration
    - msdb.dbo.syspolicy_system_health_state
    - sys.database_files
    - sys.availability_groups
    - sys.availability_replicas
    - sys.databases
    - sys.database_files
    - sys.tables
    - sys.filegroups
    - sys.syscolumns
    - sys.sysprocesses
    - sys.availability_replicas
    - sys.database_mirroring
    - sys.configurations
    - msdb.dbo.syspolicy_policies
    - msdb.dbo.syspolicy_conditions
    - msdb.dbo.syspolicy_policy_execution_history

    - msdb.dbo.syspolicy_policy_execution_history_details
    - msdb.dbo.sysjobschedules
    - msdb.dbo.log_shipping_primary_databases
    - msdb.dbo.log_shipping_secondary_databases
    - msdb.dbo.backupset

- EXECUTE permission on stored procedures

    - sys.sp_enumerrorlogs
    - sys.xp_readerrorlog
    - msdb.dbo.sp_help_jobactivity
    - sys.xp_instance_regread
    - msdb.dbo.sp_help_job

The following is a complete list of securables that are checked by the monitor targeted to SQL Server databases:

- SELECT permission on catalog views
    - sys.database_files
    - sys.tables
    - sys.filegroups
    - sys.syscolumns

Security Configuration of Management Pack

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

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 SCOM SDK Run As Profile – this profile is for SQL Server MP workflows that need access to SCOM SDK.
- Microsoft SQL Server SQL Credentials Run As Profile – this profile is for the **Agentless Monitoring Mode** only.

> ⚠ Do not bind any account to profile *Microsoft SQL Server SQL Credentials Run As Profile* if you monitor SQL Server in Local or Mixed Monitoring modes. Only a basic action account can be bound to the profile, do not use a Windows account or non-basic account with this profile.

When Local Monitoring or Mixed Monitoring mode is used, all discoveries, monitors, and tasks defined in the SQL Server MP 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.

Enabling “Allow Log On Locally” Security Policy

If a domain account is used as the action account for this management pack, make sure to enable the “Allow log on locally” policy for it. It is a requirement for SQL Server on Windows and on Linux. For more information about configuring this security policy setting, see this Docs article Allow Log On Locally.

Configuring Run As Profiles for Local and Mixed Monitoring Modes

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

SCOM Action Account is Local Administrator and SA

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 section of Low-Privilege Agent Monitoring.
- In case when servers hosting Always On Availability Replicas (at least one of them) have the machine name longer 15 characters, make sure to take steps described in How to Configure Permissions for Always On Workflows when Servers Have Machine Names Longer than 15 Characters.

SCOM Action Account is Local Administrator and Not Have SA Rights

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:

- Create a new Domain User account and add this account to the Local Administrators group on each monitored server.
- Grant SA rights to this account in SQL Server.
- Create a new Action account in SCOM and map it to the Domain User account created above.
- Map the new Action account to all SQL Server MP Run As Profiles.
- If you store SQL Server databases on an SMB file share, make sure that Default Action Account has the rights described in the corresponding section of Low-Privilege Agent Monitoring.

SCOM Action Account is Local System and Not Have SA Rights

> ⚠ This scenario is for Local Monitoring Mode only.

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.

Follow the next steps to configure your security configuration with SID:

- Configure using a service SID for HealthService as it is described in How to Configure Monitoring by Means of a Service Security Identifier.
- If you have SQL Server Cluster instances, make sure to take the steps described in How to Configure HealthService Service SID for Monitoring SQL Server Cluster Instances.

Low-Privilege Monitoring

In case you need to grant the minimally required rights to SQL MP workflows, follow the instructions provided in section Configuring Low-Privilege Monitoring.

Configuring Run As Profiles in Agentless Monitoring Mode

To configure Run As Profiles in agentless monitoring mode, create a login in SQL Server for monitoring purposes and grant it SA rights or a set of Low Privilege permissions. You can use SQL Server authentication or Windows authentication, then use this login in the **Add Monitoring Wizard** while adding a SQL Server instance.

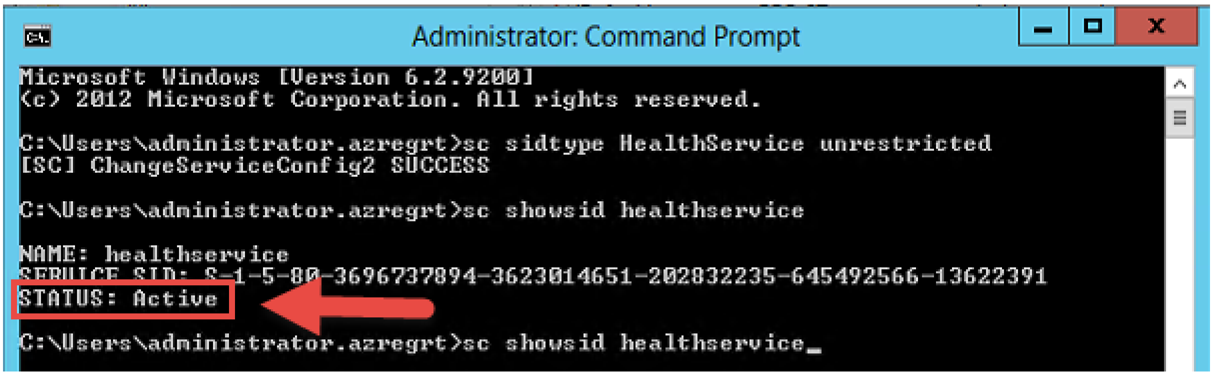
For more information on how to add a SQL Server instance to monitor it agentlessly, see the Configuring Agentless Monitoring Mode section. For more information on how to configure Low Privilege monitoring in Agentless mode, see the Low-Privilege Agentless Monitoring section.

Configuring Monitoring by Means of Service Security Identifier

Below are the steps to configure monitoring via Service SIDs for SQL Server on a Windows Server instance - was first published by Kevin Holman in his blog. The original article is available here. The SQL scripts to configure the lowest-privilege access were developed by Brandon Adams.

1. Open Command Prompt as Administrator and run `sc sidtype HealthService unrestricted`, then restart "Health Service".

2. Run `sc showsid HealthService` and make sure "STATUS" is active.



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

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

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

NAME: healthservice
SERVICE SID: S-1-5-80-3696737894-3623014651-202832235-645492566-13622391
STATUS: Active

C:\Users\administrator.azregrt>sc showsid healthservice_
```

3. Open **Registry Editor** and check that *ServiceSidType* key equals to 1 at *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HealthService*.

4. Create the login "NT SERVICE\HealthService" for the HealthService SID on every SQL Server Instance and grant it with SA rights. If you cannot grant it with the SA rights, then skip this step and take step 5.

5. Take this step only if you cannot take step "4". Use the following SQL scripts to set up the lowest privilege configuration for the account:

```
USE [master]
SET NOCOUNT ON
/*User account which SCOM will use for access
    Default is the Service SID for the HealthService*/
DECLARE @accountname sysname = 'NT SERVICE\HealthService'
-- Create the server role and grant permissions
CREATE SERVER ROLE [SCOM_HealthService]
GRANT VIEW ANY DATABASE TO [SCOM_HealthService];
GRANT ALTER ANY DATABASE TO [SCOM_HealthService];
GRANT VIEW ANY DEFINITION TO [SCOM_HealthService];
GRANT VIEW SERVER STATE TO [SCOM_HealthService]
DECLARE @createLoginCommand nvarchar(200)
SET @createLoginCommand = '
  CREATE LOGIN '+ QUOTENAME(@accountname) +
  ' FROM WINDOWS WITH DEFAULT_DATABASE=[master];'
EXEC(@createLoginCommand);
-- Add the login to the user-defined server role
EXEC sp_addsrvrolemember @loginame = @accountname
  , @rolename = 'SCOM_HealthService'
DECLARE @createDatabaseUserAndRole nvarchar(max)
SET @createDatabaseUserAndRole = '';
SELECT @createDatabaseUserAndRole = @createDatabaseUserAndRole + '
  USE ' + QUOTENAME(db.name) + ';
  CREATE USER ' + QUOTENAME(@accountname) +
  ' FOR LOGIN ' + QUOTENAME(@accountname) + ';
  CREATE ROLE [SCOM_HealthService];
  EXEC sp_addrolemember @rolename =
  ''SCOM_HealthService'', @membername
  = '+ QUOTENAME(@accountname) + ''
-- 'ALTER ROLE [SCOM_HealthService] ADD MEMBER '
  -- '+ QUOTENAME(@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);
EXEC(@createDatabaseUserAndRole)
GO
USE [master];
GRANT EXECUTE ON sys.xp_readerrorlog
  TO [SCOM_HealthService]
USE [msdb];
GRANT SELECT on [dbo].[sysjobschedules]
  TO [SCOM_HealthService];
GRANT SELECT on [dbo].[sysschedules]
  TO [SCOM_HealthService];
GRANT SELECT on [dbo].[sysjobs_view]
  TO [SCOM_HealthService];
GRANT SELECT on [dbo].[log_shipping_primary_databases]
```

```
  TO [SCOM_HealthService];
GRANT SELECT on [dbo].[log_shipping_secondary_databases]
  TO [SCOM_HealthService];
GRANT SELECT on [dbo].[log_shipping_monitor_history_detail]
  TO [SCOM_HealthService];
GRANT SELECT on [dbo].[log_shipping_monitor_secondary]
  TO [SCOM_HealthService];
GRANT SELECT on [dbo].[log_shipping_monitor_primary]
  TO [SCOM_HealthService];
GRANT EXECUTE on [dbo].[sp_help_job]
  TO [SCOM_HealthService];
GRANT EXECUTE on [dbo].[sp_help_jobactivity]
  TO [SCOM_HealthService];
EXEC sp_addrolemember @rolename='PolicyAdministratorRole'
  , @membername='SCOM_HealthService';
EXEC sp_addrolemember @rolename='SQLAgentReaderRole'
  , @membername='SCOM_HealthService';
```

6. In order to run SQL Server MP tasks, such as “Set database Offline”, “Set database Online”, and “Set database to Emergency state,” grant HealthService SID account with permission ALTER ANY DATABASE.

   ```
   USE [master]
   GRANT ALTER ANY DATABASE TO [SCOM_HealthService];
   ```

7. The login “NT AUTHORITY\SYSTEM” needs to be present as a SQL login, and must not be set to “Disabled” status, also “NT AUTHORITY\SYSTEM” login must be present and enabled for Cluster Nodes and Always On.

Configuring HealthService Service SID for SQL Server Cluster Instances

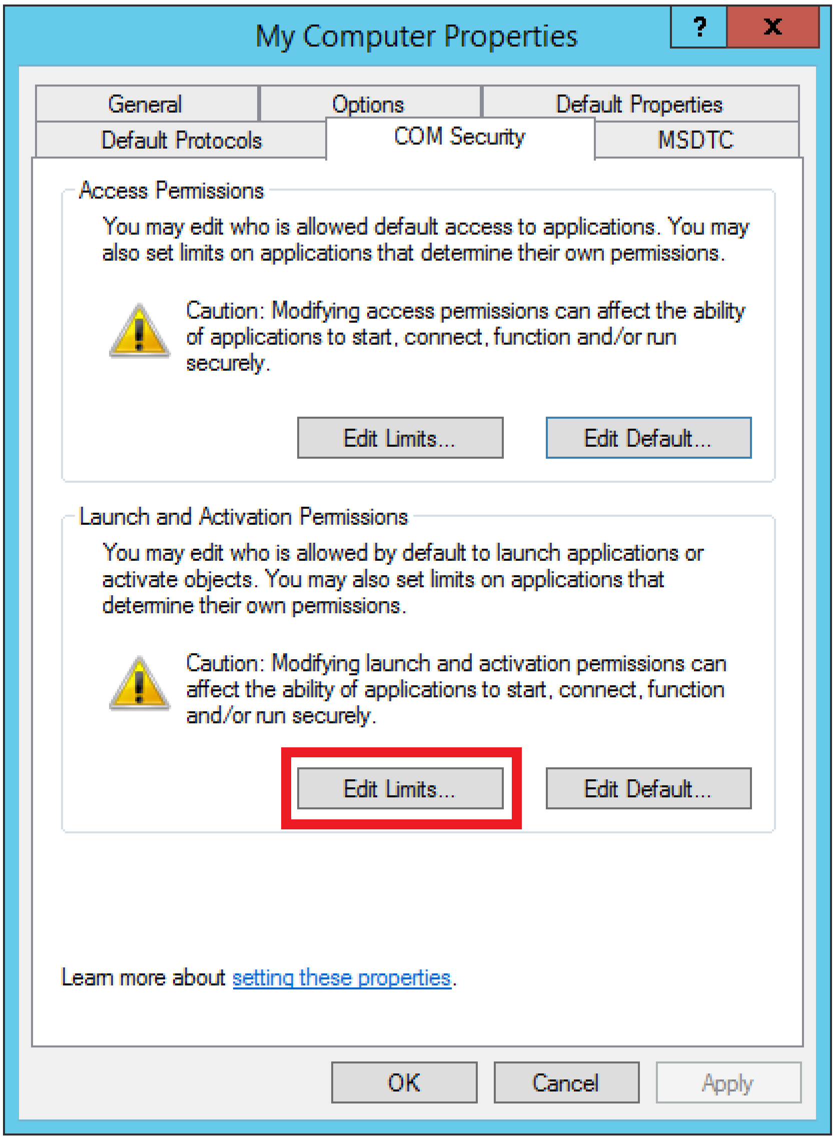
To configure HealthService Service SID for the monitoring of SQL Server Failover Cluster, take the following steps at each cluster node.

1. Launch mmc.exe and add the following two snap-ins:

   - *Component Services*
   - *WMI Control* (for local computer)

2. Expand **Component Services**, right-click **My Computer**, click **Properties** and go *Security* tab.

3. Click button **Edit Limits** in section *Launch and Activation Permissions*.



You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.

Caution: Modifying access permissions can affect the ability of applications to start, connect, function and/or run securely.

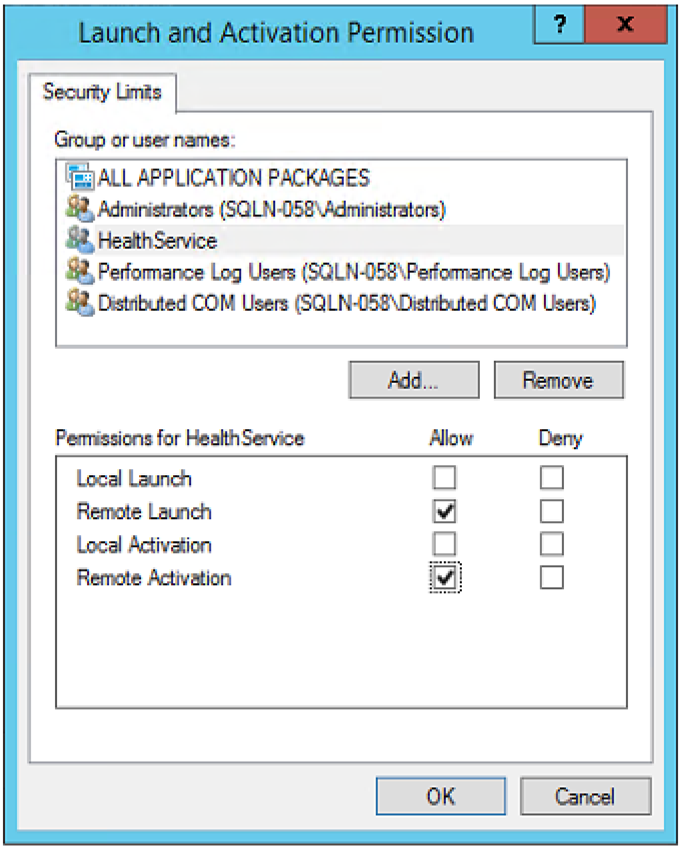
You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.

Caution: Modifying launch and activation permissions can affect the ability of applications to start, connect, function and/or run securely.

Learn more about setting these properties.

4. In *Launch and Activation Permission*, allow the following permissions for the “NT SERVICE\HealthService” account:

   - *Remote Launch*
   - *Remote Activation*



Group or user names:

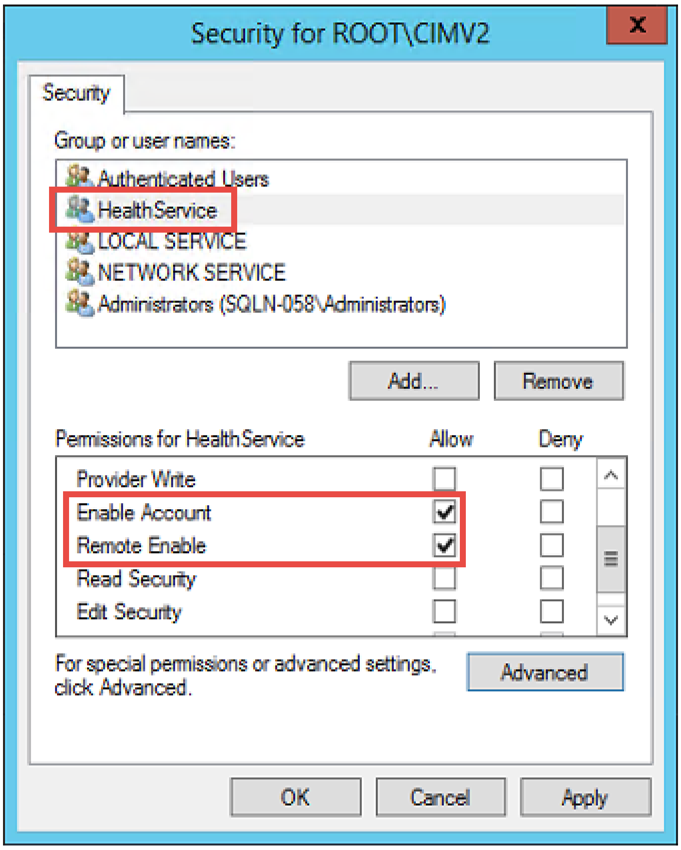
- ALL APPLICATION PACKAGES
- Administrators (SQLN-058\Administrators)
- HealthService
- Performance Log Users (SQLN-058\Performance Log Users)
- Distributed COM Users (SQLN-058\Distributed COM Users)

Permissions for HealthService	Allow	Deny
Local Launch	□	□
Remote Launch	☑	□
Local Activation	□	□
Remote Activation	☑	□

5. Go to snap-in *WMI Control* and call its properties, go to *Security* tab, select namespace *Root\CIMV2* and click button **Security**.

6. Allow the following permissions for the “NT SERVICE\HealthService” account:

   - *Enable Account*
   - *Remote Enable*



Group or user names:

- Authenticated Users
- HealthService
- LOCAL SERVICE
- NETWORK SERVICE
- Administrators (SQLN-058\Administrators)

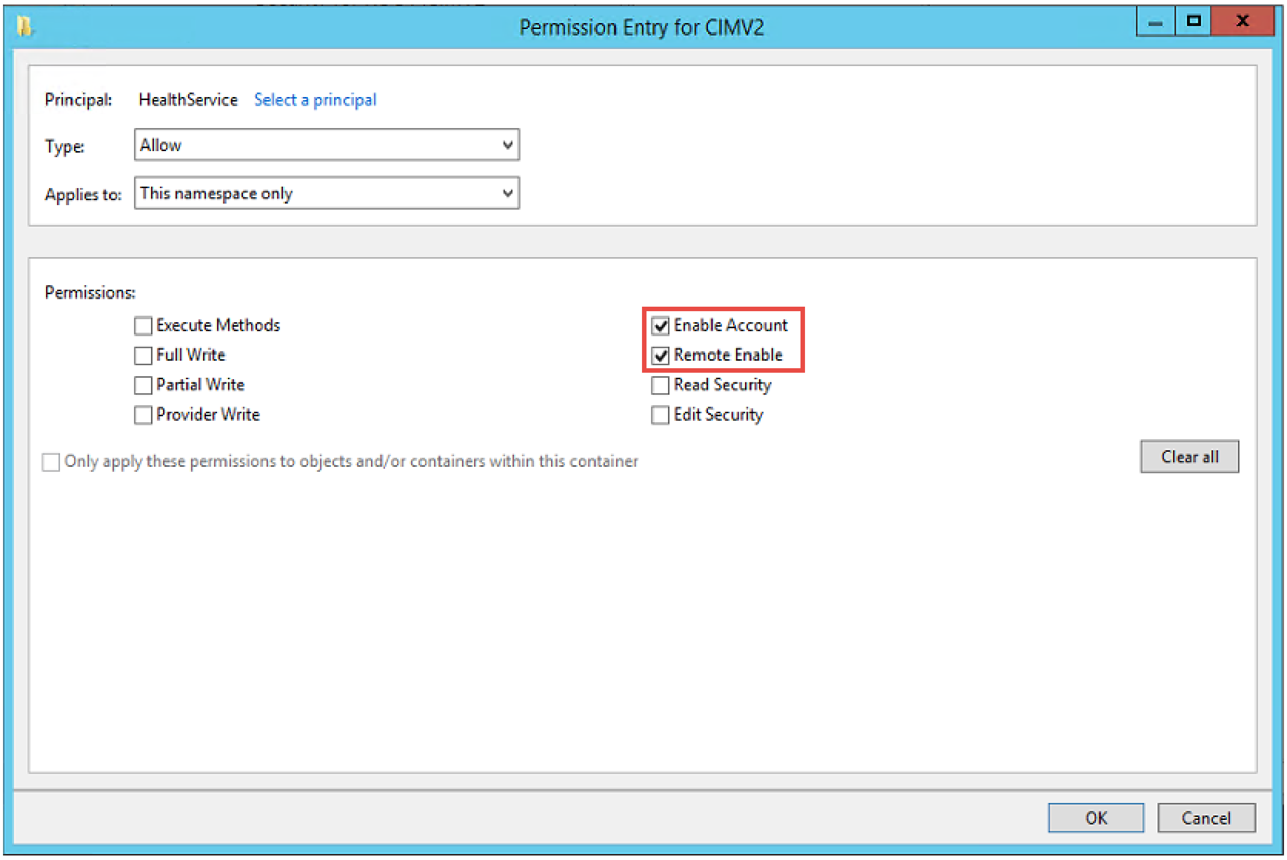
Permissions for HealthService	Allow	Deny
Provider Write	□	□
Enable Account	☑	□
Remote Enable	☑	□
Read Security	□	□
Edit Security	□	□

For special permissions or advanced settings, click Advanced.

7. Click button **Advanced**.

8. In *Permissions Entry for CIMV2*, select the "HealthService" account and click **Edit**, make sure *Applies to* is set to *This namespace only*, and enable the following permissions:

   - *Enable Account*
   - *Remote Enable*



Principal: HealthService

Type: Allow

Applies to: This namespace only

Permissions:

☐ Execute Methods
☐ Full Write
☐ Partial Write
☐ Provider Write
☑ Enable Account
☑ Remote Enable
☐ Read Security
☐ Edit Security

☐ Only apply these permissions to objects and/or containers within this container

Configuring SCOM SDK Run As Profile

This management pack needs the Author set of privileges on the SCOM SDK to be able to create a management pack and store overrides in it. If the default action account on SCOM does not have these permissions, make sure to have an account granted with them and map this account to the Microsoft SQL Server SCOM SDK Run As Profile.

Configuring Permissions for Always On Workflows when Servers Have Machine Names Longer than 15 Characters

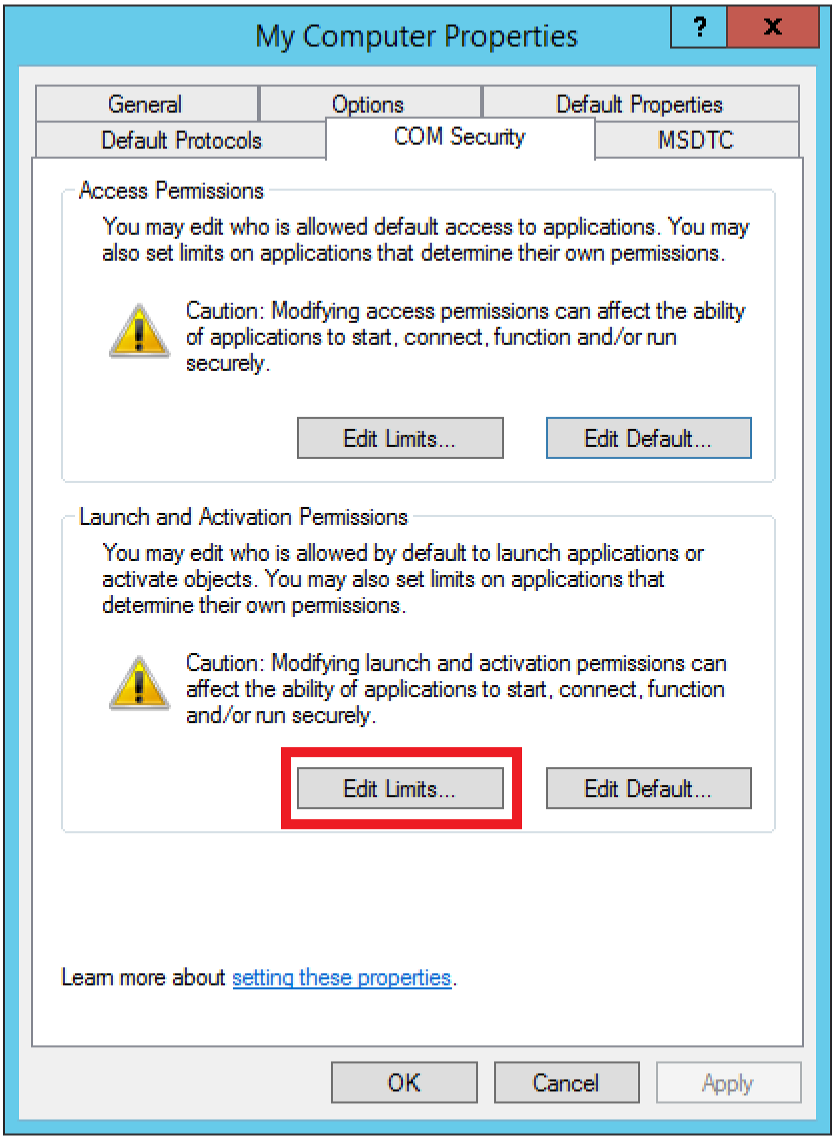
Please note that regardless of the used account (Local System or a Domain User account) and the method of rights granting, you should make sure that the account has the permissions listed below. The process of obtaining permissions is described below as a case when Local System account is used for monitoring.

Example: You have three replicas in your Availability Group, which are hosted on the following computers: comp1, comp2 and comp3. At that, comp1 hosts the primary replica. In this case, you should configure security settings for comp1 on comp2 and comp3 computers.

Note: If comp2 would host primary replica (after failover), other computers should also have configured WMI security for this computer. In general, you have to make sure that Local System account of each node, which can act as Primary one, have WMI permissions for the other nodes of the current Availability Group. The same is true for the Domain Action Account used for monitoring.

Therefore, below are the steps to configure security for configurations with Local System account (please note that in the provided instruction it is considered that SQLAON-020 computer hosts the primary replica).

1. Launch mmc.exe and add two Snap-Ins:
   - *Component Services*
   - *WMI Control* (for local computer)
2. Expand *Component Services*, right-click **My Computer**, click **Properties**, go to *COM Security* tab, and click button **Edit Limits** button in section *Launch and Activation Permissions*.



Access Permissions

You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.

Caution: Modifying access permissions can affect the ability of applications to start, connect, function and/or run securely.

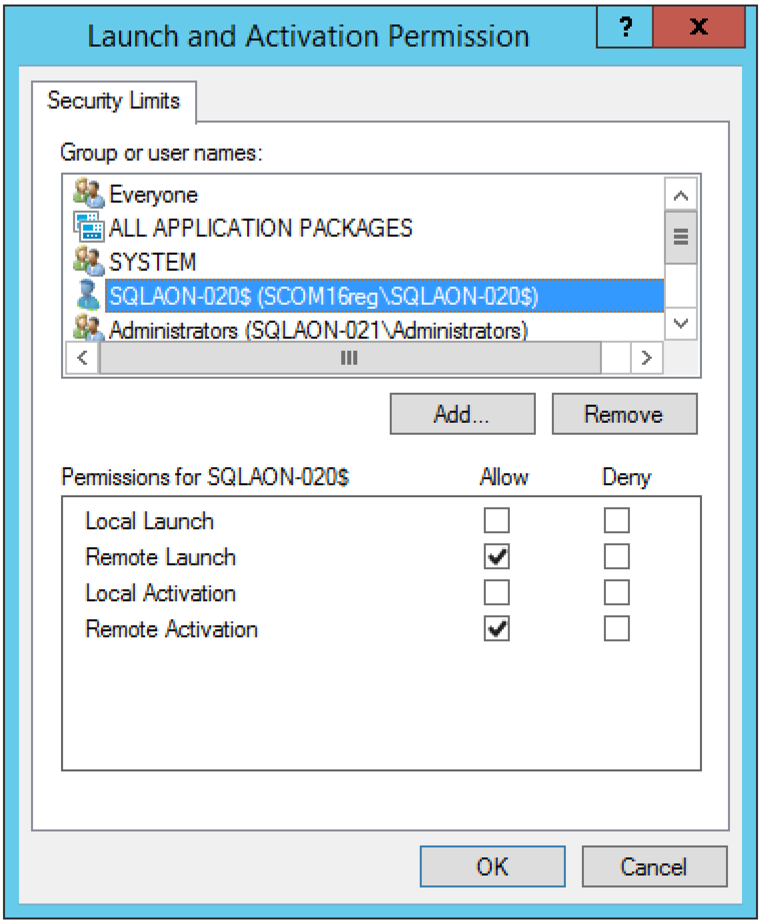
Launch and Activation Permissions

You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.

Caution: Modifying launch and activation permissions can affect the ability of applications to start, connect, function and/or run securely.

Learn more about setting these properties.

3. In form *Launch and Activation Permission*, allow the following permissions for the remote machine's account:
   - *Remote Launch*
   - *Remote Activation*



Group or user names:

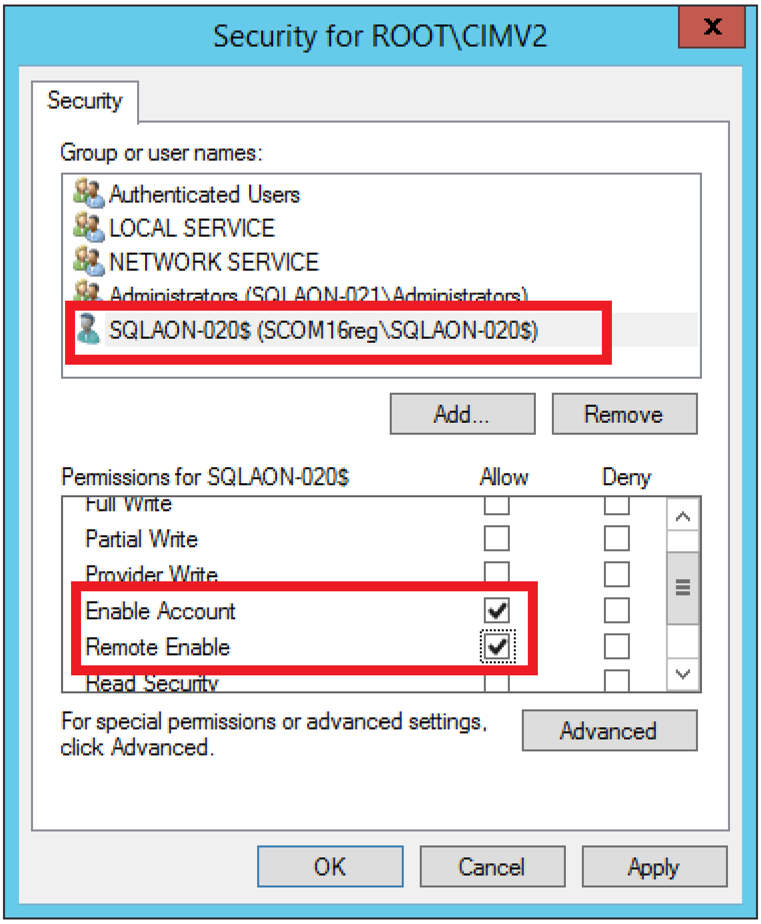
- Everyone
- ALL APPLICATION PACKAGES
- SYSTEM
- SQLAON-020$ (SCOM16reg\SQLAON-020$)
- Administrators (SQLAON-021\Administrators)

Permissions for SQLAON-020$	Allow	Deny
Local Launch	□	□
Remote Launch	☑	□
Local Activation	□	□
Remote Activation	☑	□

4. Go to *WMI Control* snap-In and open its properties, go to tab *Security*, select namespace *Root\CIMV2*, and click button **Security**.

5. Alow the following permissions for the target computer:

   - **Enable Account**
   - **Remote Enable**



Group or user names:

- Authenticated Users
- LOCAL SERVICE
- NETWORK SERVICE
- Administrators (SQLAON-021\Administrators)
- SQLAON-020$ (SCOM16reg\SQLAON-020$)

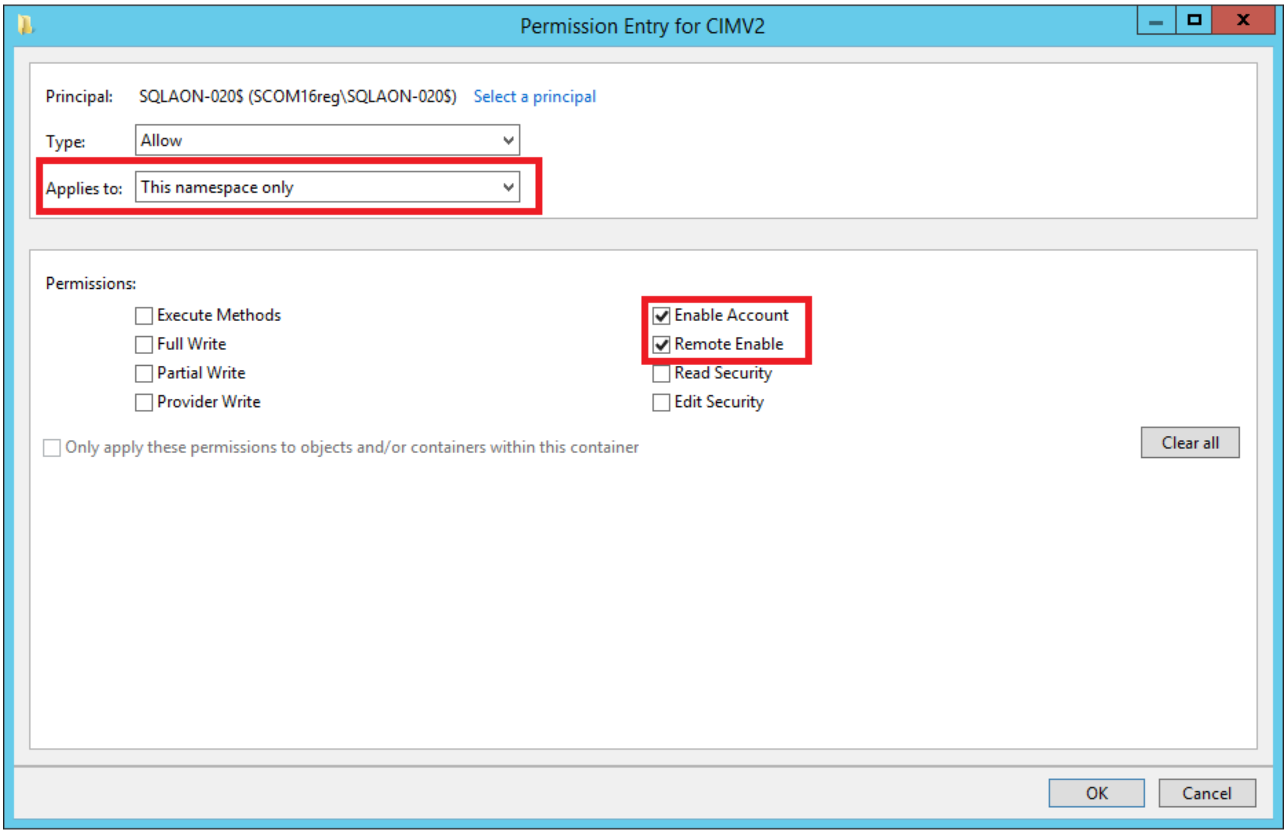
Permissions for SQLAON-020$	Allow	Deny
Full Write	□	□
Partial Write	□	□
Provider Write	□	□
Enable Account	☑	□
Remote Enable	☑	□
Read Security	□	□

For special permissions or advanced settings, click Advanced.

6. Click button *Advanced*, select the target account and click button **Edit**.

7. Make sure that parameter *Applies to* is set to *This namespace only* and enable the following permissions:

   - **Enable Account**
   - **Remote Enable**



Principal: SQLAON-020$ (SCOM16reg\SQLAON-020$)

Type: Allow

Applies to: This namespace only

Permissions:

- □ Execute Methods
- □ Full Write
- □ Partial Write
- □ Provider Write
- ☑ Enable Account
- ☑ Remote Enable
- □ Read Security
- □ Edit Security

□ Only apply these permissions to objects and/or containers within this container

Steps above should be taken on each replica participating in the target Availability Group.

Configuring Low-Privilege Monitoring

This section describes how to configure the Management Pack 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. To enable low-privilege monitoring, appropriate permissions should be granted to Run As accounts and these accounts should be bound to respective Run As profiles. Subsections below describe how to grant permissions at both Operating System and SQL Server level for all monitoring modes.

Low-Privilege Agent Monitoring

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

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

- **On Agents**

  - Grant accounts *SQLTaskAction* and *SQLMPLowPriv* with the Read permission on *HKLM:\Software\Microsoft\Microsoft SQL Server* registry path.

  - Add domain users *SQLTaskAction* and *SQLMonitor* to “EventLogReaders” local group.

  - 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.

  - Grant “Execute Methods”, “Enable Account”, “Remote Enable”, “Read Security” permissions to *SQLTaskAction* and *SQLMPLowPriv* for these WMI namespaces:

    - `root`
    - `root\cimv2`
    - `root\default`
    - `root\Microsoft\SqlServer\ComputerManagement11` *(if exists)*
    - `root\Microsoft\SqlServer\ComputerManagement12` *(if exists)*
    - `root\Microsoft\SqlServer\ComputerManagement13` *(if exists)*
    - `root\Microsoft\SqlServer\ComputerManagement14` *(if exists)*
    - `root\Microsoft\SqlServer\ComputerManagement15` *(if exists)*

  - Grant *SQLMPLowPriv* with the Read permission on *HKLM:\Software\Microsoft\Microsoft SQL Server\*`[InstanceID]`*\MSSQLServer\Parameters* registry path on each monitored instance.

- **Additional steps for cluster SQL Server instances**

  - Take steps above for each node in a cluster.

  - Grant *SQLMPLowPriv* and *SQLTaskAction* with “Remote Launch” and “Remote Activation” DCOM permissions using DCOMCNFG. Please note that both defaults and limits should be adjusted.

  - Allow Windows Remote Management through the Windows Firewall.

  - Grant “Full Control” access for the cluster to the *SQLMPLowPriv* using Failover Cluster Manager.

  - Grant “Execute Methods”, “Enable Account”, “Remote Enable”, “Read Security” permissions to *SQLTaskAction* and *SQLMPLowPriv* for WMI namespace *root\MSCluster*.

- **On SQL Server instances**

  - Open SQL Server Management Studio and connect to the instance of SQL Server Database Engine.

  - In SQL Server Management Studio, for each instance of SQL Server Database Engine running on a monitored server, create a login for both *SQLMPLowPriv* and *SQLTaskAction*.

  - Create *SQLMPLowPriv* and *SQLTaskAction* users in each user database, master, msdb, and model. Link *SQLMPLowPriv* users to *SQLMPLowPriv* login and *SQLTaskAction* users to *SQLTaskAction* login.

```
--This script is an example of the creation new users
--  in database msdb. Make sure to execute such a script
--  for every database on each SQL instance.
use msdb
go
CREATE USER [SQLMPLowPriv] FOR LOGIN [SQLMPLowPriv]
CREATE USER [SQLTaskAction] FOR LOGIN [SQLTaskAction]
```

  - Grant *SQLMPLowPriv* with the following permissions:

```
use master
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]

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]
```

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

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

- **On SMB Shares**

  - Grant share permissions by opening share properties dialog for the share, which hosts SQL Server data files or SQL Server transaction log files.

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

- **Optional steps for tasks on Agents**

  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.

    - 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, ntegration Services): on the agent machine, grant the *SQLTaskAction* user permission to start or stop an NT service This involves setting a ervice’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 *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;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD)

    - In SQL Server Management Studio, add *SQLTaskAction* to db_owner database role for each database if the task is related to performing database checks:

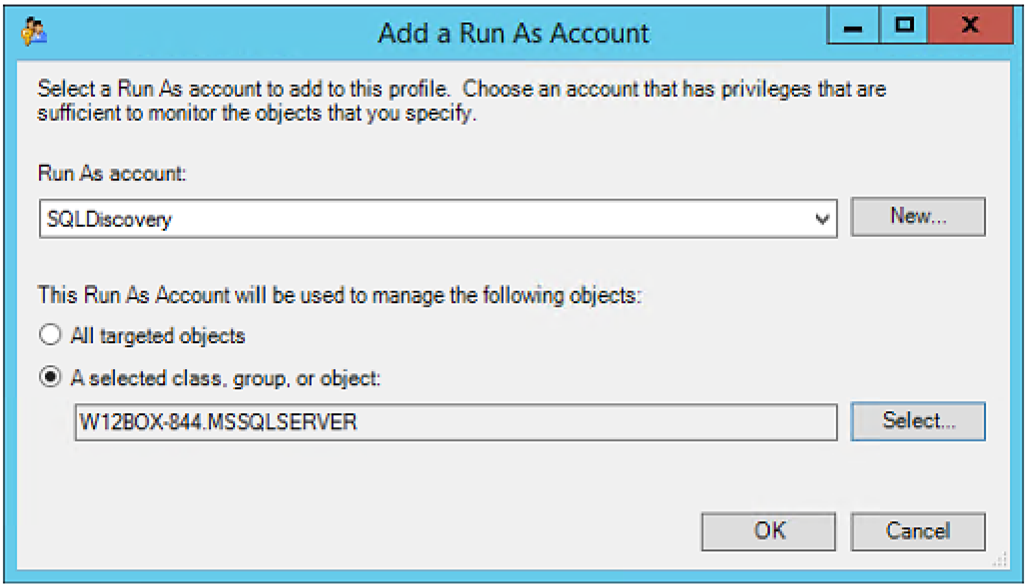
        - “Check Catalog (DBCC)”
        - “Check Database (DBCC)”
        - “Check Disk (DBCC)” (invokes DBCC CHECKALLOC)

      ```
      USE msdb
      GO
      ALTER ROLE [db_owner] ADD MEMBER [SQLTaskAction]
      ```

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

```
USE master
GO
GRANT ALTER ANY DATABASE TO [SQLTaskAction]
```

- **On System Center Operations Manager**
  - Import the SQL Server Management Pack if it has not been imported.
  - Create an *SQLTaskAction*, *SQLDiscovery* and *SQLMonitor* Run As accounts with "Windows" account type. For more information about creation of a Run As account, see How to Create Run As Account in Operations Manager 2012. For more information about various Run As Account types, see Managing Run As Accounts and Profiles in Operations Manager 2012.
  - On the System Center Operations Manager console, configure the Run As profiles as follows:
    - Set the "Microsoft SQL Server Task Run As Profile" Run As profile to use the *SQLTaskAction* Run As account.
    - Set the "Microsoft SQL Server Discovery Run As Profile" Run As profile to use the *SQLDiscovery* Run As account.
    - Set the "Microsoft SQL Server Monitoring Run As Profile" Run As profile to use the *SQLMonitor* Run As account.
  - 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.



Add a Run As Account

Select a Run As account to add to this profile. Choose an account that has privileges that are sufficient to monitor the objects that you specify.

Run As account: SQLDiscovery

This Run As Account will be used to manage the following objects:

○ All targeted objects

● A selected class, group, or object:

W12BOX-844.MSSQLSERVER

Low-Privilege Agentless Monitoring

To configure Low-Privilege Agentless Monitoring, perform the steps described below.

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

- **On SQL Instance**
  - Open SQL Server Management Studio and connect to the instance of SQL Server Database Engine.
  - 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
ALTER ROLE [db_datareader] ADD MEMBER [SQLMPLowPriv]
GRANT EXECUTE ON xp_readerrorlog to [SQLMPLowPriv]
GO
```

  - Create a user in each user database, master, msdb, and model. Link created users to login SQLMPLowPriv.

```
--This script is an example of the creation new users
--  in database msdb. Make sure to execute such a script
--  for every database on each SQL instance.
use msdb
go
CREATE USER [SQLMPLowPriv] FOR LOGIN [SQLMPLowPriv]
```

  - For msdb database, grant the user with 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 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]
```

- 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.

  - 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:

    - “Check Catalog (DBCC)”
    - “Check Database (DBCC)”
    - “Check Disk (DBCC)” (invokes DBCC CHECKALLOC)

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

  - Grant the ALTER ANY DATABASE privilege to SQL Login *SQLMPLowPriv* to performing database tasks:

    - “Set Database Online”
    - “Set Database Offline”
    - “Set Database to Emergency State”

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

  - 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
    ```

Using Add Monitoring Wizard

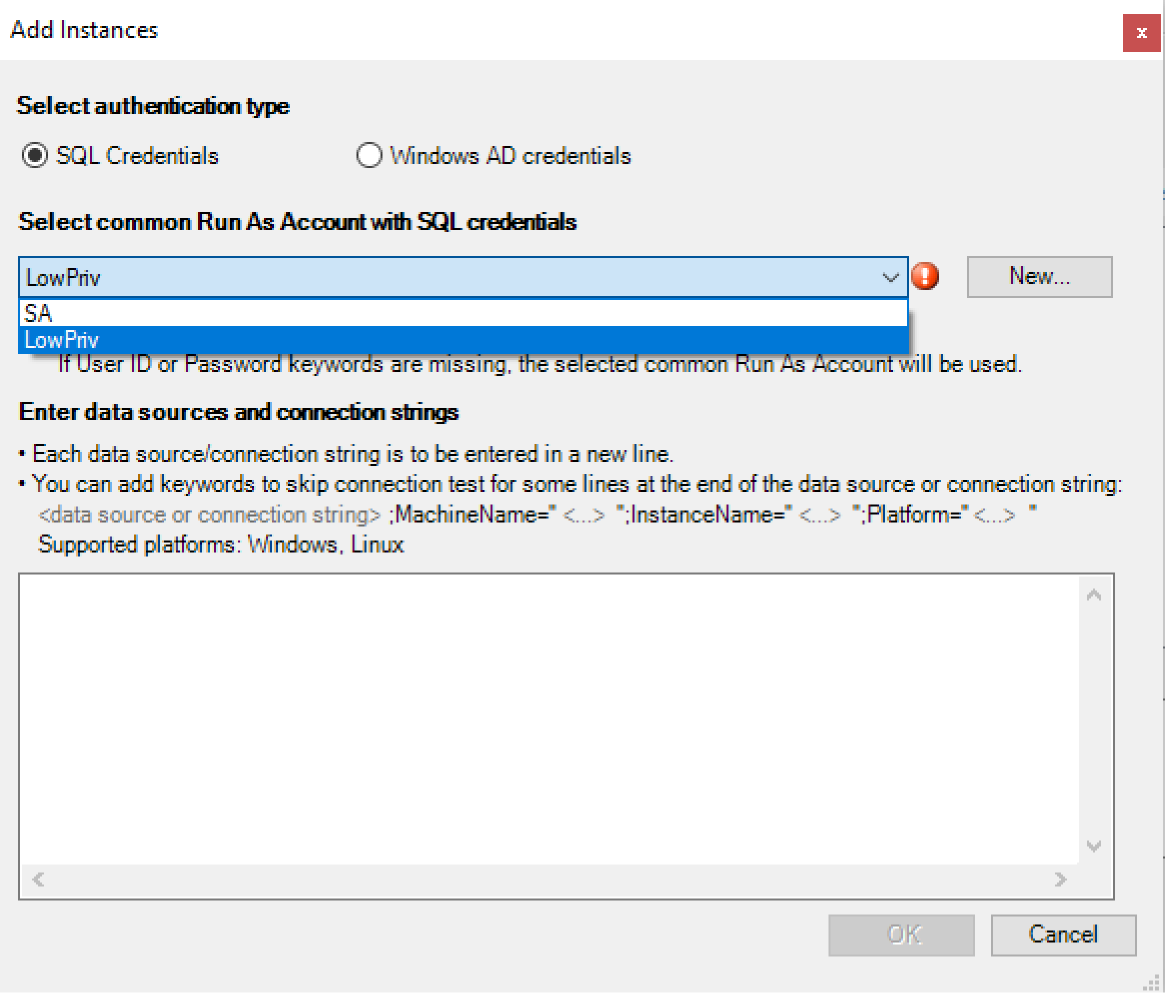
To configure low-privilege agentless monitoring using the **Add Monitoring Wizard**, perform the steps provided in the Configuring Agentless Monitoring Mode section but with the following changes:

1. In the **Add Monitoring Wizard** window, click **Add Instances**.

2. In the **Add Instances** window, select a common Run As account with the appropriate SQL low-privilege login and specify data sources and/or connection strings. For example:

   - 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"

Make sure to follow the instructions provided in this window to avoid errors.



Add Instances

Select authentication type

● SQL Credentials ○ Windows AD credentials

Select common Run As Account with SQL credentials

LowPriv

SA

LowPriv

If User ID or Password keywords are missing, the selected common Run As Account will be used.

Enter data sources and connection strings

- Each data source/connection string is to be entered in a new line.
- You can add keywords to skip connection test for some lines at the end of the data source or connection string:

  <data source or connection string> ;MachineName=" <...> ";InstanceName=" <...> ";Platform=" <...> "

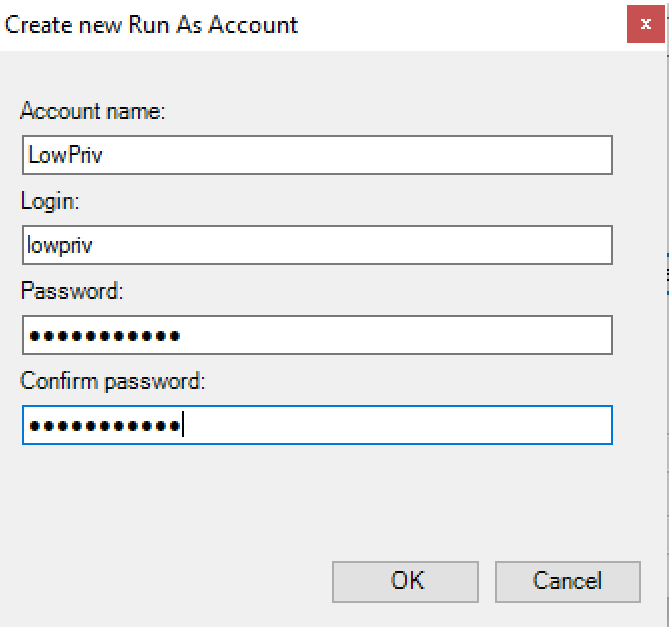
  Supported platforms: Windows, Linux

If you want to create a new Run As account, do the following:

1. In the **Add Instances** window, click **New...**.

2. Enter a new name for the Run As account.

3. Specify credentials to access the SQL Server that you want to monitor, click **OK** and wait until connection is established.



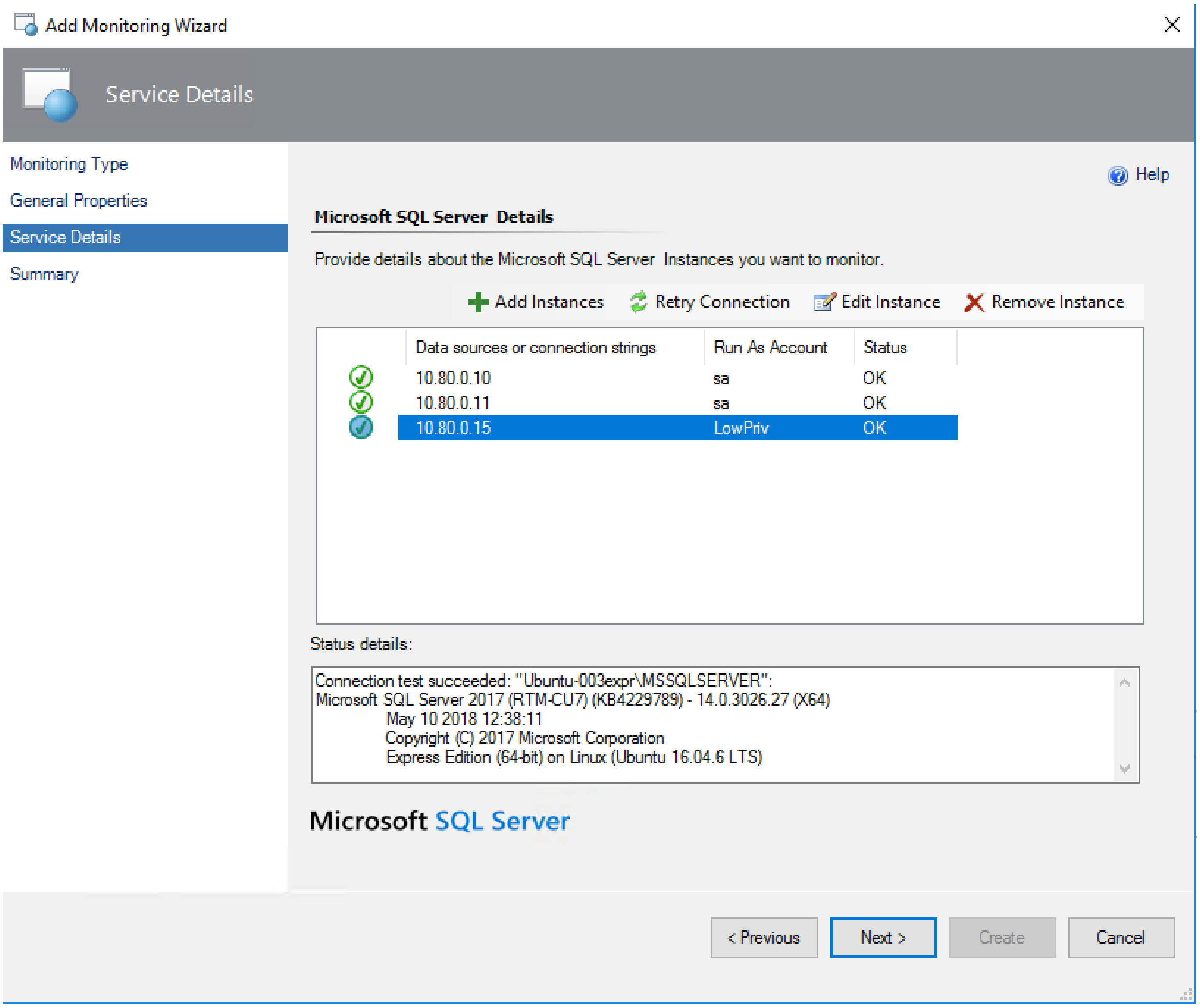
Account name: LowPriv

Login: lowpriv

Password: ●●●●●●●●●●●

Confirm password: ●●●●●●●●●●●

After connection is established, you can view and edit properties of the added instance.



Microsoft SQL Server Details

Provide details about the Microsoft SQL Server Instances you want to monitor.

	Data sources or connection strings	Run As Account	Status
✓	10.80.0.10	sa	OK
✓	10.80.0.11	sa	OK
✓	10.80.0.15	LowPriv	OK

Status details:

```
Connection test succeeded: "Ubuntu-003expr\MSSQLSERVER":
Microsoft SQL Server 2017 (RTM-CU7) (KB4229789) - 14.0.3026.27 (X64)
        May 10 2018 12:38:11
        Copyright (C) 2017 Microsoft Corporation
        Express Edition (64-bit) on Linux (Ubuntu 16.04.6 LTS)
```

Low-Privilege Mixed Monitoring

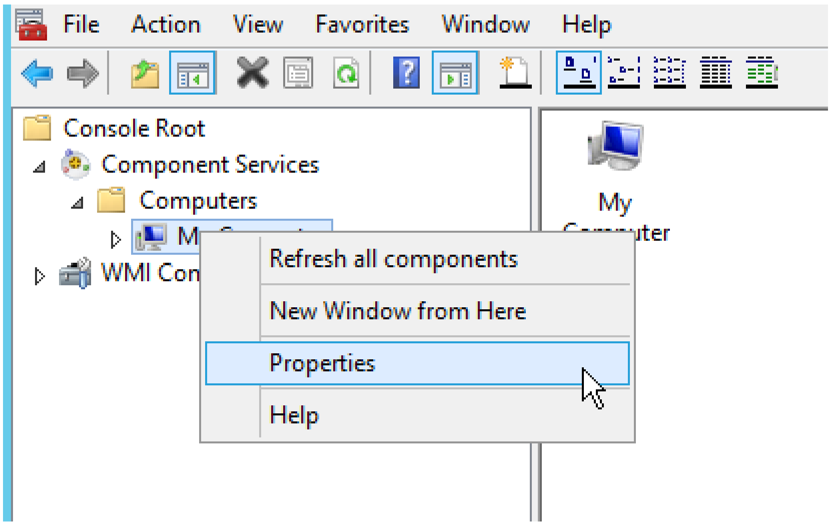
To configure low-privilege environments for mixed monitoring, perform the steps described in the Low-Privilege Agent Monitoring section and then do the following:

- Configure remote access to WMI
- Grant permissions to get information about the services
- Use a registry key to manage the remote access to the registry

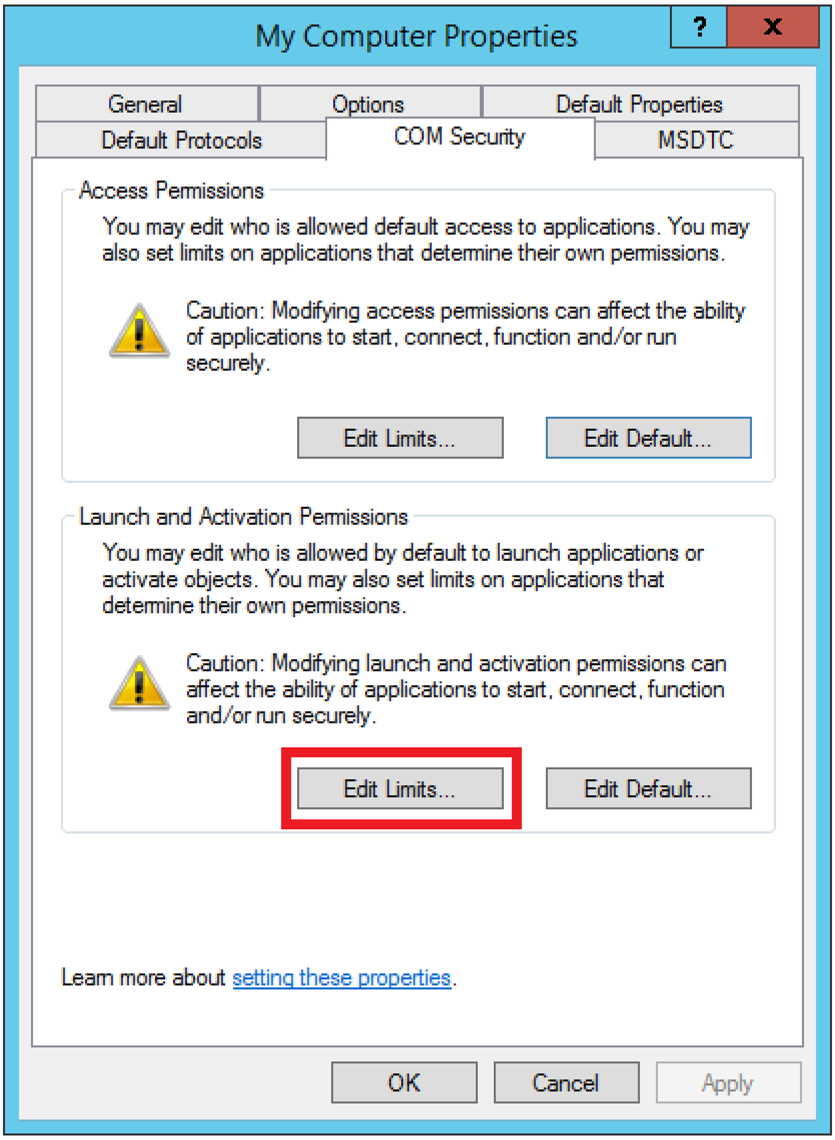
Managing Remote Access to WMI

To configure security for configurations with low-privilege accounts, perform the following steps on each mixed mode monitoring server:

1. Launch the **mmc.exe** console and add the following snap-ins:
   - *Component Services*
   - *WMI Control* (for a local computer)
2. Expand **Component Services**, right-click **My Computer** and click **Properties**.



3. Open the **Security** tab.
4. In the **Launch and Activation Permissions** section, click **Edit Limits**.



My Computer Properties

Access Permissions

You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.

Caution: Modifying access permissions can affect the ability of applications to start, connect, function and/or run securely.

Launch and Activation Permissions

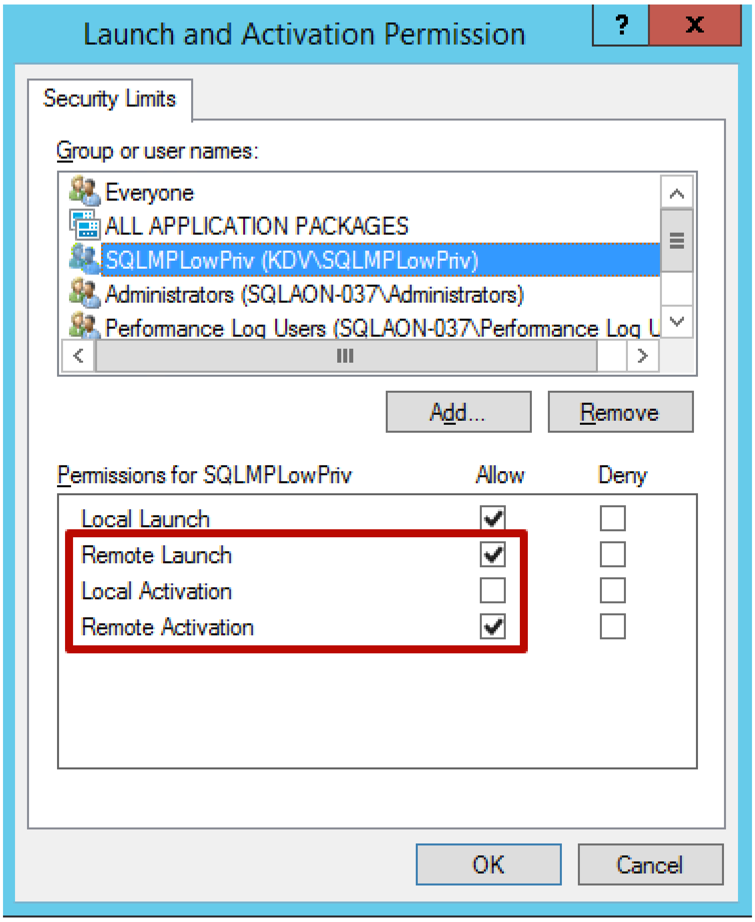
You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.

Caution: Modifying launch and activation permissions can affect the ability of applications to start, connect, function and/or run securely.

Learn more about setting these properties.

5. Set the following permissions for the remote machine account:

   - *Remote Launch*
   - *Remote Activation*



Group or user names:

- Everyone
- ALL APPLICATION PACKAGES
- SQLMPLowPriv (KDV\SQLMPLowPriv)
- Administrators (SQLAON-037\Administrators)
- Performance Log Users (SQLAON-037\Performance Log U

Permissions for SQLMPLowPriv	Allow	Deny
Local Launch	☑	☐
Remote Launch	☑	☐
Local Activation	☐	☐
Remote Activation	☑	☐

6. Go to the **WMI Control** snap-in and call its properties.

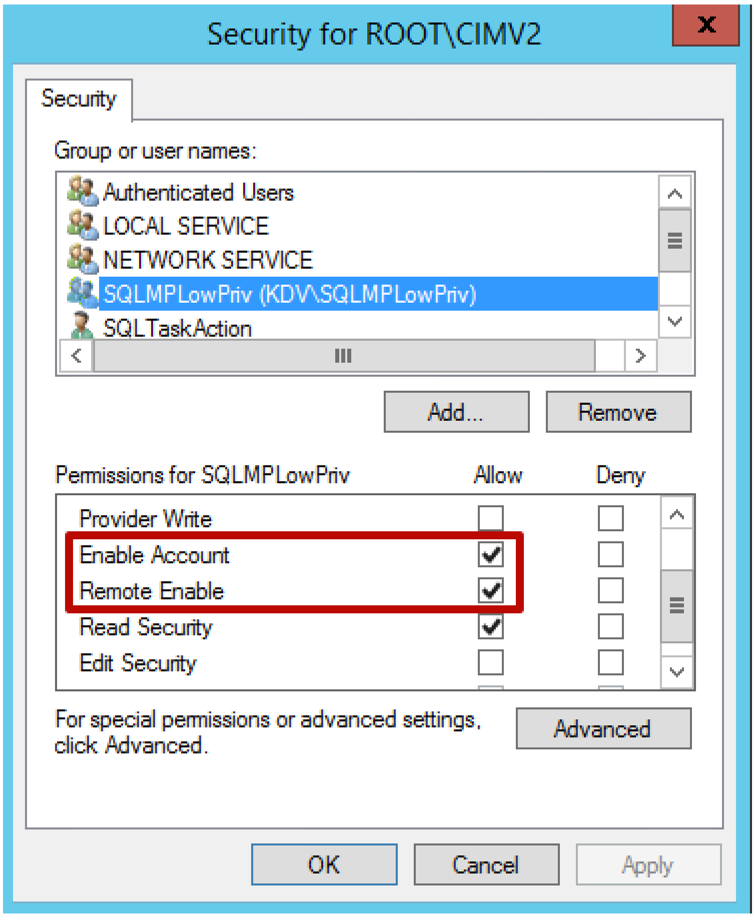
7. Open the **Security** tab and select the following namespaces:

   - *Root\CIMV2, Root\Microsoft\SqlServer*
   - *Root\Microsoft\SqlServer\ComputerManagement11* (if exists)
   - *Root\Microsoft\SqlServer\ComputerManagement12* (if exists)
   - *Root\Microsoft\SqlServer\ComputerManagement13* (if exists)
   - *Root\Microsoft\SqlServer\ComputerManagement14* (if exists)
   - *Root\Microsoft\SqlServer\ComputerManagement15* (if exists)

8. Click **Security**.

9. Add the following permissions for the target computer:

   - *Enable Account*
   - *Remote Enable*



Group or user names:

- Authenticated Users
- LOCAL SERVICE
- NETWORK SERVICE
- SQLMPLowPriv (KDV\SQLMPLowPriv)
- SQLTaskAction

Permissions for SQLMPLowPriv	Allow	Deny
Provider Write	□	□
Enable Account	☑	□
Remote Enable	☑	□
Read Security	☑	□
Edit Security	□	□

For special permissions or advanced settings, click Advanced.

10. Click **Advanced**.

11. Select the target account and click **Edit**.

12. Make sure that the **Applies to the** parameter is set to **This namespace only** and the following permissions are set:

    - *Enable Account*
    - *Remote Enable*

Granting Permissions

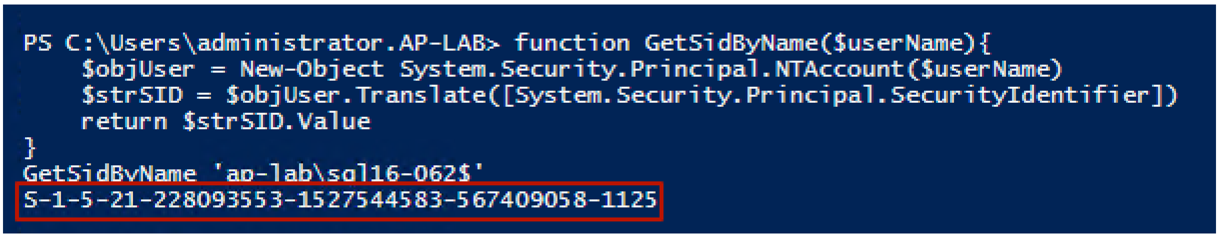
To get information about services, grant required permissions according to the following steps:

1. Open the PowerShell console.

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

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

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



```
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 $strSID.Value
}
GetSidByName 'ap-lab\sql16-062$'
S-1-5-21-228093553-1527544583-567409058-1125
```

3. From the Windows command prompt, run the following command to retrieve the current SDDL for the Services Control Manager.

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

   The SDDL is saved to the **file.txt** file and looks similar to the following one: 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). For more information, see Microsoft KB914392

4. Modify the SDDL string by copying the SDDL section that ends in **IU** (Interactive Users).

   This section is enclosed in parentheses (i.e. A;;CCLCRPRC;;;IU). Paste this clause directly after the clause you have copied.

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

   The new SDDL looks similar to the following one: 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)

5. Set security credentials for accessing the Service Control Manager by using the **sdset** command.

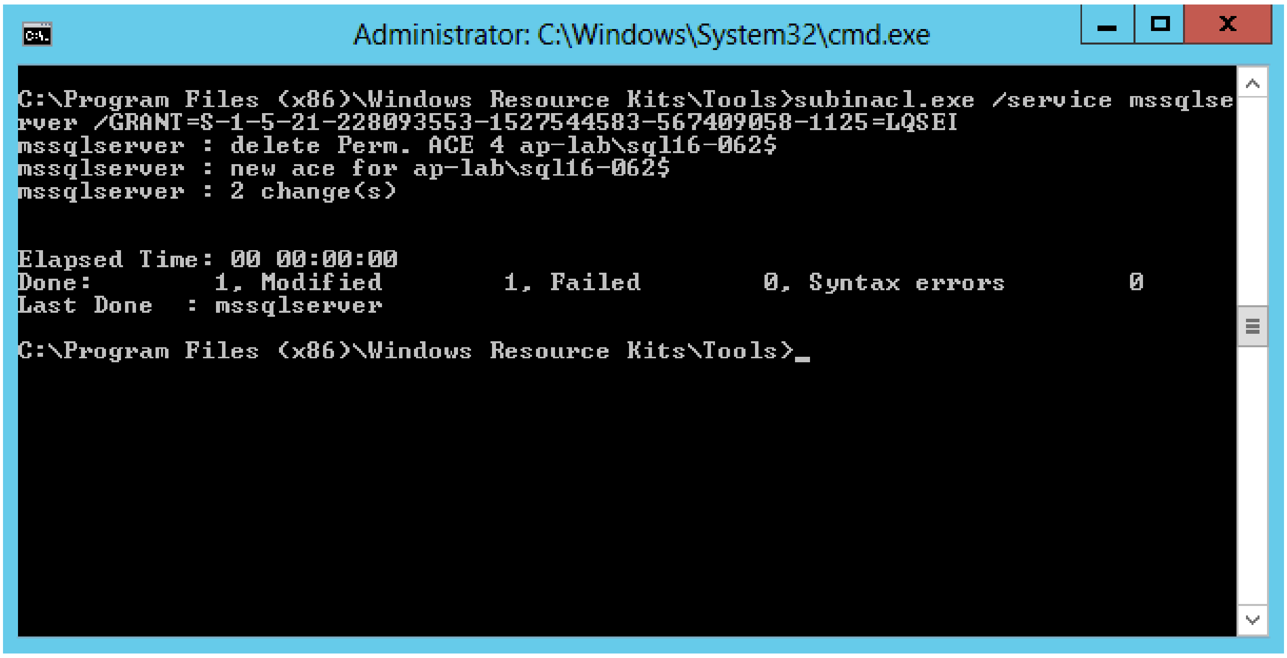
   > ⚠ Note that the permissions on **scmanager** are being replaced. Setting security credentials is not additive. That is why we needed to copy the existing permissions.

   `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)"

6. Set the rights for 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-1-5-21-214A909598-1293495619-13Z157935-75714=LQSEI
   - subinacl.exe /service sqlserveragent /GRANT= S-1-5-21-214A909598-1293495619-13Z157935-75714=LQSEI
   - subinacl.exe /service mssqlfdlauncher /GRANT= S-1-5-21-214A909598-1293495619-13Z157935-75714=LQSEI



```
C:\Program Files (x86)\Windows Resource Kits\Tools>subinacl.exe /service mssqlse
rver /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>_
```

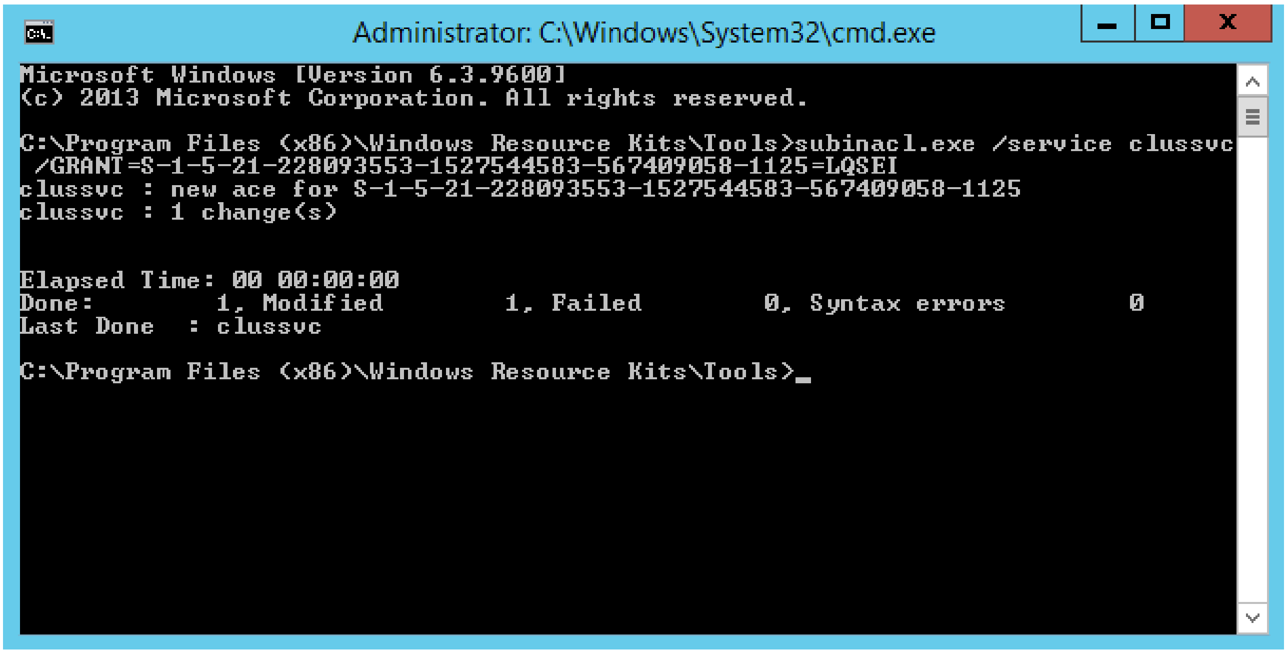
The following rights have the following meaning:

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

7. Set the rights for 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



```
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>_
```

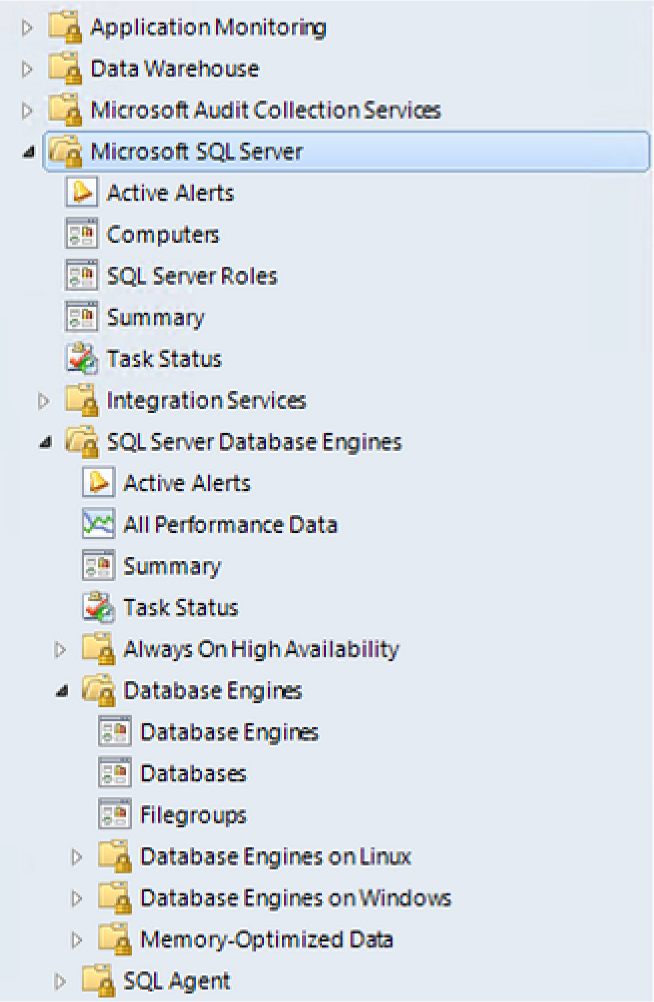
Create a registry key to manage remote access to the registry.

If you need to create a key to restrict access to the registry, follow the following steps:

1. Start **Registry Editor** (Regedt32.exe) and locate the following key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control
2. In the **Edit** menu, click **Add Key** and 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 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 Key** and 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 edit the current permissions or add users or groups to whom you want to grant access.
9. Quit **Registry Editor** and restart Windows.

Version-Independent (Generic) Views and Dashboards

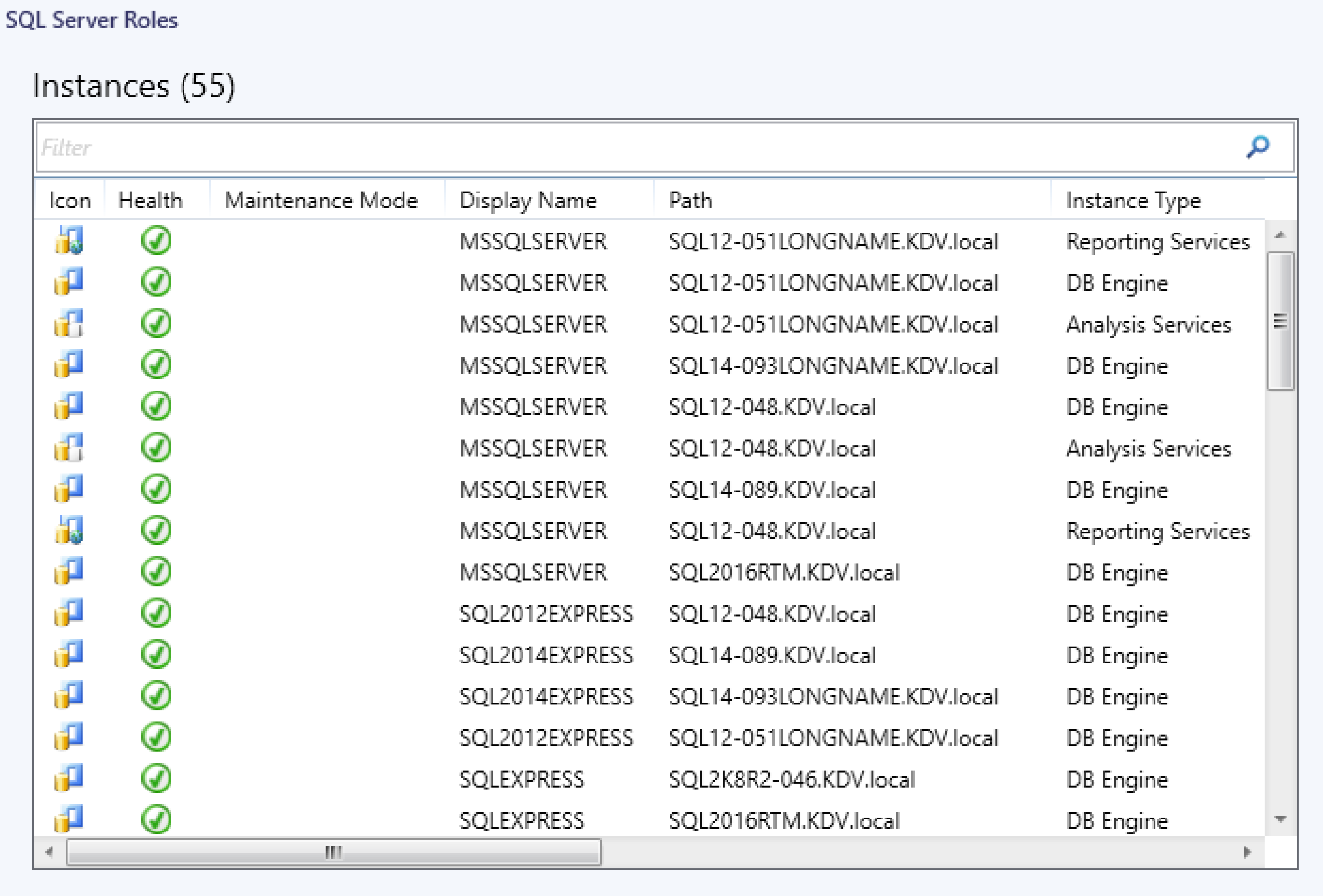
This management pack introduces a common folder structure, which will be used by future releases of management packs for different components of SQL Server. The following views and dashboards are version-independent and show information about all versions of SQL Server:



- Application Monitoring
- Data Warehouse
- Microsoft Audit Collection Services
- Microsoft SQL Server
  - 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
      - Database Engines on Linux
      - Database Engines on Windows
      - Memory-Optimized Data
    - SQL Agent

> ⚠ Note that the **Computers** node view displays computers on which the agents are installed and the management pack discovery is running. This view does not display computers configured for agentless monitoring.

The **SQL Server Roles** dashboard provides 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

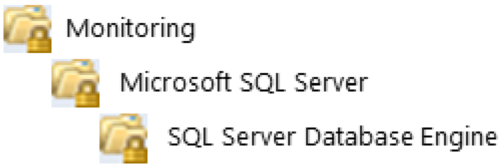
Instances (55)

Filter

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

SQL Server Views

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

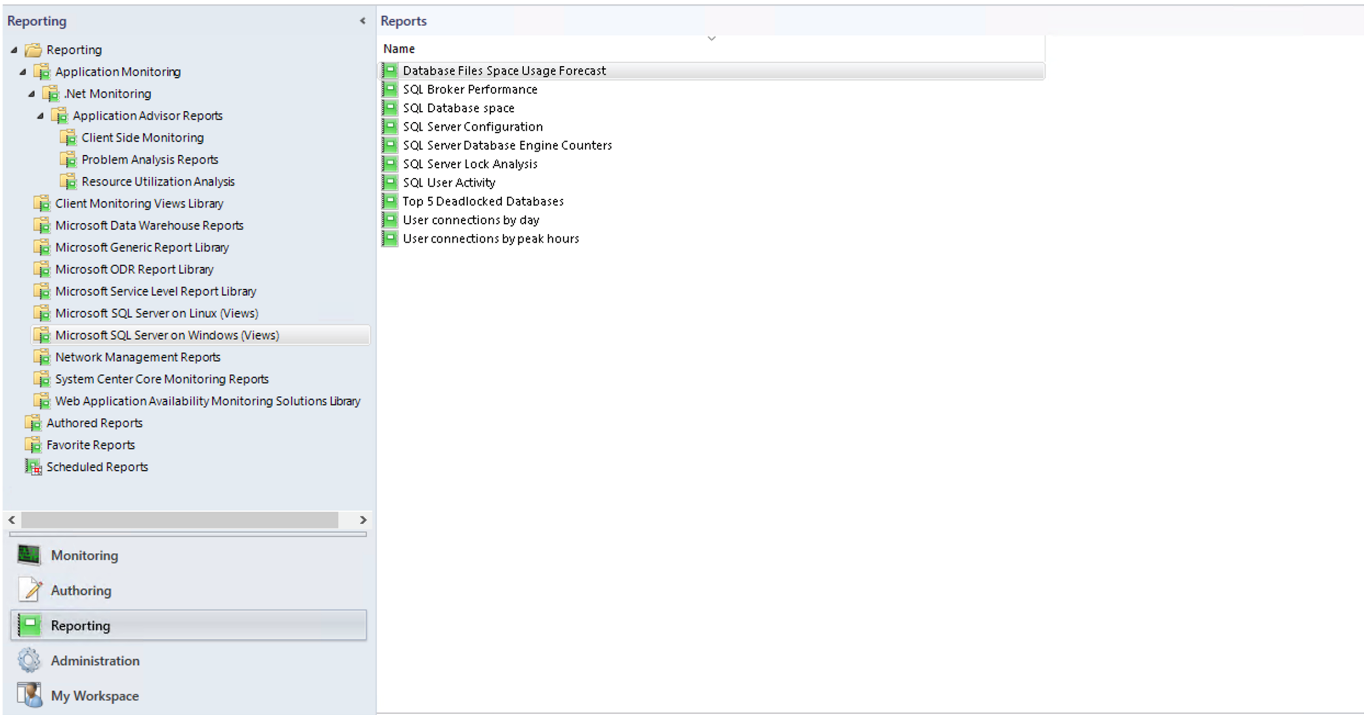


Monitoring
- Microsoft SQL Server
  - SQL Server Database Engine

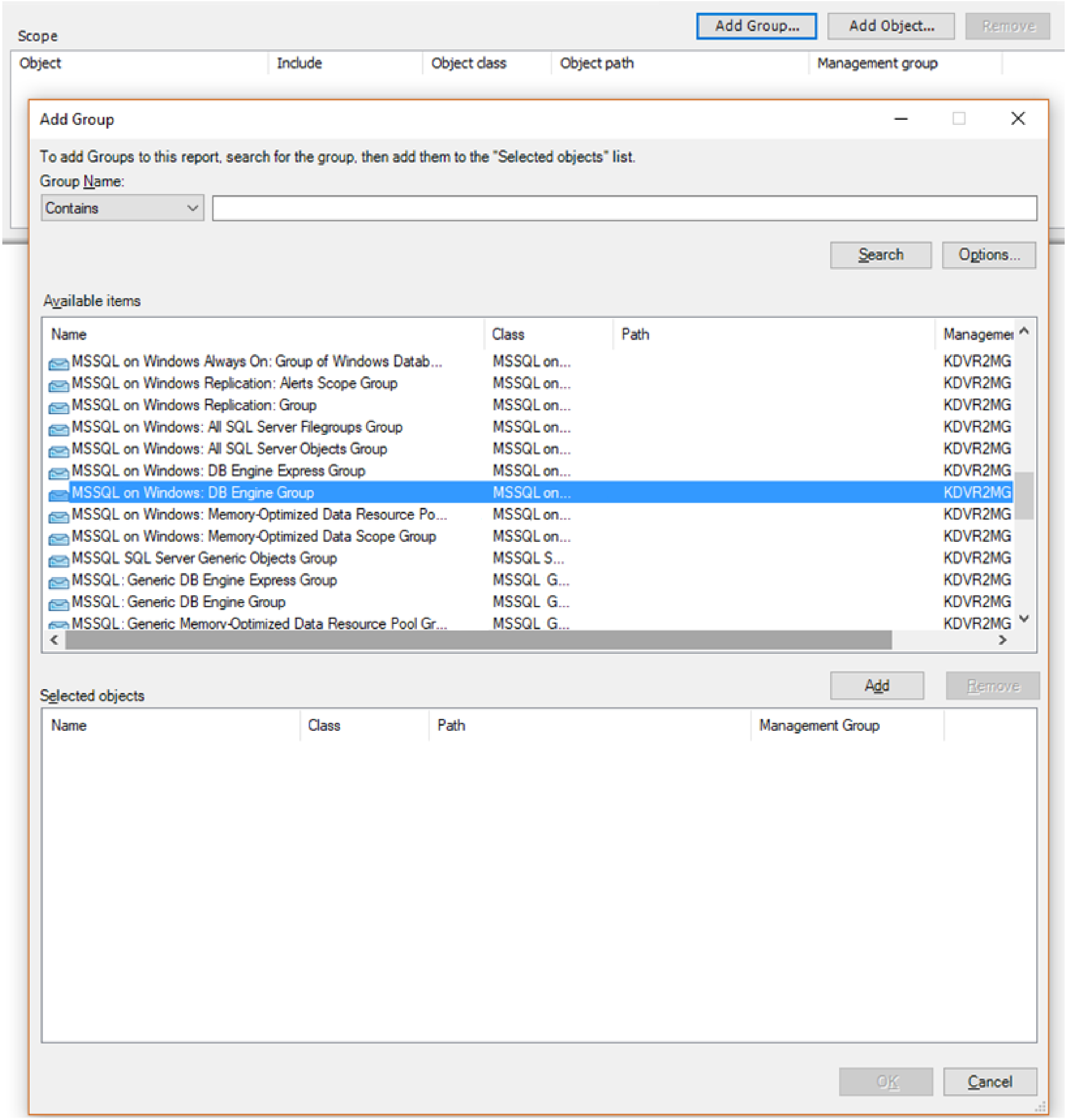
> ⚠ Some views may contain a very long list of objects and metrics. To find a specific object or group of objects, you can use the **Scope**, **Search**, and **Find** buttons on the Operations Manager toolbar. For more information, see the Finding Data and Objects in the Operations Manager Consoles article.

SQL Server Reporting

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


Reporting
Reporting
Application Monitoring
.Net Monitoring
Application Advisor Reports
Client Side Monitoring
Problem Analysis Reports
Resource Utilization Analysis
Client Monitoring Views Library
Microsoft Data Warehouse Reports
Microsoft Generic Report Library
Microsoft ODR Report Library
Microsoft Service Level Report Library
Microsoft SQL Server on Linux (Views)
Microsoft SQL Server on Windows (Views)
Network Management Reports
System Center Core Monitoring Reports
Web Application Availability Monitoring Solutions Library
Authored Reports
Favorite Reports
Scheduled Reports
Monitoring
Authoring
Reporting
Administration
My Workspace
Reports
Name
Database Files Space Usage Forecast
SQL Broker Performance
SQL Database space
SQL Server Configuration
SQL Server Database Engine Counters
SQL Server Lock Analysis
SQL User Activity
Top 5 Deadlocked Databases
User connections by day
User connections by peak hours

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:



Scope

Object	Include	Object class	Object path	Management group

To add Groups to this report, search for the group, then add them to the "Selected objects" list.

Group Name: Contains

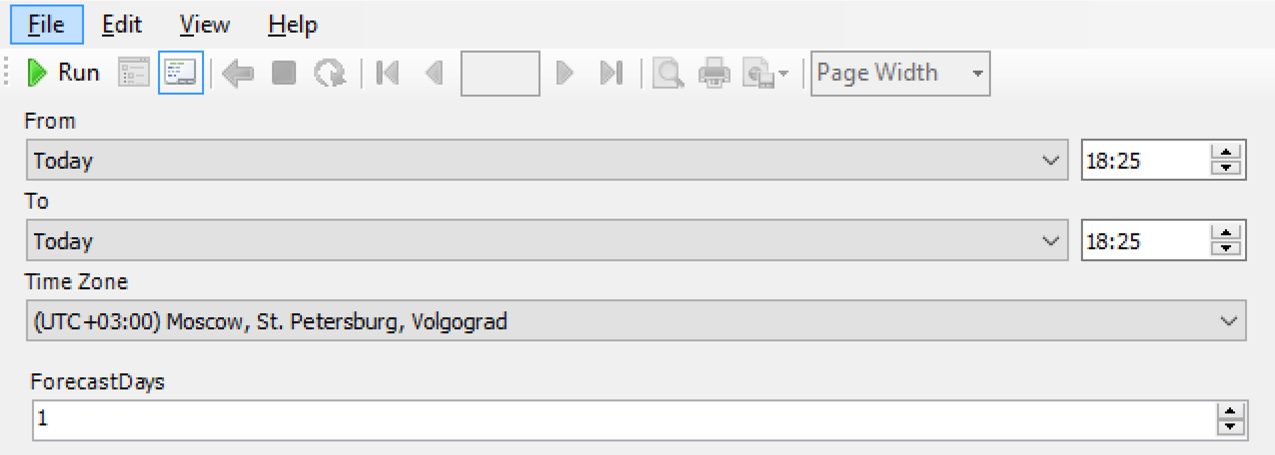
Available items

Name	Class	Path	Managemer
MSSQL on Windows Always On: Group of Windows Datab...	MSSQL on...		KDVR2MG
MSSQL on Windows Replication: Alerts Scope Group	MSSQL on...		KDVR2MG
MSSQL on Windows Replication: Group	MSSQL on...		KDVR2MG
MSSQL on Windows: All SQL Server Filegroups Group	MSSQL on...		KDVR2MG
MSSQL on Windows: All SQL Server Objects Group	MSSQL on...		KDVR2MG
MSSQL on Windows: DB Engine Express Group	MSSQL on...		KDVR2MG
MSSQL on Windows: DB Engine Group	MSSQL on...		KDVR2MG
MSSQL on Windows: Memory-Optimized Data Resource Po...	MSSQL on...		KDVR2MG
MSSQL on Windows: Memory-Optimized Data Scope Group	MSSQL on...		KDVR2MG
MSSQL SQL Server Generic Objects Group	MSSQL S...		KDVR2MG
MSSQL: Generic DB Engine Express Group	MSSQL G...		KDVR2MG
MSSQL: Generic DB Engine Group	MSSQL G...		KDVR2MG
MSSQL: Generic Memory-Optimized Data Resource Pool Gr...	MSSQL G...		KDVR2MG

Selected objects

Name	Class	Path	Management Group

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



From
Today 18:25

To
Today 18:25

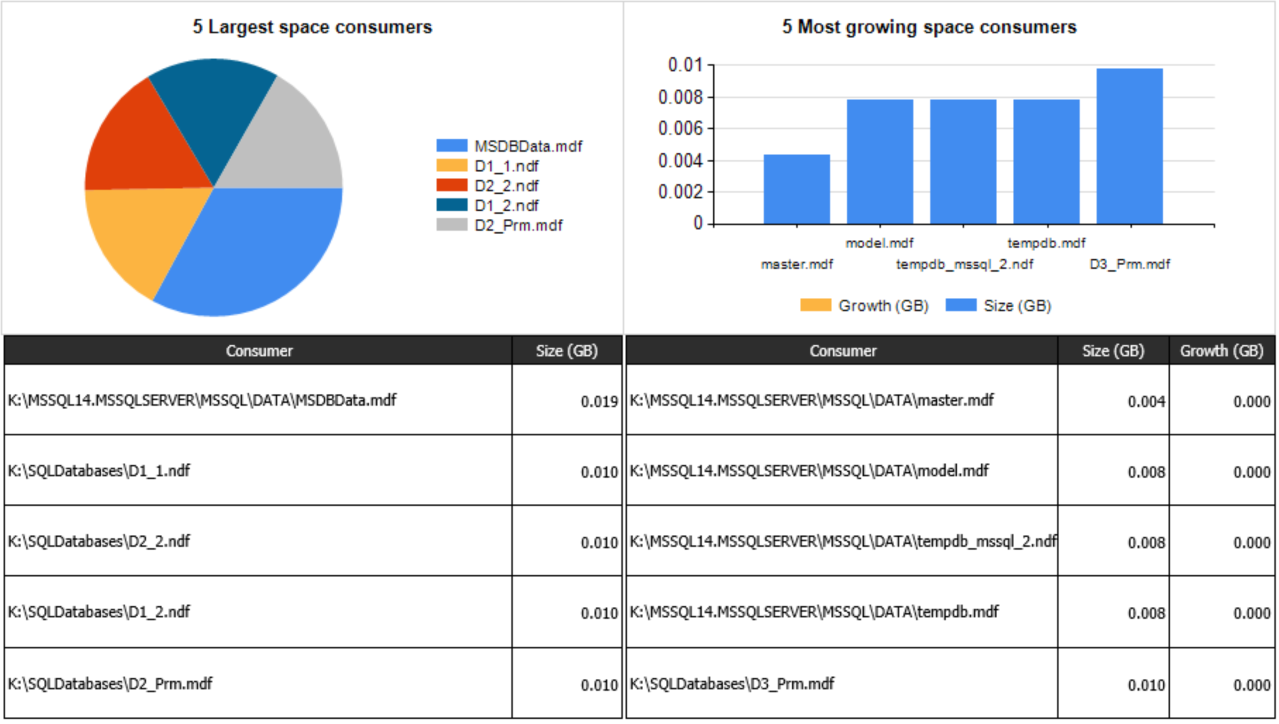
Time Zone
(UTC+03:00) Moscow, St. Petersburg, Volgograd

ForecastDays
1

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)

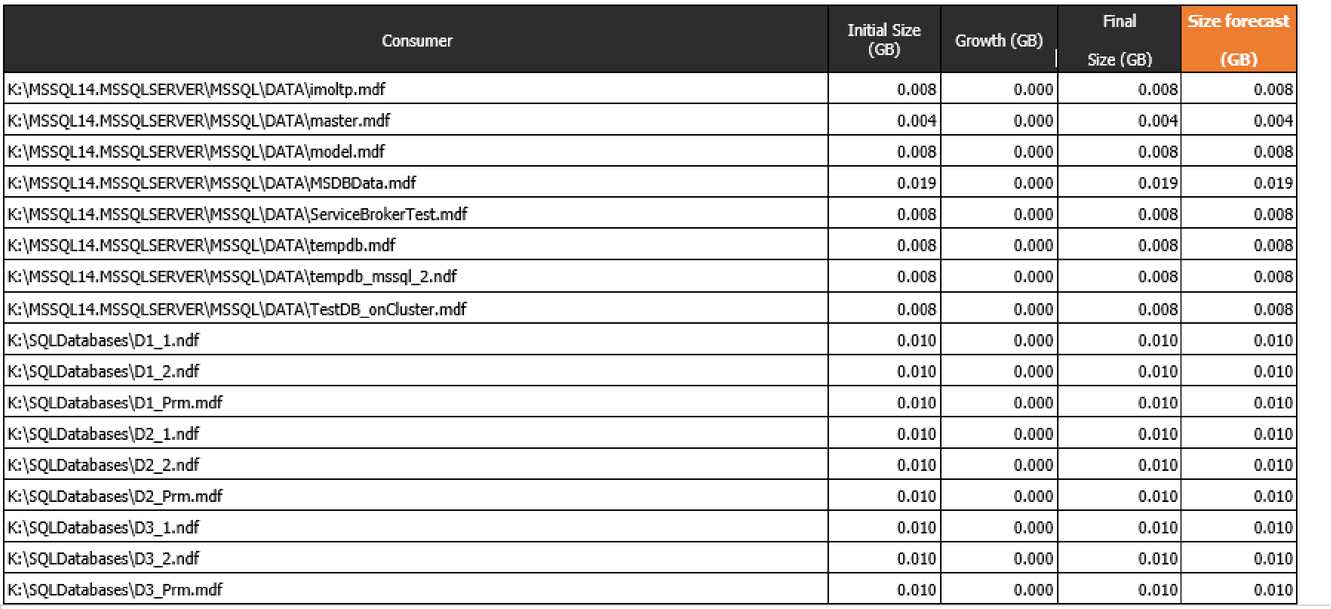
The report displays a separate chart for every selected object or a group of objects.



Consumer	Size (GB)
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\MSDBData.mdf	0.019
K:\SQLDatabases\D1_1.ndf	0.010
K:\SQLDatabases\D2_2.ndf	0.010
K:\SQLDatabases\D1_2.ndf	0.010
K:\SQLDatabases\D2_Prm.mdf	0.010

Consumer	Size (GB)	Growth (GB)
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\master.mdf	0.004	0.000
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\model.mdf	0.008	0.000
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\tempdb_mssql_2.ndf	0.008	0.000
K:\MSSQL14.MSSQLSERVER\MSSQL\DATA\tempdb.mdf	0.008	0.000
K:\SQLDatabases\D3_Prm.mdf	0.010	0.000

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_Prm.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_Prm.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_Prm.mdf	0.010	0.000	0.010	0.010

Note that this report works with Windows objects only.

Appendix: Known Issues and Troubleshooting

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

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

Resolution: Upon deletion of an operating system 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 a database is set to “True", performance rules targeting Database return empty values for this particular database.

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.

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, the RepeatCount property may contain an excess number of events.

Resolution: No resolution.

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.

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 a reboot of the Docker

Issue: Multiple errors occur after a 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".

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 the current implementation, these rules work with the English version only.

Resolution: No resolution.

Console tasks for Availability Group objects with names containing double-quote 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 an 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 the correct authentication type.

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

Issue: Configuration Service may frozen after Management Pack re-installation.

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.

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

To isolate the provider in its own host, run the script below in an elevated PowerShell session:

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

To revert the changes, run this script.

```
$a =
[WMI]'Root\Microsoft\SqlServer\ComputerManagement14:__Win32Provider.name="MSSQL_Ma
nagementProvider"'
$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 00000000003000.000000:000, 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.

“Custom user policy” discovery discovers system databases

Issue: The discovery may discover custom SQL Server policies for system databases (such as “master”, “msdb” etc.) and custom ones. In fact, a custom user policy can be performed only databases created by the user.

Resolution: No resolution.

SCOM issue: Sometimes SCOM console may show an exception in "Database Engines" state view if the selected instance is in the process of undiscovery

Issue: Sometimes SCOM console may show an "object reference not set" exception in "Database Engines" state view if the selected instance is in the process of undiscovery*.*

Resolution: No resolution.

Login fails when adding a new instance using the Add Monitoring Wizard

Issue: The following error may show up after you finish adding a new instance to the monitoring using the Add Monitoring Wizard: “An error occurred discovery: A connection was successfully established with the server, but then an error occurred during the login process.” Most likely, this error indicates that the “SQL Server MP Monitoring Pool” resource pool has not been discovered yet.

Resolution: Decrease intervals for both the "MSSQL: Generic Monitoring Pool Watcher Discovery" and the "Discover All Management Servers Pool Watcher" discoveries to force them to run right away, then restore the previous value.

“CPU Utilization” performance rule may show values greater than 100

Issue: Sometimes “CPU Utilization” performance rule may show values greater than 100. This occurs due to a known issue in the sys.dm_os_ring_buffers dynamic management view which is used by the rule to get the utilization value.

Resolution: No resolution.

A “Job Failed” error appears when adding a new SQL Server instance to monitoring with “Add Monitoring Wizard”

Issue: On the last step of the Wizard, when clicking “Create”, an error message appears that states “Job Failed.” This most likely indicates that you name the monitoring template with one of the following names: SystemCenter, Windows, System, SQLCorelib.

Resolution: Do not use any of the mentioned words as a monitoring template name.

SQL Server instances have not been discovered after importing the management pack and configuring it, and no alerts have been raised

Issue: This may indicate that you have bound a non-basic action account to the SQL Credentials run as profile.

Resolution: Configure the SQL Server MP Run As Profiles in appliance with Security Configuration.

Errors occur in workflow HKTableMemoryUsageAction related to Memory-Optimized Data: "Memory Used By Indexes (MB)", "Memory Used By Tables (MB)"

Issue: Such errors may indicate that you have performance degradation in the environments with a lot of Memory-Optimized databases:

```
Module:
Microsoft.SQLServer.Windows.Module.Monitoring.Performance.HKTableMemoryUsageAction
Version: x.x.x.x
Error(s) was(were) occurred:
Message:
---------- Exception: ----------
Exception Type: System.TimeoutException
Message: Module execution was terminated due to timeout after 300.000 seconds
Source: Microsoft.SQLServer.Module4.Helper
Stack Trace:
at Microsoft.SQLServer.Module.Helper.Base.ModuleBasePropertyHelper
1.<GetOutputDataAsync>d__13.MoveNext()
```

Resolution: Investigate performance degradation on affected servers and solve it if possible.

Service-status related monitors not working on SQL Server cluster instances whereas the SQL Server role is stopped

Issue: "SQL Server Windows Service", "SQL Full-text Filter Daemon Launcher Service" and "SQL Server Agent Service" monitors will be in a Healthy state on SQL Server cluster instances whereas the SQL Server role is stopped.

Resolution: Due to specific behavior of cluster nodes, monitoring is not performed for SQL Server cluster instances with the disabled SQL Server role.

The 'Primary replica' column shows different hosting machines for Availability Groups that are deployed on Windows and Linux-based systems under the same name and with the 'External' or 'None' cluster type

Issue: If an Availability Group is deployed on Windows and Linux-based systems under the same name and the cluster type of which is set to 'External' or 'None', the discovery results for such an Availability Group will be different each time and will be showing either of the hosting machines in the 'Primary replica' column, one after another. This is caused by non-uniqueness of IDs of such Availability Groups and forces Availability Group Discovery to pick a different hosting machine during each discovery interval and show this machine as a source.

Resolution: No resolution.