SQL diagnostic manager (SQLdm) provides an unprecedented level of diagnostic information on the health, performance, and status of SQL Server instances across your environment. You can view, diagnose, and report on critical performance statistics from a central point of control.

You can get a quick glimpse into the newest features, fixed issues, and known issues in this release of SQLdm by accessing the Release Notes.

- Learn about key new features in this release
- Review issues fixed by this release
- Review previous features and fixed issues
- See known issues
- See list of recommended Idera Solutions
SQL diagnostic manager (SQLdm) is a database management system (DBMS) solution designed exclusively for Microsoft SQL Server diagnosis and performance tuning. SQLdm lets you discover potential and existing problems, analyze the state of your SQL Server enterprise, and generate reports.
Upgrading SQLdm allows you to take advantage of the new features available in version 7.1.

**TIP**
To upgrade from SQLdm 4.x (or from earlier versions) to SQLdm 7.1, you must first upgrade to SQLdm 5.x. For more information, contact Idera Technical Support.

## Upgrade from SQLdm 5.x or higher to SQLdm 7.1 checklist

To successfully upgrade your Microsoft SQL Server environment to this build, complete the following procedures outlined in the following table.

<table>
<thead>
<tr>
<th>Follow these steps ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐ Backup your SQLdm Repository prior to upgrading.</td>
</tr>
<tr>
<td>☐ SQLdm 6.5 and later includes significant improvements to the SQLdm Repository. These improvements require that your SQLdm Query Monitor data be upgraded. Refer to Query Monitor Upgrade for more information.</td>
</tr>
<tr>
<td>☐ Choose the deployment strategy that best fits your environment and meets your monitoring needs.</td>
</tr>
<tr>
<td>☐ Ensure your environment meets the product requirements for SQLdm.</td>
</tr>
<tr>
<td>☐ Ensure that all SQLdm desktop clients are no longer running.</td>
</tr>
<tr>
<td>☐ Upgrade your environment to 7.1 on the computer where the SQLdm services are located.</td>
</tr>
<tr>
<td>☐ Test your new SQLdm deployment and setup.</td>
</tr>
<tr>
<td>☐ Upgrade your Idera Newsfeed and SQLdm Mobile installations or deploy a new Idera Newsfeed and SQLdm Mobile installation.</td>
</tr>
</tbody>
</table>
Use the following steps to upgrade your SQLdm installation in a clustered environment.

**TIP** SQLdm 7.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database for all installations.

### In a SQL Server 2003 Clustered Environment

**To upgrade SQLdm in a clustered environment:**

1. Logon with the Administrator account to the cluster node.
2. In Cluster Administrator, select the resource group that contains the SQLdm services and take the SQLdm Management and Collection services offline.

**Perform the following on each node, starting with the active node:**

1. Upgrade SQLdm components previously deployed to the node.
2. In Windows Services, stop the SQLdm Collection, SQLdm Management, and SQLdm Predictive Analytics services.
3. Set the SQLdm services to manually startup and then restart them.

**Once the steps above have been completed on each cluster node:**

1. In Cluster Administrator, bring the SQLdm Management Service online.
2. Use the Management Service Configuration wizard to force the services to re-register on the active node.

### In a SQL Server 2008 Clustered Environment

**To upgrade SQLdm in a clustered environment:**

1. Logon with the Administrator account to the cluster node.
2. In the Failover Cluster Management tool, select the resource group that contains the SQLdm services and take the SQLdm Management and Collection services offline.

**Perform the following on each node, starting with the active node:**

1. Upgrade SQLdm components previously deployed to the node.
2. In Windows Services, stop the SQLdm Collection, SQLdm Management, and SQLdm Predictive Analytics services.
3. Set the SQLdm services to manually startup and then restart them.

**Once the steps above have been completed on each cluster node:**

1. Bring the SQLdm Management Service online.
2. Use the Management Service Configuration wizard to force the services to re-register.
The Query Monitor upgrade begins one minute after the SQLdm management service starts. The upgrade is made in batches of 1,000 statements, which are moved from the old Query Monitor table to the new Query Monitor table via the SQLdm management service. This will continue for up to three hours and if the upgrade is not complete within three hours, the upgrade will continue from midnight each night until 3:00 am until it has finished.

**TIP** If you need to change the window of time the upgrade uses to complete the Query Monitor upgrade, please contact Idera Tech Support. Support can help you change the duration and the time when the upgrade occurs.
Perform a distributed upgrade

SQL diagnostic manager™

Use the following procedures to upgrade to a distributed SQLdm 7.x environment.

Install the SQLdm Repository

Ensure you install the SQLdm Repository database on a SQL Server instance that meets the product requirements. The Repository does not need to be in the same domain as the Console, but there must be a two-way trust between the target domains.

TIP SQLdm 7.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database for all installations.

To install the SQLdm Repository:
1. Log on with an administrator account to the computer on which you want to install the SQLdm Repository database.
2. Run Setup.exe in the root of the installation kit.
3. Click INSTALL on the Welcome window of the setup program.
4. Click INSTALL SQL diagnostic manager on the Install window.
5. Read the Welcome window, and then click Next.
6. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
7. Accept the default folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
8. Select the Custom Install setup type, and then click Next.
9. Select SQLdm Repository, and then click Next.
10. Specify which SQL Server instance should host the Repository database, specify a name for the Repository database, and then click Next.
11. Click Install to begin your installation.
12. Click Finish to exit the setup program.

Install the SQLdm Services

You can install the SQLdm services on the Repository computer (recommended) or a different computer.

To install the SQLdm services:
1. Log on with an administrator account to the computer on which you want to install the SQLdm services.
2. Run Setup.exe in the root of the installation kit.
3. Click INSTALL on the Welcome window of the setup program.
4. Click INSTALL SQL diagnostic manager on the Install window.
5. Read the Welcome window, and then click Next.
6. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
7. Accept the default folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
8. Select the Custom Install setup type, and then click Next.
9. Select SQLdm Services, and then click Next.
10. Specify which SQL Server instance is hosting the Repository database, verify that the displayed database name is correct, and then click Next.
11. Specify the account credentials that the SQLdm services should use to connect to the SQLdm Repository and the monitored SQL Server instances, and then click Next. Ensure this account belongs to the sysadmin role on the Repository database as well as each instance you plan to monitor.

12. Click Install to begin your installation.

13. Click Finish to exit the setup program.

Install the SQLdm Console

You can install the SQLdm Console on multiple computers or workstations. The Console does not need to be in the same domain as the Repository, but there must be a two-way trust between the target domains.

To install the SQLdm Console:

1. Log on with an administrator account to the computer on which you want to install the SQLdm Console.
2. Run Setup.exe in the root of the installation kit.
3. Click INSTALL on the Welcome window of the setup program.
4. Click INSTALL SQL diagnostic manager on the Install window.
5. Read the Welcome window, and then click Next.
6. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
7. Accept the default folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
8. Select the Custom Install setup type, and then click Next.
9. Select SQLdm Console, and then click Next.
10. Click Install to begin your installation.
11. Click Finish to exit the setup program.
Before you upgrade a SQLdm 4.x environment to SQLdm version 5.x or higher, ensure you take note of your alert configurations and any notification or destination rules. After the upgrade completes, you should recreate these configurations using the new SQLdm Console.

You can capture your alert configuration and notification rules by taking a screenshot of the Alert Thresholds window for each monitored SQL Server instance and any Alert Destinations configured for these instances.

With SQLdm 5.x and higher, you can use a single alert response rule to monitor multiple SQL Server instances. As you review your current alert configurations, consider consolidating some alerts. For example, you can monitor heavily utilized instances using a single alert that tracks SQL Server CPU usage.
Testing your deployment

To test your upgrade, you can immediately begin collecting and alerting on performance data. Verify that you are able to monitor the target SQL Server instances and that the data you are retrieving is valid.

SQL diagnostic manager is a powerful performance monitoring and diagnostics solution that proactively alerts administrators to health, performance and availability problems within the SQL Server environment.

To get started, see the following topics for more information:

- Upgrade to this build of SQL diagnostic manager
- Configure SQL diagnostic manager
- Migrate your SQLdm installation
SQL diagnostic manager (SQLdm) provides an unprecedented level of diagnostic information on the health, performance, and status of SQL Server instances across your environment. You can view, diagnose, and report on critical performance statistics from a central point of control.

Using a unique agentless architecture, SQLdm gathers diagnostic information in real time, keeping database administrators and managers informed by providing customized alerts. Easy to install and use, SQLdm is an indispensable SQL Server management tool that benefits both experienced and junior database administrators.

What is SQLdm Mobile?

SQL diagnostic manager (SQL dm) Mobile is a Web application that displays real-time SQL Server performance dashboards on a variety of mobile devices, including iPhone, iPad, Android and Blackberry. SQLdm Mobile includes the Idera Newsfeed technology. The Idera Newsfeed is a revolutionary new way for DBAs and managers to collaborate, share knowledge, and keep close tabs on their most critical SQL Server issues.
How does SQLdm help me?

Whether you need immediate diagnostic information or want to gather data for long-term capacity planning, you can use SQLdm to help:

- Increase SQL Server availability
- Provide a comprehensive view of current performance
- Provide continuous, unattended monitoring
- Store data in the centralized SQLdm Repository database
- Improve the productivity of both skilled and novice database administrators

With clear, easy-to-understand charts and reports, you can track standard SQL Server performance metrics, trend historical performance data over time, and proactively identify potential performance problems.
SQLdm uses a graphically enhanced user interface to display the current state of your SQL Server instances. A mixture of graphs, charts, and color coding makes problem identification quick and easy, allowing you to resolve issues before they reach critical levels.
The Navigation pane displays on the left-side of the SQLdm Console. The options available on the Navigation pane change depending on the currently selected feature. You can hide the Navigation pane by selecting View > Navigation Pane > Off.

The following features are available in the Navigation pane:

**Servers**
- Contains a My Views section that helps you organize your SQL Server instances by status and includes any custom views you have set up. The Servers section lists the SQL Server instances contained in the currently selected view. To view all registered instances, select All Servers from the My Views list.

**Alerts**
- Contains filter options for organizing your alerts and a link for configuring alert notifications.

**To Do**
- Contains filter options for organizing and managing your To Do list. Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.

**Reports**
- Allows you to select the report you want to generate.

**Administration**
- Allows you to control access to SQLdm and add custom counters for monitoring.
The Refresh and Pause buttons on the SQLdm Toolbar allow you to control how the information on the particular view is refreshed. You can edit the scheduled refresh of your views for more current information.

What does the Refresh button do?
The Refresh button tells SQLdm to immediately collect data and refresh the current Console view. Depending on the view, this could take several seconds to complete. Refresh status is displayed in the bottom-right corner of the SQLdm interface.

What does the Pause button do?
The Pause button tells SQLdm to stop refreshing the console. If a refresh is currently in progress, the pause will occur after the refresh has completed. To continue refreshing, click Pause again. Pausing the refresh allows you to maintain the currently monitored state so that you can see what is happening on your SQL Server instance without having incoming information overwrite what you are investigating and it also gives you time to export or print statistics that are currently displayed before the next data set is retrieved.
SQLdm provides filtering options on most views. These can range from ribbon filters that allow you to select or deselect items to show on the view to a Filter Settings window that gives you a variety of options to customize the way the view is displayed.

**How do I use the Group By feature to organize my view?**

The Group By feature allows you to group the information shown in your views. To group your information by a column, click the Group By box and drag a column heading into the Group By field.

**TIP** You can group your tables by multiple columns. To group by multiple columns, click the Group By box, drag the first column heading (first order of grouping) into the Group By field, and then drag the second column heading into the Group By field (second order of grouping).

**How can I use the Filter Settings window to customize my view?**

Every Filter Settings window in SQLdm provides specific options for the current view. These options can include true or false fields that allow you to decide whether to show a particular item, filtering fields where you can enter an item name or even use a wildcard (%), dates and time ranges, and interval periods.

Each field in the Filter Settings window includes a description that displays at the bottom of the window when the field is selected.
The Details view on the Overview, Sessions, Queries, Resources, Databases, and Services tab contain Charts that are populated by selecting the appropriate data items collected. Once populated, you can manipulate the type of chart to display and export the chart or copy it to the clipboard to paste into other programs.

Charts are populated from the point at which you navigate to the window with the chart. This allows you to view real-time (dependent on your refresh interval) charts for the data point selected.

**TIP** Click the Maximize button on the upper-right corner of a chart to display a large version of the chart. The Restore button will return the chart to the default view.

**How can I determine how much data to show in charts and how often it is refreshed?**

Charts are refreshed according to the time entered in the Server View Refresh field on the Console Options window (Tools > Console Options).

The amount of time displayed in your charts is set according to the time entered in the Show data for the last field on the Console Options window (Tools > Console Options). In addition, the time entered in the Keep data for the last field is the amount of historical data that you can possibly scroll back to in your charts.

**How can I customize my charts?**

SQLdm contains charts that are populated by selecting the appropriate data items collected. Once populated, you can manipulate the type of chart to display and export the chart or copy it to the clipboard to paste into other programs.

You can personalize charts by right-clicking anywhere inside them and selecting Toolbar from the menu. The chart toolbar contains the following options:

- **Palette**
  - The Palette menu allows you to change color scheme of your chart.

- **Gallery**
  - The Gallery menu allows you to change the type of chart that is used to represent your data.

- **Data Grid**
  - The Data Grid toggle places a table below your chart that displays the data included in your chart.

- **Copy to Clipboard**
  - The Copy to Clipboard menu allows you to copy the chart to your Windows clipboard in various formats.
Monitor multiple instances

My Views allows you to group servers to provide a high-level, global view of multiple monitored servers. This view includes basic statistics about your registered SQL Server instances and quick links to more detailed information. A monitor displays statistics for each SQL Server instance included in the group.

How often is the data shown in group list refreshed?
The data shown in group list is updated according to the interval entered in the Status and Alerts refresh field on the Console Options window, accessed by clicking Tools > Console Options. This information is collected from the last data point stored in the SQLdm Repository.

What information can I get from the group list?
The group list, by default, displays basic statistics for each of the SQL Server instances that SQLdm is monitoring. The information includes the following:

- **Response Time**: Refers to the time it takes for a query to go from the SQLdm Console to the SQL Server instance and back.
- **User Sessions**: Refers to the total number of SQL Server user sessions.
- **SQL CPU Usage**: Refers to average percentage of SQL Server processor usage on the computer hosting the SQL Server instance.
- **Memory Usage**: Refers to amount of memory in use by the SQL Server instance monitored.
- **SQL Disk I/O**: Refers to the number of physical reads and physical writes made by the SQL Server instance between refreshes. *If you see a high percentage of physical reads, you can improve performance by allocating more memory to the SQL data cache.*

How do I customize the monitors in my group list?
The monitors displayed in group lists can be customized so that you display the key statistics or charts that are most important to you.

To customize a monitor:
1. Right-click a SQL Server instance in the view.
2. Select Thumbnail and select the information type you want to display on your monitor.

Instead of listing statistics for the server, the individual chart of the data item you pick is shown. To revert back, select Summary from the Thumbnail list.

What kind of information is provided on the Server Details window?
The Server Details window is accessed by clicking the Details icon in the top-corner of the Mini-Monitors window. This window lists all the data collected at the last refresh for all your monitored SQL Server instances. This window can assist you with comparing the status of a specific metric on all your SQL Server instances.
If any of the metrics are missing from this view, you may need to enable OLE automation.

What are the default views?

**All Servers**
This view contains all monitored SQL Server instances.

**Critical**
This view contains any monitored SQL Server instance that has any associated Critical Alerts.

**Warning**
This view contains any monitored SQL Server instance that has any associated Warning Alerts.

**OK**
This view contains all monitored SQL Server instance that have no associated Warning or Critical alerts.

**Maintenance Mode**
This view contains any monitored SQL Server instance that has been placed into Maintenance Mode.

How do I add a new view?
You may want to add a custom view to the My Views list to organize your SQL Server instances into specific groups for monitoring purposes. This is especially important if you are monitoring a large number of SQL Server instances and want information for a specific group of SQL Server instances. If you create views before adding your SQL Server instances to SQLdm for monitoring, you can have these new instances added to your views by selecting the new view you create and then adding the SQL Server instances.

To add a new view:
1. Select Add > View.
2. Enter a name for the view.
3. Select the SQL Server instances you want to monitor in this view.
4. When finished, click Add to create your new view.
Organize instances into groups

By default, the My Views list places your monitored SQL Server instances into five status-based views that allow you to quickly focus on a group of specific SQL Server instances.

What are the default views?

All Servers
This view contains all monitored SQL Server instances.

Critical
This view contains all monitored SQL Server instances that have associated critical alerts.

Warning
This view contains all monitored SQL Server instances that have associated warning alerts.

OK
This view contains all monitored SQL Server instances that have no associated warning or critical alerts.

Maintenance Mode
This view contains all monitored SQL Server instances that are in maintenance mode.

How do I add a new view?
You may want to add a custom view to the My Views list to organize your SQL Server instances into specific groups for monitoring purposes. This is especially important if you are monitoring a large number of SQL Server instances and want information for a specific group of SQL Server instances.

To add a new view:
1. Select Add > View.
2. Type a name for the view.
3. Select the SQL Server instances you want to monitor in this view.
4. When finished, click Add to create your new view.
The SQLdm Today page provides a current overview of all the monitored SQL Server instances in your environment and offers shortcuts to:

- Your 5 most critical servers
- The 5 servers you recently monitored
- All active alerts in your SQL Server environment
- Most recent server status updates and coworker posts from the Idera Newsfeed

**How do I use SQLdm Today to view the general health of my environment?**

SQL diagnostic manager gives you a quick glance at the overall health of your environment in the Status section of the SQLdm Today page. The main categories of data shown in this view include the following areas:

**Status Summary**

The Status Summary section displays the number and type of active alerts for all of your SQL Server instances. SQLdm displays a large icon that shows you at a glance the current highest level of severity of your servers accompanied by the number of your servers in that state.

**Most Critical Servers**

SQLdm provides a list of up to five of your most critical servers so you can quickly access that SQL Server instance and review the issue.

**Recent Servers**

The SQLdm Today page provides a list of your five recently-accessed SQL Server instances, with the most recent visit appearing at the top of the list.

**How do I find information about getting the most from SQLdm?**

The Get Started tab of the SQLdm Today page provides links for you to access content in the SQLdiagnostic manager Trial Center including video tours that cover important features and SQLdm navigation. You can also find Webcasts, white papers, and the SQL Server Performance Blog for additional information that you may find useful when using SQLdm.

**Are these all my recent active alerts?**

Yes, SQLdm displays a list of your most recent active alerts on the SQLdm Today page Active Alerts tab. While this view displays alerts for all your monitored SQL Server instances, you can view additional alert information for a specific server instance by using the right-click context menu. This additional information includes a real-time view and a shortcut to configure your alerts.

**Can I remove an alert from the Active Alerts tab?**

If you are aware of an alert but no longer want to see it in your list or have it impact the associated server, you can "snooze" or temporarily postpone the alert from appearing for a set number of minutes. When you snooze an alert, SQLdm clears the alert which no longer impacts the state of the monitored server until the snooze period expires. To snooze an alert, right-click the specific alert, and then select Snooze Alert. Using 10-minute increments, select the appropriate number of minutes you want to alert to snooze, and then click OK.

**How can I keep track of my SQL Server issues?**
The Idera Newsfeed is a new way for you to share knowledge with other DBAs and managers regarding critical and current SQL Server issues. To learn how to set up and use the Idera Newsfeed, see the Idera Newsfeed Help.
This documentation set includes a comprehensive online Help system as well as additional resources that support you as you install and use the product. You can also search Idera Solutions, available at the Idera customer service portal (www.idera.com/support).

**How do I print the Help topics?**

Select the top level node in the table of contents and click the **Print** button.
At Idera, we deliver a new generation of tools for managing, administering, and securing your Microsoft Windows Servers, including SQL Server, SharePoint, PowerShell and Microsoft Dynamics. We employ numerous industry experts worldwide who are devoted to bringing proven solutions to you, the administrator. Idera provides solutions that help you ensure server performance and availability and reduce administrative overhead and expense. Our award-winning products install in minutes, configure in hours and deploy worldwide in days. Idera is a Microsoft Gold Certified Partner headquartered in Houston, Texas, with offices in London, UK, Melbourne, Australia, and Sao Paulo, Brazil.
Our tools are engineered to scale - from managing a single server to enterprise deployments with thousands of servers. Idera products combine ease of use with a design that installs in minutes, configure in hours, and deploy worldwide in days. To learn more about Idera products, visit http://www.idera.com/Content/Products.
Contacting Idera

Please contact us with your questions and comments. We look forward to hearing from you. For support around the world, please contact us or your local partner. For a complete list of our partners, please see our the idera website.

| Sales          | 713.523.4433               |
|               | 1.877.GO.IDERA (464.3372)  |
|               | (only in the United States and Canada) |
| Sales Email   | sales@idera.com            |
| Support       | 713.533.5144               |
|               | 1.877.GO.IDERA (464.3372)  |
|               | (only in the United States and Canada) |
|               | www.idera.com/support      |
| Website       | www.idera.com              |
Idera – a division of BBS Technologies, Inc. IntelliCompress, DTx, Idera Newsfeed, Newsfeed, Point admin toolset, PowerShellPlus, SharePoint admin toolset, SharePoint backup, SharePoint performance monitor, SQL admin toolset, SQL change manager, SQLcheck, SQLcompliance, SQL compliance manager, SQLconfig, SQL defrag manager, SQL diagnostic manager, SQL diagnostic manager mobile, SQLdm, SQLdm Mobile, SQL doctor, SQL job manager, SQLpermissions, SQL mobile manager, SQLsafe, SQLsafe Freeware Edition, SQLscaler, SQLschedule, SQLsecure, SQLsmarts, SQLstats, SQLtool, Idera, BBS Technologies and the Idera logo are trademarks or registered trademarks of BBS Technologies, Inc., or its subsidiaries in the United States and other jurisdictions. All other company and product names may be trademarks or registered trademarks of their respective companies. © 2012 BBS Technologies, Inc., all rights reserved.

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT, BBS TECHNOLOGIES, INC., PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU. YOU ARE ENCOURAGED TO READ THE LICENSE AGREEMENT BEFORE INSTALLING OR USING THIS DOCUMENTATION OR SOFTWARE.

Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. BBS Technologies, Inc., may make improvements in or changes to the software described in this document at any time.

© 2003-2012 BBS Technologies, Inc., all rights reserved.

U.S. Government Restricted Rights: The software and the documentation are commercial computer software and documentation developed at private expense. Use, duplication, or disclosure by the Government is subject to the terms of the BBS Technologies, Inc., standard commercial license for the software, and where applicable, the restrictions set forth in the Rights in Technical Data and Computer Software clauses and any successor rules or regulations.
SQLdm and SQLdm Mobile can be installed in any sized environment.

Get ready for your SQLdm deployment

- Check the supported installation scenarios
- Learn about the components and architecture
- Review system requirements
- View the installation instructions

Get ready for your SQLdm Mobile deployment

- Check the supported installation scenarios
- Learn about the components and architecture
- Review system requirements
- View the installation instructions
Supported installation scenarios

You can install and deploy SQLdm and SQLdm Mobile to meet your unique SQL Server environment needs.

The following installation scenario descriptions assume you are planning to deploy both SQLdm and SQLdm Mobile in your environment.

**TIP** In larger environments that contain more than one SQLdm installation, consider dedicating a SQLdm Mobile deployment per each SQLdm application.

---

**Typical environment**

This configuration includes the following installations:

- SQLdm Console on your workstation
- SQLdm Mobile Web site on a Web Server for smart phone access
- SQLdm Mobile & Newsfeed Service on the same computer that hosts the SQLdm Services
- SQLdm Mobile Repository and SQLdm Repository on the same SQL Server instance

---

**Clustered environment**

You can install and configure SQLdm to monitor virtual SQL Servers. A virtual SQL Server is a SQL Server running on a Microsoft failover cluster managed by Microsoft Cluster Services.

This configuration deploys all SQLdm Services to your virtual instances. A service-only deployment to a virtual instance includes the following installations:

- SQLdm Console on your workstation (not located in the cluster)
- SQLdm Mobile Web site on a non-clustered Web Server for smart phone access
- SQLdm Mobile Repository and SQLdm Repository on the same SQL Server instance
- SQLdm Mobile & Newsfeed Service and the SQLdm Services on each cluster node hosting the virtual SQL Server you want to monitor

---

**Web farm environment**

You can install SQLdm Mobile in a Web farm, ensuring this site remains accessible during a failover situation.

This configuration includes the following installations:

- SQLdm Mobile on a Web Server in the farm
- Typical SQLdm installation
- SQLdm Mobile & Newsfeed Service on the same computer that hosts the SQLdm Services
- SQLdm Mobile Repository and SQLdm Repository on the same SQL Server instance

---

**Non-trusted environment**

You can install and configure SQLdm and SQLdm Mobile to monitor SQL Server instances running in non-trusted domains or workgroups.

This configuration includes the following installations:

- SQLdm Console on your workstation in a trusted or non-trusted domain
- SQLdm Mobile Web site on a Web Server for smart phone access
- SQLdm Mobile & Newsfeed Service on the same computer that hosts the SQLdm Services, in a trusted or non-trusted domain
- SQLdm Mobile Repository and SQLdm Repository on the same SQL Server instance, in a trusted or non-trusted domain
Choose your deployment strategy

Before you begin installing SQLdm, it is important to consider your deployment choices. The following deployment strategies are recommended.

**TIP** You can deploy SQLdm Mobile after you have installed SQLdm. For more information, review the SQLdm Mobile architecture and components, as well as requirements.

**Trial Deployment**

A trial deployment installs all SQLdm components on the same computer. A trial deployment can monitor SQL Server instances anywhere on your network.

**Production Deployment**

A production deployment installs each SQLdm component on a separate, dedicated computer. This deployment ensures stable performance and availability as your environment grows and your monitoring needs change.
Deploy SQLdm only

Use the following links to prepare for your SQLdm deployment:

- Check the supported installation scenarios
- Learn about the components and architecture
- Review system requirements
- View the installation instructions
Product components and architecture

SQLdm consists of a light, unobtrusive architecture that easily runs in your SQL Server environment with minimal configuration. All SQLdm components run outside and separate from SQL Server processes.

SQLdm Console

Use the SQLdm Console to:

- View real-time status
- Configure alert notifications on specific metric thresholds at the server and database levels
- View historical reports and manage your To Do list (Note that beginning with SQLdm 7.0, the To Do feature is available only to users who activated To Do in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.)
- Perform administrative functions

The SQLdm Console retrieves historical information directly from the SQLdm Repository. All real-time requests use the SQLdm services to poll the monitored SQL Server.

SQLdm Services

SQLdm has three centralized services, the Management Service, the Collection Service, and the Predictive Service. These three services reside on the same computer.

The Management Service performs the following primary functions:

- Provides real-time data to the SQLdm Console
- Receives historical data from the Collection Service for storage in the Repository
- Raises alerts and sends alert notifications

The Collection Service performs on-demand and scheduled collection from the monitored SQL Servers.

The Predictive Service is used for Alert Forecasting and performs the following primary functions:

- Calculates the Alert Forecast every hour
- Builds a forecasting model once a day

SQLdm Repository

The SQLdm Repository is a centralized SQL Server database that stores collected metrics on a scheduled basis, historical data, and alerts information. The SQLdm Repository also stores configuration information, such as the credentials used to monitor a registered SQL Server instance.

TIP: SQLdm 7.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database for all installations.

Authentication in SQLdm

SQLdm uses the same types of authentication available in the SQL Server security model. When specifying account credentials for the SQLdm services, you can use Windows Authentication or SQL Server Authentication.

When considering which authentication to use, keep in mind that SQL Server Authentication is required when no domain trust exists between the SQLdm Services computer and the computers hosting the monitored SQL Server instances. For example, if the monitored SQL Server instances are located in an untrusted domain or behind a firewall, you must use SQL Server authentication.
Authentication to successfully deploy SQLdm. In this case, you must use the sa account or a SQL Server login that has System Administrator permissions.

For more information about these authentication types and the SQL Server security model, see the Microsoft SQL Server Books Online.
Product requirements

There are three types of SQLdm installation (Trial, General, and Production). Refer to the following list of requirements for each type of installation.

**TIP** SQLdm 7.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database.

**TIP** Tempdb monitoring is supported for SQL Server 2005 and above users only.

### Trial Requirements

Consider the following requirements when installing all SQLdm components on one computer.

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Standard or Enterprise Edition</td>
<td>SQL Server 2005 SP1 or later</td>
</tr>
<tr>
<td></td>
<td>SQL Server 2008 or later</td>
</tr>
<tr>
<td></td>
<td>SQL Server 2008 R2</td>
</tr>
<tr>
<td>Hard Disk Space</td>
<td>200 MB for all SQLdm components</td>
</tr>
<tr>
<td>Windows Permissions</td>
<td>Your Windows Logon account has local Administrator permissions</td>
</tr>
<tr>
<td>SQL Server Privileges</td>
<td>Either a Windows user account or SQL Server login that has System Administrator privileges on then monitored SQL Server instances</td>
</tr>
</tbody>
</table>

### General Requirements

The following requirements apply for all SQLdm components.

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2.0 GHz or higher</td>
</tr>
<tr>
<td>Memory</td>
<td>1 GB</td>
</tr>
<tr>
<td>32-bit Operating System</td>
<td>Windows Server 2003 SP2 or later</td>
</tr>
<tr>
<td></td>
<td>Windows XP SP2 or later</td>
</tr>
<tr>
<td></td>
<td>Windows Vista Business and Enterprise</td>
</tr>
<tr>
<td></td>
<td>Windows 7</td>
</tr>
<tr>
<td></td>
<td>Windows Server 2008 SP1 or later</td>
</tr>
<tr>
<td></td>
<td>.Net 2.0 and MDAC 2.8</td>
</tr>
<tr>
<td>64-bit Operating System</td>
<td>Windows Server 2003 SP1 or later</td>
</tr>
<tr>
<td></td>
<td>Windows XP Professional X64</td>
</tr>
<tr>
<td></td>
<td>Windows 7</td>
</tr>
<tr>
<td></td>
<td>Windows Server 2008 or later</td>
</tr>
<tr>
<td></td>
<td>Windows Server 2008 R2</td>
</tr>
<tr>
<td></td>
<td>Windows Vista Business and Enterprise</td>
</tr>
<tr>
<td></td>
<td>.Net 2.0 and MDAC 2.8</td>
</tr>
<tr>
<td>Port Access</td>
<td>5166, 5167</td>
</tr>
<tr>
<td>Client Browser</td>
<td>Internet Explorer 7.0 or later</td>
</tr>
</tbody>
</table>
**Production Requirements**

Consider the following requirements when installing SQLdm in an environment where the components are on separate computers.

**SQLdm Console**

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Space</td>
<td>40 MB</td>
</tr>
<tr>
<td>SQL Server Permissions</td>
<td>Your Windows logon account has Read Access to the SQLdm Repository. You can specify a SQL Server login instead.</td>
</tr>
</tbody>
</table>

**SQLdm Services**

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Space</td>
<td>50 MB</td>
</tr>
<tr>
<td>Collection Service Permissions</td>
<td>System Administrator privileges on the monitored SQL Server instances</td>
</tr>
<tr>
<td>Management Service Permissions</td>
<td>Read and Write privileges on the SQLdm Repository database</td>
</tr>
</tbody>
</table>

**SQLdm Repository**

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Space</td>
<td>1GB to 3GB for one year of data per monitored SQL Server instance</td>
</tr>
</tbody>
</table>
| SQL Server Standard or Enterprise Edition | • SQL Server 2005 SP1 or later  
• SQL Server 2008 or later  
• SQL Server 2008 R2 |

**Monitored SQL Server Instances**

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
</table>
| SQL Server Standard or Enterprise Edition | • SQL Server 2000 SP4 or later  
• SQL Server 2005 SP1 or later  
• SQL Server 2008 or later  
• SQL Server 2008 R2 |
How to install SQLdm

This procedure guides you through the installation of SQL diagnostic manager.

**Start your install**

You can install SQLdm on any computer that meets or exceeds the product requirements.

**To start installing SQLdm:**

1. Log on with an administrator account to the computer on which you want to install SQLdm.
2. Run `Setup.exe` in the root of the installation kit.
3. Click **Install** on the **Welcome** window, and then click **Install SQL diagnostic manager** on the **Install** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Review the license agreement. To accept this license agreement, click **I accept the terms of this license agreement** and then click **Next**.

**Where do you want to install SQLdm and who will use the product on this computer?**

You can use the default install location or specify a different path. For your first install, we recommend using the default location. By default, the Management Service is also installed in the same location.

Choose whether you want any user or only the current user to access this application, and then click **Next**.

**Begin configuring your install**

For your first install, we recommend using the Typical setup type. This type ensures you install and configure all required SQLdm components so you can immediately begin monitoring your SQL Server environment.

By default, the setup program installs the Management Service on the local computer where you are running the setup program. When you install the SQLdm Console, such as in a Console-only installation, the Management Service is also installed on the same computer.

**Which SQL Server instance do you want to use for the Repository database?**

The SQLdm Repository is where the collected data, alert information, and alert configuration settings are stored.

**TIP** SQLdm 7.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database for all installations.

To specify the SQL Server instance to install the SQLdm Repository on:

1. Click **Browse** to select the SQL Server instance you want to use. The SQL Server window lists SQL Server instances provided by your domain controller. **If you do not see your SQL Server instance in the list**, type the host name and instance in the space provided.
2. **If your SQL Server environment uses SQL Server Authentication**, select the checkbox and click **Change** to enter SQL Server credentials. By default, SQLdm uses your Windows credentials when connecting to the Repository.
3. Click **Next**.

**Which Windows user account do you want to use for the service account?**

SQLdm uses the service account to communicate between the monitored SQL Server instance and the SQLdm Repository.
To specify a service account:
1. Type appropriate credentials in the provided fields.
2. Click Next. SQLdm will verify the connection before continuing.

Complete the install
Click Install to indicate that you are ready to complete your install with the configurations you specified.

Decide whether to deploy SQLdm Mobile
SQLdm Mobile includes the Idera Newsfeed as well as several dashboard views that define your server health and expose critical alerts. The SQLdm Mobile Web application can be accessed from most smart phones and mobile devices.

The Idera Newsfeed is a revolutionary new way for DBAs and managers to collaborate, share knowledge, and keep close tabs on your most critical SQL Server issues. With the Idera Newsfeed, you can quickly share information to stay informed, be more productive and shorten the time to problem resolution.

For more information, review the SQLdm Mobile architecture and components, as well as its requirements.
Deploy SQLdm Mobile

Use the following links to prepare for your SQLdm Mobile deployment:

- Check the supported installation scenarios
- Learn about the components and architecture
- Review system requirements
- View the installation instructions

What is SQLdm Mobile?

SQLdm Mobile is a Web-based application that includes the Idera Newsfeed technology as well as several dashboard views that define your server health and expose critical alerts. The SQLdm Mobile Web site can be accessed from most smartphones and mobile devices.
SQLdm Mobile components and architecture

SQLdm Mobile is an easy-to-use, Web-based application that seamlessly integrates into your existing diagnostic and monitoring workflows. The SQLdm Mobile Web site includes many metrics and alerts dashboards that let you quickly ascertain performance issues in your environment.

SQLdm Mobile also includes the Idera Newsfeed technology. The Newsfeed provides an easy-to-use, social-media-based collaboration tool you can use to solve issues faster while keeping your team appraised of changing server statuses.

Whether you are tied to your desktop, on the go, or managing remote servers, these features can fit your unique monitoring needs.

The SQLdm Mobile architecture easily runs in your SQL Server environment with minimal configuration. All components run outside and separate from SQL Server processes, and do not add to or modify any of your native SQL Server files or services.

Components

Idera Newsfeed for SQLdm

The Idera Newsfeed is a user interface that seamlessly integrates with the SQLdm Console, letting you monitor your instances and collaborate with coworkers at the same time, in real time.

Idera Newsfeed Platform

The Idera Newsfeed Platform consists of the SQLdm Mobile Repository database and the SQLdm Mobile & Newsfeed Service.

SQLdm Mobile Repository database

The SQLdm Mobile Repository (Mobile Repository) is a central database that stores your configuration settings, comments and other activity on stories, as well as notifications. Consider deploying the Mobile Repository to the same computer that hosts your SQLdm Repository.

SQLdm Mobile & Newsfeed Service

The SQLdm Mobile & Newsfeed Service facilitates communications between the SQLdm Mobile Repository and your SQLdm application. By default, this service is installed on the same computer where you run the SQLdm Mobile & Newsfeed setup program. Consider SQLdm Mobile & Newsfeed Service to the same computer that hosts your SQLdm Services, such as the SQLdm Management Service.

Action Provider

The Idera Newsfeed Action Provider converts SQLdm alerts about a server’s status and performance metrics into notification emails that SQLdm Mobile & Newsfeed Service can send to the appropriate people. The setup program installs the action provider on the computer hosting your SQLdm Repository.

Authentication Provider

The Idera Newsfeed Authentication Provider is the SQLdm application that the Idera Newsfeed Platform uses to authenticate a user when he or she signs up or logs on. The Idera Newsfeed Platform leverages the application security settings as defined in SQLdm, determining which monitored SQL Server instances a given user can access.

SQLdm Mobile Web application (Web site)

The SQLdm Mobile Web site provides access to the same monitoring and collaboration features of the Idera Newsfeed technology via a mobile device, such as a smart phone or iPad. The SQLdm Mobile Web site also includes SQLdm dashboard screens so you can quickly check a server’s performance when you are away from your desk.
You can easily and quickly install SQLdm Mobile on any computer that meets or exceeds the following requirements.

The SQLdm Mobile installation includes the Idera Newsfeed technology and these related components:

- SQLdm Mobile Web application
- SQLdm Mobile & Newsfeed Service
- SQLdm Mobile Repository

For more information, review all the SQLdm Mobile components and architecture.

### General requirements for all components

All SQLdm Mobile components require Microsoft .NET Framework version 4.0 or later.

### Permission requirements

The SQLdm Mobile & Newsfeed Service account requires the following permissions. By default, setup program will assign the Local System account to the SQLdm Mobile & Newsfeed Service. During install, you can input credentials for a Windows user account or SQL Server login.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Why It’s Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log on as Service</td>
<td>Allows the SQLdm Mobile &amp; Newsfeed Service account to run as a service.</td>
</tr>
<tr>
<td>Read and write privileges on the SQLdm Mobile Repository database</td>
<td>Allows the SQLdm Mobile &amp; Newsfeed Service to receive and maintain performance and configuration data in the SQLdm Mobile Repository.</td>
</tr>
</tbody>
</table>

### Port requirements

By default, the SQLdm Mobile components use the following ports.

<table>
<thead>
<tr>
<th>Port Number</th>
<th>Where</th>
<th>Which Component Uses It</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Mail Server</td>
<td>SQLdm Mobile &amp; Newsfeed Service</td>
<td>Allows the SQLdm Mobile &amp; Newsfeed Service to send notification emails.</td>
</tr>
<tr>
<td>80</td>
<td>Web Server hosting the SQLdm Mobile Web site</td>
<td>SQLdm Mobile Web application</td>
<td>Allows access to the SQLdm Mobile screens via a mobile device inside or outside your corporate firewall. By default, IIS uses port 80 for the local virtual directory, so check your Web Server settings to pick an appropriate port for SQLdm Mobile. You can specify a different port during install or later when you need to change this configuration.</td>
</tr>
<tr>
<td>5166 5167</td>
<td>Computer running the SQLdm services</td>
<td>SQLdm services</td>
<td>Allow communications between SQLdm and the Idera Newsfeed Platform for signup and login authentication.</td>
</tr>
<tr>
<td>Port Number</td>
<td>Where</td>
<td>Which Component Uses It</td>
<td>Why</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>--------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>5168</td>
<td>Computer running the SQLdm Mobile &amp; Newsfeed Service</td>
<td>SQLdm Mobile &amp; Newsfeed Service</td>
<td>Allows communications between the Idera Newsfeed and SQLdm for server status updates.</td>
</tr>
</tbody>
</table>

**Mobile device requirements**

You can access SQLdm Mobile on any smart phone or other mobile device that meets these minimum operating system requirements:

- Android version 2.1 or later
- iOS version 3.2 or later
- RIM (BlackBerry OS) version 6.0 or later

You can run the SQLdm Mobile Web application in a native browser that supports HTML5, such as Google Chrome or Mozilla Firefox. SQLdm Mobile does not support Microsoft Internet Explorer 8 or earlier.

**TIP** If you plan to use IE 9 or later to access the SQLdm Mobile Web site, ensure the Display intranet sites in Compatibility View setting is disabled. To change this settings, click Compatibility View Settings on the Tools menu in the IE browser window.

**Web Server requirements**

The SQLdm Mobile Web application requires the following Web Server configuration.

<table>
<thead>
<tr>
<th>Type of Requirement</th>
<th>Minimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2.0 GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>1 GB</td>
</tr>
<tr>
<td>Disk space</td>
<td>21 MB</td>
</tr>
<tr>
<td>Microsoft Internet Information Services (IIS)</td>
<td>version 7.0 or later</td>
</tr>
<tr>
<td></td>
<td>By default, IIS 7 installs with Windows Server 2008. You can enable this role using the native Server Manager.</td>
</tr>
</tbody>
</table>

**TIP** To successfully run the SQLdm Mobile Web application, ensure the Common HTTP Features setting and the following service roles are enabled in IIS:

- Static Content Compression
- Default Document
- Directory Browsing
- HTTP Errors

**Repository requirements**

You can install the SQLdm Mobile Repository on a computer running the following versions of SQL Server software:

- SQL Server 2008 R2
- SQL Server 2008 SP2 or later
- SQL Server 2005 SP1 or later

Keep in mind that the Repository database and corresponding SQLdm Mobile & Newsfeed Service require 19 MB of disk space.
How to install SQLdm Mobile

This procedure guides you through the installation of SQLdm Mobile. Before installing SQLdm Mobile, review the associated architecture and components. Although you can install different components on different computers, this procedure recommends the following deployments:

- Install the Mobile and Newsfeed components on the same computer where your SQLdm Services reside
- Install the Mobile Web components on your corporate Web Server

**TIP** In larger environments that contain more than one SQLdm installation, consider dedicating a SQLdm Mobile deployment per each SQLdm application.

**Install or upgrade to SQLdm 7.1**

SQLdm Mobile, and the Idera Newsfeed technology, is supported by SQLdm 7.0 or later. To successfully install and deploy SQLdm Mobile, first install or upgrade to SQLdm 7.1.

**Start your SQLdm Mobile install**

You can install the SQLdm Mobile components on any computer that meets or exceeds the product requirements. By default, the SQLdm Mobile & Newsfeed setup program installs the SQLdm Mobile & Newsfeed Service to the local computer.

**To start installing SQLdm Mobile:**

1. Log on with an administrator account to the computer on which you want to install the SQLdm Mobile & Newsfeed Service. We recommend that you install this service on the same computer that already hosts your SQLdm services.
2. Run Setup.exe in the root of the SQLdm installation kit.
3. Click Install on the Welcome window, and then click Install SQLdm Mobile & Newsfeed on the Install window.
4. On the Welcome to SQL diagnostic manager Mobile, click Install, and then click SQLdm Mobile.
5. On the Welcome window of the SQLdm Mobile & Newsfeed setup program, click Next.
6. Review the license agreement. To accept this license agreement, click I accept the terms of this license agreement, and then click Next.

**Begin configuring your install**

We recommend installing the Mobile & Newsfeed components in your SQLdm environment, and then re-running the setup program to install the SQLdm Mobile Web application on your Web Server.

You can use the default install location or specify a different path. For your first install, we recommend using the default location. By default, the SQLdm Mobile & Newsfeed Service is also installed in the same location.

On the Custom Setup window, choose the Mobile & Newsfeed components, and then click Next.

**Which Windows credentials do you want to use for the SQLdm Mobile & Newsfeed Service account?**

You can specify the Local System account or a Windows user account. Ensure the specified account has the required permissions.

**To specify a service account:**

1. Select the Local System account or type the appropriate credentials in the provided fields.
2. Click Test Credentials to validate the account.
3. Click Next.
**Which SQL Server instance do you want to use for the Repository database?**

The SQLdm Mobile Repository is where the published status updates and other configuration settings are stored.

**TIP**  
SQLdm Mobile requires Microsoft SQL Server 2005 running on the computer that hosts the Mobile Repository database.

**To specify the SQL Server instance to install the SQLdm Repository on:**

1. Select the SQL Server instance you want to use. The SQL Server menu lists SQL Server instances provided by your domain controller. *If you do not see your SQL Server instance in the list*, type the host name and instance in the space provided.
2. Click Next.

**Which authentication should be used to create the Repository database?**

You can use either Windows or SQL Authentication to create the Mobile Repository database. Ensure the specified Windows account has the **required permissions**.

**To specify a service account:**

1. Choose the authentication that should be used, and then type the appropriate credentials in the provided fields.
2. Click Test Connection to validate the account can connect to the specified SQL Server instance.
3. Click Next.

**Which authentication should be used to connect to the Repository database at run time?**

You can use either Windows or SQL Authentication to connect to the Mobile Repository database. Ensure the specified Windows account has the **required permissions**.

**To specify a service account:**

1. Choose the authentication that should be used, and then type the appropriate credentials in the provided fields.
2. Click Test Connection to validate the account can connect to the specified SQL Server instance.
3. Click Next.

**Where is the SQLdm Management Service located?**

SQLdm Mobile uses its own Authentication Provider to enforce application security across your SQLdm and SQLdm Mobile deployments.

On the Authentication Provider window, specify the computer that hosts the SQLdm Management Service, and then click Next.

**Complete the Mobile & Newsfeed components install**

Click Install to indicate that you are ready to complete your install with the configurations you specified.

**Begin the SQLdm Mobile Web components install**

We recommend installing the SQLdm Mobile Web components on your corporate Web Server. Ensure your Web Server meets the software and port requirements.

**To start installing the SQLdm Mobile Web application:**

1. Log on with an administrator account to the computer on which you want to install the SQLdm Web application. We recommend that you install this application on the same Web Server that already hosts your corporate Web site.
2. Restart the SQLdm Mobile & Newsfeed setup program and navigate to the Custom Setup window.
3. Choose the Mobile Web components, and then click Next.

**Which Web Server settings should the SQLdm Mobile Web application use?**

The setup program installs the SQLdm Mobile Web application in a unique virtual directory on your Web Server computer. You can install the SQLdm Mobile Web application as a new Web site or add it to an existing Web site, such as your corporate intranet.

You also need to specify the name of the computer that hosts the SQLdm Mobile & Newsfeed Service so that communications can be sent between the SQLdm Mobile components and SQLdm.

Ensure your Web Server meets the SQLdm Mobile requirements.

**To configure the Web Server settings:**

1. Choose whether to install the SQLdm Mobile Web application as a new Web site.
2. Specify the name of the computer on which you installed the SQLdm Mobile & Newsfeed Service.
3. Click Next.

**Complete the Mobile Web components install**

Click Install to indicate that you are ready to complete your install with the configurations you specified.
After initially installing SQLdm, use the following configuration tasks to help you get the most out of SQLdm:

- Add your SQL Server instances
- Configure console options
- Connect to a SQLdm Repository
- Set your Repository grooming schedule
- Use tags to organize your monitoring environment
- Use custom counters to track metrics
The first time you open the SQLdm Console, the Connect to SQLdm Repository window opens. This window allows you to select the SQL Server instance, SQLdm Repository database name, and authentication the SQLdm Console uses to communicate with the SQLdm Repository.

**What access does the SQLdm Console require on the SQLdm Repository database?**

The SQLdm Console requires Read Access on the SQLdm Repository database. You can either use Windows Authentication or SQL Server Authentication for this connection.

**TIP**

The Connect to SQLdm Repository window can be open from the SQLdm Console menu bar by selecting File > Connect to SQLdm Repository. The ability to connect to another Repository is important in disaster recovery situations where you need to change the location of your SQLdm Repository.
Add your SQL Server instances

The Add Servers wizard allows you to add SQL Server instances to be monitored. Along with selecting the SQL Server instances you want to monitor, the Add Servers wizard also allows you to configure all the common options you use for monitoring SQL Server collection settings and alerts, and apply an alert template. These configuration settings are saved as a template that can be applied whenever you register new SQL Server instances for monitoring. To open the Add Servers wizard, select File > Manage Servers and click Add.

**TIP** If you are adding several SQL Server instances and you want to configure the same alert thresholds on each instance, consider setting up an alert template before adding any of your SQL Server instances.

**What is an alert template?**

SQLdm allows you to configure generic alert settings as a template that you can apply to servers and groups of servers in your organization. Click Tools > Alert Configuration Template and complete the required fields to configure an alert template.
Configure authentication

The Configure Authentication window allows you to select the authentication credentials SQLdm will use when collecting performance and statistics information on your SQL Server instances. You can also set encryption to secure the connection.

What connection type should I use?

When selecting a connection type, Windows Authentication is recommended. SQLdm uses a number of SQL trace statements that require a trusted connection. When Windows Authentication is selected, the account the SQL diagnostic manager Collection Service is running as will be used to connect to the SQL Server instance.

What encryption options, if any, should I choose?

SQLdm allows you to designate encryption methods used to encrypt data between the Collection Service and the monitored SQL Server instance. You can choose to use SSL (Secure Sockets Layer) or SSL with Trust Service Certificate.

Encrypt Connection (SSL)

This option sets a flag in the connection properties that is used when the collection service connects to a monitored server that specifies that SSL is to be used to encrypt the data between the collection service and the monitored SQL Server instance. For this option to work correctly, the monitored SQL Server instance needs to have been configured to use SSL encryption.

Trust Server Certificate (Bypass Certificate Validation)

This option can only be selected when the Encrypt Connection (SSL) option is selected. This option will allow the validation of the certificate to be skipped when a SQL Server instance establishes a connection. If SSL on the monitored SQL Server instance is not configured to use a certificate that the collection service trusts, the connection will be rejected unless the Trust Server Certificate option has been selected.
The Select Servers To Monitor window allows you to select one or more SQL Server instances to monitor in SQLdm. If the name of the SQL Server instance you want to monitor does not appear in the list, enter the name in the space provided. You can also designate specific ports for your SQL Server instances by entering a comma between the instance name and port number.

For example: domain\instance name,port

The Available Licenses field shows you the number of instances your current license permits SQLdm to monitor.

Click Next after you have added all the SQL Server instances to your list and are ready to continue.

TIP The Select Servers to Monitor window lists all the SQL Server instances provided by your domain controller. This list can include unsupported SQL Server instances. Be sure to select only SQL Server instances that are supported by SQLdm.
The Configure SQL diagnostic manager collection window allows you to set the collection interval and enable the Query Monitor.

**What collection interval should I choose?**

The collection interval you specify refers to the time SQLdm waits to collect statistical information on your SQL Server instance. This data includes the resource, session, database and tables, query, and service information.

Choosing the appropriate interval depends on your specific monitoring needs. When your SQL Server instance is performing poorly, you may want to decrease the collection interval so that you can track changes more frequently. When monitoring a SQL Server instance that has a very heavy client load and a large number of databases, you may want to increase the time period between collections to reduce the possibility of the frequent collections causing a performance impact on your clients.

**When do I use the Query Monitor?**

The Query Monitor is a standard SQL Server trace that collects the events that occur on your SQL Server instance over a period of time. This option can be enabled if you experience query timeouts or other performance issues.

| TIP | Enabling the Query Monitor can cause SQL Server performance degradation while the Query Monitor is running, and therefore should only be enabled when diagnosing specific query issues. |

Once your query diagnosis is complete, you can disable Query Monitor on the Monitored SQL Server Properties window (File > Properties).

For more information about the Query Monitor, see SQL Server Books Online.
This window allows you to assign an alert template and tags while you add instances. Alert templates allow you to configure generic alert settings to apply to your servers or groups of servers. Tags are labels that allow you to organize server instances into meaningful groups.

To apply an alert template, select the appropriate template from the list. You can create an alert template after you add your SQL Server instance, and then apply the template to your server. You may select one or more existing tags from the list, or create new ones by clicking Add Tag.

**What happens if I don't select an alert template?**

If you create an instance before creating any custom alert templates, SQLdm applies the default alert template automatically. You can then create an alert template and apply it to the server once the new template is available.

**What type of tags should I create?**

Tags are completely customizable for your needs. For example, you may want to view only server instances used by a particular department (like Accounting), in a specific location, or by some other method (like SQL Server version).

Each instance can be assigned multiple tags so that it appears in several different groups.
You can specify a number of console options in the Console Options window that affect overall product behavior. To access the Console Options window, select Tools > Console Options from the SQLdm Console toolbar.

The following tabs are available on the Console Options window:

**General**
- Allows you to set how often the console refreshes data, how chart data is managed, and allows you to hide the console when minimized.

**Alerts**
- Allows you to limit the number of alerts displayed in the Alerts view, which can improve the performance of the Alerts View when there are a large number of active alerts.

**Notifications**
- Allows you to configure the way that you are notified when alert thresholds are crossed.
Define common properties

The options on the General tab represent the most common properties users can change in the SQLdm Console.

How should I set my refresh intervals?

The Refresh Interval refers to the amount of time the SQLdm Console waits before gathering data from the SQLdm Repository. There are two types of refreshes. Each type is associated with different SQL Console components:

**Active View Refresh**

- Refers to the amount of time the SQLdm Console waits before refreshing the currently active view. When you first open a view, the view is refreshed with the latest data SQLdm has collected directly from the monitored SQL Server instance.
- You may want to lower the amount of time between refreshes if you are trying to diagnose a specific problem that requires a shorter interval. The interval for this field can be between one and 300 seconds.

**Status and Alerts Refresh**

- Refers to the amount of time the SQLdm Console waits before polling every monitored SQL Server instance and displaying notification pop-ups, updating the monitored SQL Server instance tree with status changes, and updating the SQLdm Today view with status changes. The interval for this field can be between one and 60 minutes.

What happens when I hide the desktop client?

The Hide when minimized toggle determines whether SQLdm displays as an active application in your Windows taskbar. When you want to open your SQLdm Console again, double-click the Idera SQL diagnostic manager icon in your taskbar. Enabling or disabling this option has no effect on data collection or the refresh rate of your currently active view.

How should I set up the chart options?

Chart options allow you to determine how much data is shown in your real-time charts.

**Keep data for the last**

- This field determines the length of time data is stored in the SQLdm Repository.

**Show data for the last**

- This field determines the time period displayed in real-time charts. The time entered cannot be any longer than the time specified in the Keep data for the last field.
Configure notification settings

The Notifications tab allows you to configure when and how alert notifications are displayed in the SQLdm Console.

What happens when I show alert notifications for state changes?

When Show Alert Notifications is selected, pop-up notifications will appear in your taskbar each time a threshold is met. To see what a console alert looks like, click Preview.

What state should I show notifications for?

You can select from the following alert notification options:

- **Always show console alert notifications**
  
  Selecting this option causes alert notifications to occur at every refresh.

- **Only show console alert notifications when a state transition occurs**
  
  Selecting this option limits the alert notifications you receive on refreshes to only those that have a state change.

- **Never show console alert notifications**
  
  Selecting this option will keep all console notifications from occurring. You will still receive all email alerts and To Do items you have configured. Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users. This setting is purely dependent upon your particular needs and can be modified at any point. If you are receiving too many notifications, consider limiting notifications to only occur when there is a alert state transition change. On the other hand, if you are monitoring a problematic SQL Server and want to know precisely when a threshold is met, consider monitoring all three states.

How do the notification popup settings work?

Notifications appear in your Windows taskbar and look very similar to a Microsoft Outlook notification.
The settings in this window allow you to control what occurs when you click a metric on the Dashboard view.

You can choose from the following options:

- **Show help popup**
  - Select this option to have informational help display when you click a metric item.

- **Drill down to the associated view**
  - Select this option to go directly to the view associated with the metric.

These options determine what happens when you left-click a metric. You can always right-click a metric and choose to see the Help, the details for the metric, or configure the associated alerts.

**When do help pop-ups occur?**

Help pop-ups give you information on particular metrics and the reason for any elevated status levels. This information can help you determine why elevated status levels exist so that you can diagnose problems with your SQL Server instance.
The Grooming Options window allows you to schedule and configure SQLdm Repository grooming. Grooming is the process of deleting metrics stored in the SQLdm Repository that are older than a given date and time. It is important to groom data from your Repository to optimize the performance of processes such as report generation and data collection.

**TIP** You can limit the amount of data stored in the SQLdm Repository by lowering the number of days data is stored. This is particularly important when monitoring a large number of servers or when storing Query information at a low threshold.

**What Grooming options are available?**

You can set the time SQLdm will groom the Repository and how many days your metrics, session details and queries, inactive alerts, and completed To Dos are saved in the Repository. In addition, you can select how often query data is aggregated and how often that data is groomed from the SQLdm Repository.

The current grooming status is listed at the bottom of the Grooming Options window.

**How do I access the Grooming Options window?**

To open the Grooming Options window, select **Tools > Grooming Options** from the toolbar menu.
SQL diagnostic manager, by default, displays a wide variety of the most common SQL Server and Operating System performance metrics. In addition to providing these common metrics, custom counters provide the ability to add additional performance metrics to the SQLdm console to be used for historical trending, on-demand monitoring, and schedule refresh alerting.

How do I create a custom counter?

To add a custom counter:

1. Select Administration in the navigation pane.
2. Select Custom Counters.
3. If this is your first time to access this page, click Click here to add a new custom counter now. If you have accessed this page before, click Add.
4. Click Next on the Add Custom Counter wizard Welcome page.
5. Select a counter type, and then click Next.
6. Select the counter you want to monitor, and then click Next.
7. Select the counter calculation type and scale factor, and then click Next.
8. Type a name for the custom counter, select a Category, add a Description, and then click Next.
9. Configure your alert settings related to the new custom counter, and then click Next.
10. Click Finish.

What custom counters can I add to the SQLdm interface?

You can add the following types of metrics:

- All Windows System Counters, including all Perfmon and WMI counters
- Any SQL Server System Counter stored in the sysperfinfo system table
- Any numerical value you would like to return from a custom SQL script

TIP In addition to providing all of the above counters, you can add any additional counters contained in a new Windows or SQL Server Service Pack. This allows you the flexibility to monitor any metric that is important to your enterprise.

Where can I find a list of Operating System counters?

You can find a list of counters by searching for "Windows Performance Counters" on technet.microsoft.com.

What kind of custom SQL scripts can I enter?

Custom SQL scripts can be used to monitor metrics that are not connected to system or SQL counters, under-utilized SQL 2005 DMVs as well as metrics that are very specific to your environment, such as rowcount for a production table.

TIP SQLdm does not scrub custom SQL batches for SQL Injection protection. Entering custom SQL scripts is an advanced feature that should be used with care. The go command is not a Transact-SQL statement and cannot be used in custom SQL batches. The go command is a batch separator for Management Studio, Query Analyzer, sqlcmb, and osql.

Where do my custom counters appear whenever I add them?

Custom counters appear in the following places in the SQLdm console:
**Alert Configuration window**

Your new custom counters appear in the Alert Configuration and the Default Alert Configuration windows in the Custom category. You can edit the alert thresholds, and even decide whether you want an alert associated with the new counter.

**Server Details view**

The Server Details view lists all the metrics monitored by SQL diagnostic manager, including all your new custom counters. You can filter this view to display only your custom counters by selecting the Custom Counters radio-button in the Filter ribbon. This window can be accessed either by selecting details on the Server view or by double-clicking Custom Counters in the Servers tree.

**Custom Counters table**

You can see a list of all your custom counters in the Custom Counters table on the Administration window. In addition to viewing a list of custom counters, you can add a new counter, edit a counter, delete a counter, test a counter, or link a counter to a SQL Server instance for monitoring.
Add a custom counter

Custom counters are Windows Operating System or SQL Server metrics that are not monitored, by default, by SQL diagnostic manager. In addition to Windows Operating System and SQL Server metrics, you can enter T-SQL scripts to monitor any variable in your environment. Once added to the SQLdm interface, you can monitor the results, view historical trending of the metric, configure alerts and receive notifications on them, and generate reports based on collected data.
The Select Counter Type window allows you to pick from the following counter types:

**Windows System Counter**
Select this option to pick from counters that include both those available in Windows System Performance Monitor (PerfMon) and Windows Management Instrumentation (WMI) performance counter types.

**SQL Server System Counter**
Select this option to pick from counters that are gathered via standard T-SQL from SQL Server.

**Custom SQL Script**
Select this option to enter a custom T-SQL script to monitor variables such as those which are not connected to the Operating System or SQL Server counters or metrics that are very specific to your environment such as a row count on a particular production table.
The Select Windows System Counter window allows you to select the counter you want from a list populated from those currently available on an existing monitored SQL Server instance, or you can enter the counter information manually.

**What do I do if I want to select the counter from the list?**

As you select from a populated list from those currently available on an existing monitored SQL Server instance, the next drop-down list will populate (filtered by your selection). This allows you to quickly find the specific counter you want to add to SQLdm.

1. Select whether you want to populate the list with counters that are contained in PerfMon, or populate it with every WMI counter.
2. Pick the SQL Server instance to populate counter information from the drop-down list.
3. Select the object name from the drop-down list.
4. Select the name of the specific counter you want to add from the drop-down list.
5. Select the instance from the drop-down list. The instance name is used to distinguish objects of a given type. For example, _Total refers to all instances, a drive letter such as C: for a specific disk drive, or the name of an application.

**TIP** Click the Details button for information on the object name and counter name you have selected.

**What do I do if I want to enter a counter manually?**

If you have the counter information available, you can enter the information quickly by simply pasting or typing the Object Name, Counter Name, and Instance name into their associated fields. The counter information entered must be the exact complicated name or the counter will not work correctly.

For example, to enter the Processor object name (as seen in PerfMon) the programmatic name entry could be the following depending on the system configuration:

```
Win32_PerfFormattedData_PerfOS_Processor
```

**TIP** Custom counter programmatic names can be found on Microsoft TechNet.
The Select SQL Server System Counter window allows you to select the counter you want from a list populated from those currently available on an existing monitored SQL Server or you can enter the counter information manually.

**What do I do if I want to select the counter from the list?**

As you select from a populated list from those currently available on an existing monitored SQL Server instance, the next drop-down list will populate (filtered by your selection). This allows you to quickly find the specific counter you want to add to SQLdm.

1. Pick the SQL Server instance to populate counter information from the drop-down list.
2. Select the object name from the drop-down list.
3. Select the name of the specific counter you want to add from the drop-down list.
4. Select the instance (if applicable) from the drop-down list. The instance name is what distinguishes it from other similar objects.

**What do I do if I want to enter a counter manually?**

If you have the counter information available, you can enter the information quickly by simply pasting or typing the Object Name, Counter Name, and Instance name into their associated fields. The counter information entered must be exact or the counter will not work correctly.

**What do I do if there are no counters in the drop-down list?**

If no counters appear in the drop-down list either the counters have not been installed or they may have been disabled by a recent Windows Service Pack.

For more information about the algorithm used to calculate a specific metric, see SQL diagnostic manager cannot load the performance counters needed for the statistics view pane - Solution #00000133 Idera Solution available at the Customer Service Portal (www.idera.com/support).

**Why does my SQL counter keep growing?**

Many SQL Server counter values start at zero when a server starts and then grow persistently from there, including most of the values in the sysperfinfo system table.

For example, the Buffer Manager Page Lookups/Sec counter in the value collected mode will always show the total number of page lookups since the server was first started, while when in the value per second mode it will show the number of lookups per second during the monitored period.
The Provide Custom SQL Script window allows you to enter or paste a T-SQL script for your custom counter. For information on T-SQL scripts, see SQL Server Books Online.

Refer to the following tips when using custom SQL Scripts:

- Dangerous or destructive T-SQL that is entered as a custom counter will be executed as entered. This could have a negative impact on the linked monitored SQL Server instances.
- The `go` command is not a Transact-SQL statement and cannot be used in custom SQL batches. The `go` command is a batch separator for Management Studio, Query Analyzer, sqlcmd, and osql.
- Custom counter scripts run at every scheduled refresh and on-demand on the Custom Counters view. For best performance, custom scripts should run in under 30 seconds. If the script takes longer than 2 minutes to execute, timeouts may occur.
- Custom SQL scripts must return a single result set with a single numeric field. Additional fields or non-numeric values will result in an error.
- Custom SQL scripts are displayed in Queries tab with "User Defined Counter" in their application name.
- The following options are set by default for custom counter scripts. If you need to use a different value, particularly another transaction isolation level, you should add the appropriate SET statement to the beginning of the script.
  - `set transaction isolation level read uncommitted`
  - `set lock_timeout 20000`
  - `set implicit_transactions off`
  - `set language us_english`
  - `set cursor_close_on_commit off`
  - `set query_governor_cost_limit 0`
  - `set numeric_roundabort off`
The Custom Counter Value window allows you to select the type of calculation used by your counter and set a scale factor so that your counter information displays values in their correct format. While the Scale drop-down list displays the same scale factors available in PerfMon, you can also type in a custom scale factor to use.

**Custom Calculation Type**
Choose to either have your counter display the value collected or calculate the value per second between refreshes.

**Customize Scale Factor**
Select a custom scale to convert the value into a more manageable number.

**What are some examples of scale factors?**
The following table covers some conversion factors that will allow you to precisely convert common units:

<table>
<thead>
<tr>
<th>Collected Value</th>
<th>Desired Value</th>
<th>Scale Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kilobyte</td>
<td>Byte</td>
<td>1024</td>
</tr>
<tr>
<td>Megabyte</td>
<td>Byte</td>
<td>1048576</td>
</tr>
<tr>
<td>Megabyte</td>
<td>Kilobyte</td>
<td>1024</td>
</tr>
<tr>
<td>Gigabyte</td>
<td>Kilobyte</td>
<td>1048576</td>
</tr>
<tr>
<td>Terabyte</td>
<td>Megabyte</td>
<td>1048576</td>
</tr>
<tr>
<td>Terabyte</td>
<td>Gigabyte</td>
<td>1024</td>
</tr>
<tr>
<td>Byte</td>
<td>Kilobyte</td>
<td>0.0009765625</td>
</tr>
<tr>
<td>Byte</td>
<td>Megabyte</td>
<td>0.000000953674316</td>
</tr>
<tr>
<td>Kilobyte</td>
<td>Megabyte</td>
<td>0.0009765625</td>
</tr>
<tr>
<td>Kilobyte</td>
<td>Gigabyte</td>
<td>0.0009765625</td>
</tr>
<tr>
<td>Megabyte</td>
<td>Gigabyte</td>
<td>0.0009765625</td>
</tr>
<tr>
<td>Megabyte</td>
<td>Terabyte</td>
<td>0.000000953674316</td>
</tr>
<tr>
<td>Gigabyte</td>
<td>Terabyte</td>
<td>0.0009765625</td>
</tr>
<tr>
<td>Kilobyte</td>
<td>SQL Server Page</td>
<td>0.125</td>
</tr>
<tr>
<td>Megabyte</td>
<td>SQL Server Page</td>
<td>128</td>
</tr>
<tr>
<td>Gigabyte</td>
<td>SQL Server Page</td>
<td>131072</td>
</tr>
<tr>
<td>SQL Server Page</td>
<td>Kilobyte</td>
<td>8</td>
</tr>
<tr>
<td>SQL Server page</td>
<td>Megabyte</td>
<td>0.007813</td>
</tr>
<tr>
<td>SQL Server Page</td>
<td>Gigabyte</td>
<td>0.000007629395</td>
</tr>
<tr>
<td>Kilobyte</td>
<td>SQL Server Extent</td>
<td>0.015625</td>
</tr>
<tr>
<td>Collected Value</td>
<td>Desired Value</td>
<td>Scale Factor</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Megabyte</td>
<td>SQL Server Extent</td>
<td>16</td>
</tr>
<tr>
<td>Gigabyte</td>
<td>SQL Server Extent</td>
<td>16384</td>
</tr>
<tr>
<td>SQL Server Extent</td>
<td>Kilobyte</td>
<td>64</td>
</tr>
<tr>
<td>SQL Server Extent</td>
<td>Megabyte</td>
<td>0.0625</td>
</tr>
<tr>
<td>SQL Server Extent</td>
<td>Gigabyte</td>
<td>0.000061035156</td>
</tr>
</tbody>
</table>
The Custom Counter Definition window is where you name, assign to a category, and enter a description of your new custom counter.

**How is the information I enter used in SQLdm?**

The information you enter on the Custom Counter Definition window appears in various places in the SQLdm interface.

**Name**

The name you give your new custom counter is the most important way to distinguish your custom counters. You should give them a name that directly relates to the information monitored.

**Category**

The category you assign to your new custom counter is important for organizing your custom counters, if you have many, on the Custom Counters window and on the Alert Configuration window.

**Description**

The description you give your custom counter is an important way of distinguishing your custom counters if the name you give them does not provide enough of a difference.

**TIP**

Once your custom counter has been created, you cannot change the name or category the counter has been assigned.
The Configure Alert Settings window allows you to configure how SQLdm uses and monitors your new custom counter information.

**Configure Alert Evaluation**
This option allows you to configure whether lower values are worse than higher values, such as the percent of free disk space on a drive, or higher values are worse than lower values, such as the percentage of free memory used.

**Configure Alert Thresholds**
The informational, warning, and critical thresholds you enter dictate the values that trigger alerts. Once configured, you can change these values on the Alert Configuration window.

**How do I configure a custom response when an alert is generated by SQLdm?**
SQLdm allows you to configure when and how alerts generate responses. You can select from the following alert responses (providers):
- Send an email to a person or group of people
- Log an event in the Windows Event Log
- Enable the Query Monitor
- Execute a program or utility
- Send an event to your Network Management tool
- Execute a SQL Agent Job
- Execute a Script
- Create a To Do (Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.)

To configure a custom response when an alert is generated, select Tools > Alert Actions and Responses from the console toolbar.
The Edit Custom Counter wizard allows you to edit your custom counter settings. To edit a custom counter select it from the list on the Custom Counters window and click the Edit button.

**TIP** The only custom counter information you cannot change is the name and category you have assigned it.
Delete a custom counter

Deleting a custom counter from SQLdm removes it from the interface and stops collection. Select a custom counter on the Custom Counters window and click Delete to remove it from SQLdm.
The Test Custom Counter window allows you to verify that the custom counter can be collected on the SQL Server instances you have linked it to and that the value returned is the value expected. A successful test displays the value of the metric collected, while a failed test provides the associated failure details.

TIP: Common causes for failed custom counter tests include inaccurate counter information entered manually and counters that do not exist on a particular SQL Server instance.
Link Custom Counter window allows you to select the monitored SQL Server instances upon which you want your custom counter to appear. Select Server tags from the Tags drop-down list or choose New tag from the list.

Use the Add and Remove buttons to select the SQL Server instances to link to your custom counter and click OK to save your changes.

Can I automatically add a customer counter to a new registered SQL Server instance?
Yes. By using links and tags, you can set SQLdm to automatically add any customer counter to a newly-registered SQL Server instance.

To set SQLdm to automatically add a customer counter to a new registered SQL Server instance:
1. Create a custom counter if the appropriate counter does not exist.
2. Create a tag if the appropriate tag does not exit.
3. Link a customer counter to a tag by selecting the custom counter, and then clicking Link. You can include more than one custom counter per tag.
4. Select the tag you want to use, and then click OK. The custom counter is now associated with the tag.
5. Select the tag you just linked when adding a new SQL Server instance.
Work with tags

Tags are customizable labels that allow you to organize and view your monitored resources in a manner best suited to your needs. You can assign tags to monitored SQL Server instances, custom counters, and permissions.

By creating and assigning tags to different objects, you can better manage how SQLdm presents information. For example, you may create an "Accounting" tag for all SQL Server instances used by the Accounting department. You could then copy alert configuration and assign custom counters and permissions to all the instances labeled "Accounting."

Tags are also accessible to all users. This gives you greater freedom, allowing users to work with the same tags across all SQLdm consoles.
The Manage tags window allows you to Add, Edit and Remove Tags. For more information on Tags and their functionality, see Working with tags.

**TIP**
This window is only available to SQLdm Administators.

**How do I access the Manage Tags window?**

To access the Manage Tags window:
1. Select **File**.
2. Select **Manage Tags**.

**What actions are available?**

The Manage Tags window shows a list of all existing tags. From here, you can:
- Modify the list of servers, custom counters, and permissions associated with a tag
- Create a new tag
- Edit an existing tag
- Remove a tag

**TIP** Removing a tag does not delete items associated with that tag.
The Create Tags window allows you to create and assign a tag while creating an instance. Tags are labels that allow you to organize server instances into meaningful groups.

You may select one or more existing tags from the list, or create new ones by clicking Create Tags.

**What type of tags should I create?**

Tags are customizable for your needs. For example, you may want to view only server instances used by a particular department, in a specific location, or by some other method, such as SQL Server version.

Each instance can be assigned multiple tags so that it appears in several different groups.
The Tag Properties window allows you to view and change the properties of an existing Tag. To access the Tag Properties window, right-click a Tag and select Tag Properties.

The following information is available and editable on the Tag Properties window:

**Servers**
- The Servers tab lists the SQL Server instances monitored by SQLdm and allows you to select or deselect the servers included in the tag.

**Custom Counters**
- The Custom Counters tab allows you to choose the custom counters you want to include in your tag.

**Application Security**
- The Application security tab allows you to select the application security permissions you want to apply to your tag.
Import historical data from a SQLdm 4.x installation

The Import wizard allows you to import historical data from your SQLdm 4.x Metrics Repository into the SQLdm Repository. To open the Import wizard, select File > Import from the Toolbar menu.

**TIP** For more information on the SQLdm upgrade process, see the Upgrading SQLdm Technical Solution in the documentation folder of your SQLdm installation.

**What historical data can I update?**

You can only import historical data from SQL Server 2000 or SQL Server 2005 instances that are monitored by SQLdm 5.x and registered in SQLdm 4.x
The Specify SQL diagnostic manager Repository window allows you to select the SQL Server instance and database of the SQLdm Repository you want to connect to.

**What does the Test button do?**

The Test button lets you confirm that SQLdm is able to connect to the Repository and use the credentials entered.
The Select SQL Servers window allows you to select the eligible SQL Server instances you want to import previously collected data into the new SQLdm Repository. To select a SQL Server instance, click the SQL Server instance name and click Add. After you have selected all the eligible SQL Server instances to import, click Next to continue.
The Specify Import Date window allows you to enter the date you want to import back to. If you are importing historical data from multiple SQL Server instances, you must select a date that is older than the oldest date in the Current As Of date for any of the imported SQL Server instances.

**How can I reduce the performance impact when importing a large amount of data?**

If you are importing a large amount of data, or are importing data from a large number of SQL Server instances, it is recommended that you import backwards in small steps (such a daily) to limit the performance impact of the import process.
Verify your import settings

The Import Verification window lists which SQL Server instances and associated dates you are importing historical data. If the settings are correct, click Next to begin the import process.
View the import status

The SQL diagnostic manager Import Wizard Status window displays the overall progress of your historical data import and any status messages. To hide the status window and receive a popup notification message when the import has completed, click Hide and Notify When Complete.
Monitor SQL Server Performance

SQLdm allows you to analyze the performance of your monitored SQL Server instances from My Views and from the Dashboard view.

**See the general status of all your monitored SQL Server instances**

The SQLdm Today view allows you to view the status of all your monitored SQL Server instances, including a listing of all your active alerts.

**Get an overview of a group of SQL Server instances**

My Views allows you to monitor either specific groups of SQL Server instances, or all groups. This overview includes monitors that allow you to view several important specifics, see at-a-glance if there are any problems, and drill down to detailed metrics for each component on the SQL Server instance.

**Get an overview of a specific SQL Server instance**

The Dashboard view provides a graphical representation of the activity on your SQL Server instance. This dashboard allows you to diagnose issues that are inter-related, using quick links to see the associated details and review possible solutions to specific issues.

**Interact with the selected SQL Server instance using Idera Newsfeed**

On each server view, you can interact with the selected SQL Server instance by either:

- Viewing the server profile
- Posting to the server's wall
- Following the server

For more information, see the Idera Newsfeed Help.
The Overview tab contains the following views:

- Dashboard view where you can see an overview of the instance.
- Details view that lets you create charts with the specific data items you need.
- Configuration view that allows you to view the configuration information of your SQL Server.
- Active Alerts view where you can see all outstanding alerts for this instance.
Navigate to your SQL Server instance

The Server tree contains all the SQL Server instances for the selected view, and allows you to drill down to the specific SQL Server components. Selecting these components opens corresponding views, providing detailed information.

The Server tree also displays tags, which are a way of labeling and organizing servers. The tree shows the defined tags and highlights when servers within the tag are in an alert state. Tags can be selected and used to show a filtered view of the servers within that tag.

How can I view alerts on the Server tree?

SQL Server instances in the Servers tree also display status icons. When a SQL Server instance has a warning or critical metric associated with it you can hover over the Server name and view the top three associated alerts. This allows you to prioritize your SQL Server instances and the work that needs to be performed.

What information does the Overview tab show?

When you select a SQL Server instance in the Servers tree, the Overview tab opens.

The Overview tab contains the following options:

- **Dashboard**
  - The Dashboard view is where you can see the status of many of the potentially related metrics graphically.

- **Details**
  - The Details view lets you view the values of each data item collected and create charts with the specific data items you need.

- **Configuration**
  - The Configuration view allows you to view and control the operational configuration of the monitored SQL Server.

What else can I do from the Servers tree?

By right-clicking a SQL Server instance in the Servers tree you can perform the following actions:

- **Open**
  - Opens the Dashboard view for the selected instance.

- **Refresh**
  - Gathers the alert status and refreshes the tree view information for the selected server.

- **Delete**
  - Allows you to delete the SQL Server instance from your SQLdm installation. When you select Delete, SQLdm displays a message asking whether you want to retain the collected data for the SQL Server instance, and it also allows you to cancel out of the deletion process.

  **TIP** You cannot retrieve collected data once it is deleted from the SQLdm Repository. Use the Delete option with care.

- **Maintenance Mode**
  - Allows you to take an individually monitored SQL Server instance offline for scheduled maintenance or other reasons.
Configure Alerts

Allows you to specify the alert criteria for each collected metric on the selected SQL Server instance.

Properties

Allows you to view and edit the collection properties of your SQL Server instance.
Get the SQL Server performance Overview

The Dashboard view contains a graphical representation of the activity on your SQL Server instance. The Dashboard view is structured to not only display the status of your SQL Server instance, but it allows you to diagnose where problems originate.

Each dashboard pane includes additional information specific to the metrics within that pane. Hover your mouse cursor over an area within a chart on which you want to get actual statistics. Click a specific metric to drill down for more detailed information. By using the options in the menu available when you right-click over an area, you can configure alerts, view associated help topics and details, print, save the image, or export the dashboard to Microsoft Excel as a .csv file.

How is the Dashboard view organized?

The Dashboard view is organized to display all the key metrics collected by SQLdm by category.

The main categories of data shown on the Dashboard view include the following:

**Cache**

The Cache dashboard tracks the database and procedure processes using a buffer pool of memory on your monitored SQL Server instance, and includes the following charts and gauges:

- Page Life Expectancy
- Areas
- Hit Ratios

**CPU**

The CPU dashboard includes key CPU metrics for your SQL Server instance and the computer that hosts the instance. The metrics displayed in CPU include the following charts and gauges:

- Usage Chart
- Processor Queue Length
- Call Rates

**Databases**

The Databases dashboard includes key metrics covering the health of the databases on your monitored SQL Server instance. The metrics displayed in Databases include the following charts and gauges:

- I/O Stall
- Log Flushes
- Reads
- Transactions
- Writes

**Disk**

The Disk dashboard includes key metrics that monitor the health of the disk of your monitored SQL Server instance. The metrics displayed in Disk include the following charts and gauges:

- Latency
Throughput
SQL Server Physical I/O

Lock Waits
The Lock Waits dashboard tracks the total number waits caused when a task is waiting to acquire a lock on your monitored SQL Server.

Memory
The Memory dashboard includes key Memory metrics for your SQL Server instance and the computer that hosts the instance. The metrics displayed in Memory include the following charts:
- Usage
- Paging

Network
The Network dashboard tracks the performance of the network connection being used by your monitored SQL Server, and includes the following chart and gauge:
- Throughput
- Response Time

Server Waits
The Server Waits dashboard tracks the total number and time spent on waits affecting your monitored SQL Server.

Sessions
The Sessions dashboard includes session-specific data items. The metrics displayed in Sessions include the following chart and gauge:
- Activity & Blocking
- Clients

What do the colors on the Dashboard view signify?
The following colors are associated with a status for each of the provided panes.

Gray
Indicates the status of the data is normal, or within the thresholds assigned to the metric.

Yellow
Indicates the status of the data is at a warning level, or outside the threshold of normal but not yet at the critical status.

Red
Indicates the status of the data is at a critical level and requires immediate action.

Why does a metric have no value?
If a metric on the Dashboard has no value, or displays as blank, this means SQLdm has not been able to collect the corresponding counter. This can happen due to one of the following reasons:
- The SQLdm Console has not captured the metrics. It can take up to two refreshes for this to occur.
- OLE Automation Procedures are disabled. SQLdm requires OLE Automation Procedures to collect data via WMI.
- OLE Automation Procedures are unavailable. SQLdm is unable to locate required OLE Automation Stored Procedures.
- WMI Service is not running. SQLdm requires that the WMI service be running on the computer that hosts the SQL Server instance in order to collect OS metrics.
- The counters do not exist on this SQL Server instance.

You can enable OS metrics monitoring to track the performance of the computers on which your monitored SQL Server instances are hosted.

**How often is the information on the Dashboard view refreshed?**

The Dashboard view is refreshed according to the time set in the Server View Refresh field on the Console Options window, which is accessible by clicking Tools > Console Options.
The Cache dashboard tracks the database and procedure processes using a buffer pool of memory on your monitored SQL Server instance. Unexpected spikes in the buffer and procedure cache may alert you to check that you have enough memory for the current pool to function properly. Chronically high cache metrics may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload.

**Page Life Expectancy chart**

The Page Life Expectancy chart provides the time in seconds that a page remains in the cache or buffer pool. The longer your page stays in the cache, the more likely the chance that SQL Server will find that data quickly without a read of physical media and minimize processing resource use. Use the Page Life Expectancy Alert to warn you of issues.

**Areas chart**

The Areas chart display the usage in megabytes of each type of cache in use.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Indicates the amount of space used by database processes.</td>
</tr>
<tr>
<td>Free</td>
<td>A low amount of free cache indicates a higher risk of performance issues.</td>
</tr>
<tr>
<td>Procedure</td>
<td>Indicates the amount of space used to store query plans.</td>
</tr>
<tr>
<td>Cache</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>Indicates the amount of space used by other processes.</td>
</tr>
</tbody>
</table>

**Hit Ratios gauge**

The Cache Hit Ratios gauge displays the hit rates of the Buffer and Procedure cache in this SQL Server instance. Use the Procedure Cache Hit Ratio Alert to warn you of issues.

**Available Alerts**

- Page Life Expectancy Alert
- Procedure Cache Hit Ratio Alert
CPU dashboard

The CPU dashboard tracks the performance of the processor being used by your monitored SQL Server. Unexpected spikes in CPU usage and call rates may warn you about the beginning of a serious performance issue. Chronically high CPU metrics may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload.

Usage chart

The CPU Usage chart displays the percentage of processing power in use (on the computer that hosts the SQL Server instance) over a period of time. The CPU view on the Resources view allows you to track your CPU usage over a period of time, along with other key CPU statistics.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL usage</td>
<td>A higher SQL usage indicates that SQL Server may be spending too much time processing queries.</td>
</tr>
<tr>
<td>Total usage</td>
<td>A high total CPU usage indicates that this server may not have enough resources to adequately process its current workload.</td>
</tr>
</tbody>
</table>

Processor Queue Length gauge

The CPU Processor Queue Length gauge displays the current value of the processor queue length metric. Use this metric to determine how much work is waiting to be done by this server. A high processor queue could indicate a blocking session or other performance issues.

Call Rates chart

The CPU Call Rates chart breaks down the processor's workload into the number of batches, compilations, and transactions completed each second, giving you a detailed view of which activities are resource-intensive.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batches/sec</td>
<td>A high throughput rate indicates a higher risk of network, CPU, and resource issues as the SQL Server performance degrades.</td>
</tr>
<tr>
<td>Compiles/sec</td>
<td>A high number of compilations (greater than 100 per second) can indicate a high server workload or that there is more recompilation overhead than necessary.</td>
</tr>
<tr>
<td>Re-Compiles/sec</td>
<td>A high re-compilation rate may be a sign of excessive re-compilation overhead.</td>
</tr>
<tr>
<td>Transactions/sec</td>
<td>A high rate of transactions indicates a higher risk of resource issues, such as blocks or locks, due to the heavier workload.</td>
</tr>
</tbody>
</table>

Available alerts

- OS Processor Time Alert
- Processor Queue Length Alert
- SQL Server CPU Usage Alert
The Databases dashboard tracks the performance of the databases on your monitored SQL Server. Unexpected spikes in database usage may warn you about the beginning of a serious performance issue. This feature offers a number of different metric sets based on your preference, which allows you to view your database transactions, log flushes, reads, writes, or I/O stalls. For each metric set, you can view a list of your top 100 databases sorted by current usage of that metric, and a bar representing the relative size of the associated database. A trends chart displays the same data over a period of time for only your top five databases.

I/O stall chart (ms/sec)

The I/O Stall ms/sec chart displays the total time per second (in milliseconds) that the databases on your SQL Server instance were waiting on an I/O to complete. Chronically high I/O stall metrics may indicate a disk bottleneck.

Log flushes chart (per second)

The Log Flushes/sec chart displays the number of log flushes performed per second and usually matches the number of transactions per second. If you experience a disk bottleneck, review to see if your log flushes greatly outnumber your transactions.

Reads chart (per second)

The Reads/sec chart displays the number of reads per second performed by the databases on your SQL Server instance.

Transactions chart (per second)

The Transactions/sec chart displays the number of transactions per second occurring on the databases on your SQL Server instance.

Writes chart (per second)

The Writes/sec chart displays the number of writes per second performed by the databases on your SQL Server instance.
The Disk dashboard tracks the performance of the disk being used by your monitored SQL Server. Unexpected spikes in latency, throughput, or SQL Server I/O may warn you about the beginning of a serious performance issue. Chronically high disk metrics may indicate something I don't know.

Use the Disk view on the Resources tab to track several key disk statistics over a period of time.

**Latency chart**

The Disk Latency chart plots the average time (in milliseconds) required to complete disk reads and writes, tracking the disk IO speed over time.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Disk ms/Read</td>
<td>Higher disk read times impact the ability of SQL queries to retrieve data from your databases.</td>
</tr>
<tr>
<td>Average Disk ms/Write</td>
<td>Higher disk write times impact the ability of SQL queries to update your databases, from simple text column modifications to dropping tables.</td>
</tr>
</tbody>
</table>

**Throughput chart**

The Disk Throughput chart plots the number of disk reads and writes over time.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk reads/sec</td>
<td>A high number of disk reads may indicate that the SQL Server is executing poorly-performing queries.</td>
</tr>
<tr>
<td>Disk writes/sec</td>
<td>A high number of disk writes may indicate that the SQL Server is running low on allocated memory.</td>
</tr>
</tbody>
</table>

**SQL Server Physical I/O chart**

The SQL Server Physical I/O chart breaks down the SQL Server throughput into its key component pieces, giving you a detailed view of how well SQL Server is processing your data requests and updates.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkpoint writes</td>
<td>A high number of checkpoint writes may indicate that pages are not being written to disk quickly enough for optimal performance.</td>
</tr>
<tr>
<td>Lazy writer writes</td>
<td>A high number of lazy writer writes may indicate that the SQL Server is running low on allocated memory.</td>
</tr>
<tr>
<td>Page Reads</td>
<td>A high number of page reads may indicate that the SQL Server is executing inefficient queries or that the database indexes should be optimized.</td>
</tr>
<tr>
<td>Page Writes</td>
<td>A high number of page writes may indicate that the SQL Server is executing inefficient queries or that the database indexes should be optimized.</td>
</tr>
<tr>
<td>Metric</td>
<td>Why it is important</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Read ahead reads</td>
<td>A high number of page writes may indicate that query execution plan should be tuned or that the disk should be defragmented.</td>
</tr>
</tbody>
</table>

**Available Alerts**

- Average Disk Milliseconds Per Read Alert
- Average Disk Milliseconds Per Write Alert
- Disk Reads (Per Second) Alert
- Disk Writes (Per Second) Alert
The Lock Waits dashboard tracks the total number waits caused when a task is waiting to acquire a lock on your monitored SQL Server. Unexpected spikes in the number of lock waits may warn you about the beginning of a serious performance issue. Chronically high lock wait numbers may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload.

**TIP** Lock Waits chart data is available only for monitored servers using SQL Server 2000. For SQL Server 2005 and above users, SQLdm monitors server waits.

**Lock Waits chart**

The Lock Waits chart displays the number of waits and the request type occurring.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Indicates the time waiting for a database lock.</td>
</tr>
<tr>
<td>Extent</td>
<td>Indicates the time waiting for an extent lock.</td>
</tr>
<tr>
<td>Key</td>
<td>Indicates the time waiting for a key lock.</td>
</tr>
<tr>
<td>Page</td>
<td>Indicates the time waiting for a page lock.</td>
</tr>
<tr>
<td>RID</td>
<td>Indicates the time waiting for a relative identifier (RID) lock.</td>
</tr>
<tr>
<td>Table</td>
<td>Indicates the time waiting for a table lock.</td>
</tr>
</tbody>
</table>
The Memory dashboard tracks the usage and availability of memory on your monitored SQL Server computer. Unexpected spikes in memory usage may warn you about the beginning of a serious performance issue. Chronically high memory metrics may indicate the need for server maintenance, query tuning, or other diagnostics to better handle the ongoing workload.

Use the Memory view on the Resources tab to track key memory statistics over a period of time.

**Usage chart**

The Memory Usage chart compares the amount of memory allocated to and used by SQL Server to the total memory consumed on this computer.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL allocated</td>
<td>Indicates how much available memory has been allocated for SQL Server. Compare this metric to how much memory is actually being used by SQL Server.</td>
</tr>
<tr>
<td>SQL used</td>
<td>A higher SQL memory percentage indicates that you may have poorly performing queries or a fragmented database.</td>
</tr>
<tr>
<td>Total used</td>
<td>A higher total memory percentage indicates that the overall workload of the SQL Server computer is too intensive for the available resources.</td>
</tr>
</tbody>
</table>

**Paging chart**

The Memory Paging chart plots the number of swapped pages per second, over time, as calculated by the memory paging metric.

**Available Alerts**

- O/S Memory Usage (Percent) Alert
- O/S Paging Alert
- SQL Memory Usage (Percent) Alert
The Network dashboard tracks the performance of the network connection being used by your monitored SQL Server. Unexpected spikes in packet rates and response times may warn you about the beginning of a serious performance issue. Chronically high network metrics may indicate excessive network traffic or a high server workload, which may require you to move some monitored instances to other computers to free up network resources and balance the workload.

**Throughput chart**

The **Network Throughput** chart plots how many packets are being sent from and received on this computer over time.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets received/sec</td>
<td>A consistently high rate may indicate that the packet size is too small.</td>
</tr>
<tr>
<td>Packets sent/sec</td>
<td>A consistently high rate may indicate that the packet size is too small.</td>
</tr>
</tbody>
</table>

**Response Time gauge**

The **Response Time** gauge displays the time in milliseconds SQL diagnostic manager currently needs to send a simple SQL command to the SQL Server instance, have it processed, and receive the returned result set. This value corresponds to the SQL Server Response Time metric.

**Available Alerts**

- SQL Server Response Time Alert
The Server Waits dashboard tracks the total number and time spent on waits affecting your monitored SQL Server. Unexpected spikes in the number of server waits may warn you about the beginning of a serious performance issue. Chronically high server wait times may indicate the need for server maintenance, query tuning, or index updates to better handle the ongoing workload.

**TIP** Server Waits chart data is not available for monitored servers using SQL Server 2000. For SQL Server 2000 users, SQLdm monitors lock waits.

### Server Waits chart

The Server Waits Chart displays the overall wait time for key areas on your monitored SQL Server instance.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O</td>
<td>Indicates the amount of input and output occurring on your SQL Server instance. A consistently high rate may indicate a higher risk of performance issues as the SQL Server processes wait for I/O to become available.</td>
</tr>
<tr>
<td>Locks</td>
<td>Indicates the wait time caused by locks on databases hosted by this SQL Server instance.</td>
</tr>
<tr>
<td>Log</td>
<td>Indicates the wait time related to database transaction logs.</td>
</tr>
<tr>
<td>Memory</td>
<td>Indicates all wait time related to memory operations.</td>
</tr>
<tr>
<td>Signal</td>
<td>Indicates that a SQL Server process is waiting for CPU time after the associated resources became available for processing.</td>
</tr>
<tr>
<td>Other</td>
<td>Indicates all lock waits not included in other categories and not excluded from monitoring.</td>
</tr>
</tbody>
</table>
The Sessions dashboard tracks the performance of the active and blocked sessions running on your monitored SQL Server. Unexpected spikes in the number of concurrent blocked sessions may warn you about the beginning of a serious performance issue. Chronically high blocked and deadlocked sessions are a significant performance concern.

Activity & Blocking chart

The Activity and Blocking chart shows the total number of active sessions on your monitored SQL Server instance and any blocking or deadlocks that may occur between sessions.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Indicates the number of active sessions on the SQL Server instance.</td>
</tr>
<tr>
<td>Blocked Sessions</td>
<td>Indicates the number of sessions on the SQL Server instance being blocked by other sessions holding requested locks.</td>
</tr>
<tr>
<td>Lead Blockers</td>
<td>Indicates sessions that block at least one session, which can then block other sessions.</td>
</tr>
<tr>
<td>Total Deadlocks</td>
<td>Indicates the total number of deadlocks on the SQL Server instance. A deadlock has no lead blocker and is circular in nature.</td>
</tr>
</tbody>
</table>

Clients gauge

The Client Computers gauge displays the total number of unique client computers connected to the SQL Server instance.

Available Alerts

- Blocked Sessions Alert
- Client Computers Alert
- Session CPU Time (seconds) Alert
- Total Deadlocks Alert
Get the SQL Server performance details

The Details view provides a list of all collected metrics, current values, state of the metric, and your alert settings. To add metrics to the Details chart, check the box in the chart column.

**TIP** All fields that are highlighted in the State column are associated with alerts. Right-click any of these metrics and select Configure Alerts to edit the associated thresholds.
The Configuration view allows you to view the SQL Server configuration settings for the selected SQL Server instance.

**How do I use the Configuration view?**

Primarily, the Configuration view is used for informational purposes. When diagnosing problems with SQL Server performance it can be helpful to see how it has been configured. In addition, SQLdm offers the ability to edit some configuration settings, depending on the option selected.

**To edit a configuration setting:**

1. Select an option from the list.
2. Click the Edit Value hyperlink.
3. Enter the new value.
4. Click OK.
The Active Alerts view allows you to view a list of all the alerts that were active as of the last data collection. Click any alert in the list to view details at the bottom of the window.

What can I do on the Active Alerts window?

The Active Alerts window includes the 12 Hour Forecast option to display the likelihood an alert is generated at a given time during the day, broken down into two-hour segments. This information helps you know on which areas you should concentrate your focus to avoid problems within your SQL Server environment.

In addition to providing the ability to see all your active alerts in a single window, the Active Alerts window allows you to do the following via the right-click context menu for each alert:

- Open the Real Time view to for more information on the metric that caused the alert (Show Real Time View).
- Open the alert configuration for the metric.
- Snooze the alert for a set period of time.
- Print, Export to Excel, or Copy the Alert to the Clipboard.
Get a 12-hour forecast of alert activity

Using current and historical data, SQLdm uses statistical analysis to estimate the likelihood that an event will generate a particular alert. This forecasting lets you proactively focus on certain areas in your SQL Server environment and prevent problems from occurring.

**How much forecasting data is required before the 12 Hour Forecast is available?**

SQLdm begins building forecasting models once you receive three critical or warning alerts for a given metric.

**What do the icons on the 12 Hour Forecast mean?**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![OK]</td>
<td>OK - There is a low chance (less than 20%) that a warning or critical alert will occur in the two-hour time segment leading to the forecast time.</td>
</tr>
<tr>
<td>![Warning]</td>
<td>Warning - There is a moderate likelihood (less than 50%) that a critical alert will occur, or a high likelihood (50% or higher) that a warning alert will occur, in the two-hour time segment leading to the forecast time.</td>
</tr>
<tr>
<td>![Critical]</td>
<td>Critical - There is a very high likelihood (50% or higher) that a critical alert will occur in the two-hour time segment leading to the forecast time.</td>
</tr>
</tbody>
</table>

**How do I use the 12 Hour Forecast?**

The 12 Hour Forecast is separated into two-hour segments. Each two-hour segment displays the chance that an alert will occur. Click the associated percentage chance to see the alert forecast details. This window displays the alerts that may occur over the two-hour time segment.
The Alert Forecast Details window displays the Server, Alert Metric, the Likelihood the alert occurs, and the type of alert expected to be generated in the two-hour window leading up to this forecast time. To sort the list, click any column heading.
You have the ability in SQLdm to snooze, or pause all alert generation for a specified period of time. This can be important if there are known issues occurring that you are working on and you want to avoid alert traffic.
The History Browser is intended to provide information pertaining to the state of your SQL Server instance at the time a standard snapshot is taken. This information can then be used to diagnose and resolve issues to keep the issue from happening again. The History Browser allows you to select a historical snapshot collected by the standard refresh and view the data that was collected using almost all the SQL diagnostic manager real-time views.

**How do I open the History Browser?**

To open the History Browser, click the History Browser button on a supported view (such as Overview > Summary).

**How do I select a specific snapshot in time in the History Browser?**

To select a specific snapshot:

1. Open the History Browser.
2. Select the date from the calendar. You can also select the time range to filter the number of snapshots displayed.
3. Select the corresponding snapshot from the Historical Snapshots list.

**TIP**
The icons next to each of the snapshots represent the highest level of alert thresholds that were violated. This can help you identify a specific snapshot. In addition, you can hover over snapshots in the list to see the most critical alerts associated with the snapshot.

**How do I return to Real-Time mode?**

To return to Real-Time mode, either click the You are currently viewing a historical snapshot. Click here to switch to Real Time Mode. text at the top of the window you are viewing historical data or click the associated text on a non-supported view.

**Which alerts have associated views that support historical snapshots?**

You can right-click any of the alerts (on the Alerts view) listed in the table below and select View Historical Snapshot to open the associated view at the time the alert was activated.

The following table displays all the alerts that have an associated view in SQLdm:

<table>
<thead>
<tr>
<th>Alert</th>
<th>Associated View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases Read/Write Errors Occurred</td>
<td>Databases</td>
</tr>
<tr>
<td>OS Average Disk Queue Length (Count)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Disk Time (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Memory Usage (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Paging (Per Second)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Privileged Time (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Processor Queue Length (Count)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Processor Time (Percent)</td>
<td>Resources</td>
</tr>
</tbody>
</table>
### Alert | Associated View
--- | ---
OS User Time (Percent) | Resources
SQL Server CPU Usage (Percent) | Resources
SQL Server Memory Usage (Percent) | Resources
Blocked Sessions (Count) | Sessions
Blocked Sessions Wait Time (Seconds) | Sessions
Client Computers (Count) | Sessions
Oldest Open Transactions (Minutes) | Sessions
Sessions CPU Time (Seconds) | Sessions
SQL Server Response Time (Milliseconds) | Sessions
User Connections (Percent) | Sessions

### Which views support historical snapshots?

Users may notify you of SQL Server problems that have happened in the past. Using the History Browser, you can go back to a point in time and view the state of your system and also compare system states at different points in time.

The following views support historical snapshots:

- Overview > Summary
- Overview > Details
- Sessions > Summary
- Sessions > Details
- Sessions > Locks
- Sessions > Blocking
- Resources > Summary
- Resources > CPU
- Resources > Memory
- Resources > Disk

### How long is historical data stored?

By default, Session details and inactive alerts are stored for 31 days while standard metric information is stored for 365 days. You can edit these options on the Grooming Options window.
Maintenance Mode allows you to temporarily stop alert generation and the collection of performance metrics for the time period that your SQL Server instance is offline.

**Why would I want to put a SQL Server instance into Maintenance Mode?**

Enabling Maintenance Mode lets you temporarily stop alert generation and performance metrics collection for the time period that the instance will be off-line. You should enable Maintenance Mode before you begin any regularly scheduled maintenance or testing on a monitored SQL Server instance. You can also enable Maintenance Mode when you need to bring a SQL Server instance off-line for unplanned diagnostics.

Be aware that the total number of monitored SQL Server instances allowed by your license includes monitored instances that are in Maintenance Mode.

**How do I bring the SQL Server back online in SQLdm?**

Disabling Maintenance Mode allows SQLdm to resume performance statistics collection and alert generation for the selected SQL Server instance. Before you enable Maintenance Mode, ensure all maintenance and testing has been completed on the target SQL Server instance.

If you have a SQL Server instance in Maintenance Mode and are ready to bring it back online in SQLdm, click the Maintenance Mode button on the Server Overview tab.
Configure SQLdm monitoring properties

The Monitored SQL Server Properties window allows you to edit the SQLdm monitoring settings for the registered SQL Server instance. To access the Monitored SQL Server Properties window, right-click a SQL Server instance from the Servers tree and select Properties.

The Monitored SQL Server Properties window includes the following tabs:

General
- Displays common properties such as the data collection interval, credentials to be used, and encryption options.

Query Monitor
- Allows you to enable or disable and set the options for the Query Monitor.

Table Statistics
- Allows you to select the time, days, SQLdm will collect table statistics on your SQL Server instance. In addition, you can select the minimum size a table needs to be for SQLdm to rebuild the index.

Maintenance Mode
- Allows you to enter the maintenance schedule for your server. This allows you to turn off alerts and notifications for the time periods when you know the server will be unavailable.

Custom Counters
- Allows you to link custom counters to this monitored SQL Server instance. Select from the available counters in the list and click the Add button to move the counters to the Linked Counters list.
Set general server options

The General tab allows you to edit the most common property settings. These settings include the Data Collection Interval or the amount of time between data collections, the server availability verification interval, and the credentials used to collect data.

Which tags should be associated with this SQL Server?

The tag selection drop-down allows you to either select a tag you have corrected to add your server to or add a new tag.

What data collection interval should I choose?

The data collection interval is the amount of time SQLdm waits before each collection of data on the selected SQL Server instance. The following factors should be taken into consideration when selecting your data collection interval:

- **How often you really need data collected**
  
  When diagnosing specific problems or when working with a problematic SQL Server instance, a short interval may be necessary to capture enough data to diagnose the issue. In most situations, the default interval should provide sufficient data for your diagnostic needs. Also, less used SQL Server instances may require less monitoring attention.

- **System resource impact**
  
  Although SQLdm limits the amount of system resource impact when collecting data, short collection times on SQL Server instances with large amounts of data could potentially cause system performance degradation. Using the default interval should meet your data collection needs while limiting any system resource impacts.

What server availability verification interval should I choose?

The server availability verification interval is the amount of time SQLdm waits before verifying availability on the selected SQL Server instance. If the connection test collector does not complete within the specified time, SQLdm creates an Unable to Connect event. The timeout used in this event is independent of the connection and command timeouts, which you can set at higher values to support probes unrelated to the connection test.

Because of these additional settings, the server refresh can also trigger an Unable to Connect event. For example, you may set one to 30 seconds and the other to six minutes, which means that in any six-minute interval, you might get 13 alerts, i.e., one every 30 seconds plus an additional hit for the six-minute setting.

Should I allow SQLdm to collect extended session data?

The extended session collection data includes session details, locks and blocks. If this information is important to you, ensure the check box is selected.

Which credentials should I use for collecting data?

SQLdm uses the specified credentials to collect data from the monitored SQL Server instance. You can choose to use either Windows authentication or SQL Server authentication.

**Windows authentication**

Windows Authentication uses the security of the operating system to create a trusted connection only if the account matches a security account defined in SQL Server. This security account must have sufficient permissions on the monitored instance to collect data and OS metrics.

**SQL Server authentication**

Select this option to use the credentials of a specific SQL Server account.
What encryption options, if any, should I choose?

SQLdm allows you to designate encryption methods used to encrypt data between the Collection Service and the monitored SQL Server instance. You can choose to use SSL or SSL with Trust Service Certificate.

**Encrypt Connection (SSL)**

This option sets a flag in the connection properties that is used when the collection service connects to a monitored server that specifies that SSL (Secure Sockets Layer) is to be used to encrypt the data between the collection service and the monitored SQL Server instance. For this option to work correctly, the monitored SQL Server instance needs to have been configured to support encryption.

**Trust Server Certificate (Bypass Certificate Validation)**

This option can only be selected when the Encrypt Connection (SSL) is selected. This option will allow the validation of the certificate to be skipped when a SQL Server instance establishes a connection. If SSL on the monitored SQL Server instance is not configured to use a certificate that the collection service trusts, the connection will be rejected unless the Trust Server Certificate option has been selected.
Set query monitor options

The Query Monitor is a standard SQL Server trace that collects all the events that occur on your SQL Server instance over a period of time. You can enable this option if you experience query timeouts or other performance issues. The Query Monitor window allows you to enable or disable the Query Monitor. If you enable the Query Monitor, you can select the settings that are used.

When do I use the Query Monitor?

You should enable the Query Monitor for the time period required to diagnose specific issues you experience. When your diagnosis is complete, disable the Query Monitor.

Once your query diagnosis is complete, you can disable Query Monitor on the Monitored SQL Server Properties window (File > Properties).

For more information about the Query Monitor, see SQL Server Books Online.

What happens if I capture deadlocks?

Capturing deadlock information allows the associated deadlock alert to provide details on deadlocked processes on the Alerts view. The details provided include:

- SPID
- Hostname
- Login Name
- Client Application
- Database
- Status
- Wait Time
- Wait Resource
- Transaction Count
- Input Buffer
- Lock Object
- Lock Mode
- Lock Owners
- Lock Waiters

What events should I capture in the Query Monitor Trace?

Select the events that help you best diagnose the specific issues occurring with query performance on your SQL Server instance. Enabling the Query Monitor can cause SQL Server performance degradation while the Query Monitor is running. You should enable the Query Monitor only when diagnosing specific query issues. For example, you may want to monitor queries taking a long time to complete, require heavy CPU usage, and cause a large number of logical disk reads or physical disk writes, which indicates a memory issue.

Why do I see scrambled output on the Queries tab when I use the Query Monitor to trace DTS packages?

SQLdm uses tracing to locate and flag worst-performing T-SQL. SQLdm cannot display binary data in a readable format. If you captured DTS packages, you may see unreadable characters in the Command field of the Tree View. Capturing DTS data
may also adversely affect the performance of the SQL Server or Data Warehouse you are monitoring.

What advanced query monitor configuration options are available?

You can choose the applications, databases and SQL text you want to exclude using the options available on the Advanced Query Monitor Configuration window accessible by clicking the More Options button on the Query Monitor tab.
The Advanced Query Monitor Configuration window allows you to filter out applications, databases, and SQL text from the Query Monitor Trace collection. This allows you to both improve the results you see in the Query Monitor view and reduce the impact of the Query Monitor trace on your server.

To exclude data from your Query Monitor trace:

1. On the Queries view for the server you want to exclude data from, press the Configure Query Monitor button.
2. Click Advanced.
3. Enter your filter information in the provided fields, using : to separate items and % as a wildcard.
4. Click OK.
The gathering of replication statistics can require a significant amount of system resources. The Replication tab of the Monitored SQL Server Properties window provides the option to disable the collection of replication statistics and conserve system resources.

**Why are replication statistics important?**

Collecting replication statistics requires a large amount of system resources, especially when your publisher or distribution queues contain a large number of items. You can easily enable this feature when you need replication statistics, and then disable it once you’ve collected enough data.

**Why would I disable the collection of replication statistics?**

Collecting replication statistics requires a large amount of system resources, especially when your publisher or distribution queues contain a large number of items. You can easily enable this feature when you need replication statistics, and then disable it once you’ve collected enough data.
Edit table statistics options

SQLdm lets you collect statistics and perform trend analysis over time. Table Statistics allows you to designate the days and times SQLdm collects table growth and reorganization statistics. Because statistic collection can impact performance, SQLdm allows you to designate an individual Table Statistics time for the SQL Server instances. Configure a time you know the instance is utilized less than normal.

In addition, you can limit the amount of performance impact by utilizing the following:

- Exclude non-essential databases from having table statistics collected from them by clicking Select Databases to Exclude and selecting the appropriate databases.
- Limit reorganization to databases over a set size, which eliminates the reorganization of small rarely used tables.

Scheduling table statistics collection

Table Statistics collection can be a resource-intensive process. Schedule your collection time for off hours to reduce the possibility of your users experiencing decreased performance.

TIP

Fragmentation data for a particular table may not be collected for any of the following reasons:

- Table statistics collection has not occurred
- The database hosting the table has been excluded from collection
- The table size does not meet the minimum size threshold
- The database hosting the table is not accessible
- The table is locked
SQL diagnostic manager, by default, displays a wide variety of the most common SQL Server and Operating System performance metrics. In addition to providing these common metrics, custom counters provide the ability to add additional performance metrics to the SQLdm console to be used for historical trending, on-demand monitoring, and schedule refresh alerting.

What custom counters can I add to the SQLdm interface?

You can add the following types of metrics:

- All Windows System Counters, including all Perfmon and WMI counters
- Any SQL Server System Counter stored in the sysperfinfo system table
- Any numerical value you would like to return from a custom SQL script.

**Tip** In addition to providing all of the above counters, you can add any additional counters contained in a new Windows or SQL Server Service Pack. This allows you the flexibility to monitor any metric that is important to your enterprise.

Where can I find a list of Operating System counters?

You can find a list of counters by searching for "Windows Performance Counters" on technet.microsoft.com.

What kind of custom SQL scripts can I enter?

Custom SQL scripts can be used to monitor metrics that are not connected to system or SQL counters, under-utilized SQL 2005 DMVs as well as metrics that are very specific to your environment, such as rowcount for a production table.

**Tip** SQLdm does not scrub custom SQL batches for SQL Injection protection. Entering custom SQL scripts is an advanced feature that should be used with care. The `go` command is not a Transact-SQL statement and cannot be used in custom SQL batches. The `go` command is a batch separator for Management Studio, Query Analyzer, sqlcmd, and osql.

Where do my custom counters appear whenever I add them?

Custom counters appear in the following places in the SQLdm console:

**Alert Configuration window**

Your new custom counters appear in the Alert Configuration and the Default Alert Configuration windows in the Custom category. You can edit the alert thresholds, and even decide whether you want an alert associated with the new counter.

**Server Details view**

The Server Details view lists all the metrics monitored by SQL diagnostic manager, including all your new custom counters. You can filter this view to display only your custom counters by selecting the Custom Counters radio-button in the Filter ribbon. This window can be accessed either by selecting details on the Server view or by double-clicking Custom Counters in the Servers tree.

**Custom Counters table**

You can see a list of all your custom counters in the Custom Counters table on the Administration window. In addition to viewing a list of custom counters, you can add a new counter, edit a counter, delete a counter, test a counter, or link a counter to a SQL Server instance for monitoring.
Schedule maintenance mode

Maintenance mode allows you to temporarily stop alert generation and the collection of performance metrics for the time period that your SQL Server instance will be offline. This allows you to perform necessary maintenance on a SQL Server without causing numerous misleading alerts to be sent.

What settings are available?

- **Never**
  The server will never be in maintenance mode and will always generate alerts.

- **Until further notice**
  The server will be in maintenance mode and not generate alerts until this setting is changed.

- **Recurring every week at the specified time**
  The server can be set to enter maintenance mode at regular intervals, for the day, time, and duration you select.

- **Occurring once at the specified time**
  The server will enter maintenance mode at a specified date and time, and leave maintenance mode at a specified date and time.
Configure OLE automation

SQLdm uses OLE Automation when collecting Operating System metrics and service status on clustered SQL Servers. By default, OLE Automation is enabled. Check Disable collection requiring OLE Automation if you do not want to collect these metrics.

What are the repercussions if I disable OLE Automation?

SQLdm collects diagnostic information using OLE Automation. While not required, if you disable OLE Automation, SQLdm ceases to collect this information and does not raise any previously associated alerts. You can easily re-enable OLE Automation at any time if you decide that you want to continue collecting these metrics.
Disk drive statistics collection provides the ability for SQLdm to alert on disk drives and database space calculations. By default, SQLdm attempts to auto-discover local disk drives on monitored instances. In some environments, the auto-discovered list may be incomplete or incorrect. The Customize disk drive statistics collection window allows you to turn off the auto-discover feature and manually add the drives for drive statistics collection.

**What do I do if my disk drive list is incomplete?**

SQLdm automatically discovers local disk drives on your monitored instances. Sometimes this list is incomplete or contain drives that are not related to SQL Server. If you need to modify this list, clear the Discover disk drives check box, and then type the drive information including mount points, if necessary, into the Available disk drives field. Select the appropriate drive or drives and then click Add to move the drive information to the Selected disk drives box. Click OK.
Configure cluster settings

If you have a clustered environment, SQLdm can trigger an alert whenever a cluster is running on a non-preferred node. The preferred cluster node is the node that should always be the "active" node in a cluster. By default, the preferred cluster node is set to be the first node added when deploying SQLdm in your clustered environment.
SQLdm allows you to collect and view wait statistics on the selected SQL Server instance. Wait statistics offer one of the best means in diagnosing the reason a bottleneck is occurring on your system.

The Wait Monitoring tab of the Monitored SQL Server Properties window allows you to configure how SQLdm collects wait statistics.

- View, start, stop, and refresh the query-level wait collector
- Collect query-level wait statistics historically (and set the time to collect them)

**When do I use wait monitoring?**

You should enable the Query Monitor for the time period required to diagnose specific issues you are having. When your diagnosis is complete, disable the Query Monitor.
Enable OS metrics monitoring

SQLdm collects metrics that track the performance of the computers that your monitored SQL Server instances are hosted on including the following:

- CPU Usage
- Processor Queue Length
- Memory Paging
- Total Memory Usage
- SQL Server Memory Usage
- Disk Queue Length

SQLdm requires that OLE Automation procedures be enabled for monitoring OS metrics. In addition, if Lightweight Pooling is enabled on your SQL Server instance, SQLdm will be unable to collect OS Metrics information. If SQLdm is unable to provide OS Metrics, a message displays on the Dashboard and Services Summary views providing more information.

**TIP**

When SQLdm is unable to collect OS Metrics due to disabled OLE Automation, a bar displays with a link to enable OLE Automation procedures for SQL Server 2005 and SQL Server 2008.

**What do I need to do if I have configured OLE Automation to run out-of-proc?**

You will need to edit specific registry settings to allow WMI to run out-of-proc.

**To edit these registry settings, perform the following steps:**

1. In the cd image of the SQLdm install, open the Scripts folder.
2. Copy the SQLdmoutofprocoleautomation.reg file onto each of your monitored SQL Server instances.
3. Run this file on each of your monitored SQL Server instances.

The following problems require user action before SQLdm can begin collecting OS metrics:

**OLE Automation Procedures Disabled**

Click the Enable OLE Automation Procedures message to enable OLE Automation Procedures.

**OLE Automation Procedures Unavailable**

The stored procedures associated with OLE Automation may have been removed from the SQL Server instance. SQLdm cannot capture OS metrics until these stored procedures are available on the SQL Server instance.

**WMI Service is not running**

The WMI Service on the monitored SQL Server instance may not be running. SQLdm cannot collect OS metrics until this service has been started.

**Lightweight Pooling is enabled**

This is an advanced feature used in symmetric multiprocessing environments in SQL Server 2005. SQLdm cannot collect OS metrics until this option has been disabled on your monitored SQL Server instances.
The Sessions tab contains the following views:
- Summary view where you can view charts for each Sessions statistic.
- Details view that lets you create charts with the specific data items you need.
- Locks view that allows you to create charts based on each lock present on your SQL Server.
- Blocking view where you can create charts specific to all the Blocking sessions.

The Sessions tab provides the following options:
- The ability to diagnose performance bottlenecks caused by blocking or locked sessions.
- The ability to trace specific sessions to see why the session may be blocking or locked.
- The ability to kill sessions to improve system performance.

The Query tab provides views for analyzing the data that is captured by the Query Monitor trace. The Query Monitor is used to diagnose issues with your SQL Server instances.

What alerts are associated with the Sessions and Query Performance tabs?
The following alerts are associated with the Sessions and Query Performance tabs:

- **User Connections (Percent)**
  Is the percentage of user connections allowed that are currently used.

- **SQL Server Response Time (Milliseconds)**
  Calculating the sum of the response time (in milliseconds) to send a batch of SQL statements to the server, process the query, and return the results to the SQLdm Console.

- **Session CPU Time**
  Provides the CPU time (in seconds) used by a SQL Server sessions.

- **Blocked Session Wait Time (Seconds)**
  Is the amount of time (in seconds) that a SQL Server session has been blocked.

- **Client Computers**
  Is the number of unique client computers connected to the instance.

- **Blocking Sessions**
  Is the number of sessions blocked by other sessions holding requested locks.

- **Query Monitor Status (Count)**
  Is the number of queries captured by Query Monitor on the monitored SQL Server instance during the last refresh period.

- **Oldest Open Transaction (Minutes)**
  Displays the longest amount of time between a transaction using a Being Transaction statement and the corresponding Commit Transaction or Rollback Transaction statement, in minutes.

- **Deadlock**
  Is when two processes are waiting for a resource held by the other process.
The Sessions Details view provides an in-depth analysis of sessions running on your monitored SQL Server instance. You can view a wide range of information from performance details to open transactions and configured options or track process activity at the statement level.

Individual sessions appear in the top portion of the window. Right-click any session in the list to view locks, trace the session, kill the session, print the associated statistics, or export statistics to Excel. When a session is selected, the details for that session appear in the Details section at the bottom of the window. The details section includes the Last Command which can be copied to the Windows clipboard.
The Sessions Summary view displays real-time charts that allow you to quickly view key diagnostic statistics for sessions on your SQL Server instance. Click any chart in this view to see detailed information on the chart. The Sessions Summary tab includes the Response Time, Sessions, Lock Statistics, and Blocked Sessions charts.

**How do I use the Response Time chart?**

The **Response Time chart** displays the response time, in milliseconds, it takes a *Select1* SQL statement to go to the SQL Server instance and back.

You can see the specific time and response time of any data point in the chart by mousing over it.

The Response Time chart indicates the current speed and congestion of the network as well as the speed with which SQL Server processes small queries. Unusually high response times could indicate the need to upgrade server or network hardware.

**How do I use the Sessions chart?**

The **Sessions chart** displays the number of concurrent connections to the monitored SQL Server instance over time. The total number of sessions includes the active, idle, and system sessions.

You can see the specific metric value for a given time by hovering your mouse over the chart.

**How do I use the Lock Statistics chart?**

The **Lock Statistics chart** provides a breakdown of the various lock types taken by SQL Server in the last refresh. The Lock Statistics chart offers, by using the drop-down menu, charts that include information on the Average Wait Time, Deadlocks, Requests, Timeouts, Waits, and Wait Time.

You can see the specific time and statistic on any data point on the Lock Statistics for a given time by hovering your mouse over the chart.

**How do I use the Blocked Sessions chart?**

The **Blocked Sessions charts** lists all the Blocked Sessions, Lead Blockers, and the Total Deadlocks, in real-time, for the selected SQL Server instance. This allows you to easily see blocks as they occur.

You can see the specific time and number of blocked sessions at a given time by hovering your mouse over the chart.
On the Locked Sessions view, you can see which problem locks are causing data availability problems and unacceptable responses. By combining the information provided on the Problem Locks view (including the type of Lock) with the Lock Statistics: Waits chart on the Sessions Summary tab, you can pinpoint, correct, or reschedule automated applications that cause extensive row, table, or database locking.

How do I use the Lock Statistics chart?

Select the drop-down list in the chart title to select charts that include:

- Average Wait Time
- Deadlocks
- Requests
- Timeouts
- Waits
- Wait Time information

You can right-click any of these charts and either print, save as image, or export them to Microsoft Excel. In addition, you can select Toolbar for advanced customization options such as changing the chart color scheme and the type of chart shown.
The Blocking view provides a blocking tree, along with the lead blockers, for the selected SQL Server instance. A lead blocker is a session that blocks at least one session, which can in turn block other sessions.

**TIP**
You can use the History Browser to view past performance and go back to the point in time to view the sessions that were involved with the blocking alerts you receive.

Navigate through the blocking tree to find sessions that cause blocks, and then either trace sessions to find out what is causing the blocks or stop the session to have it removed. To trace a blocked session or lead blocker select it from the list and click **Trace Session**. To stop a blocked session, or lead blocker, select it from the list and click **Kill Session**.

**TIP**
Deadlocks have no lead blockers and are circular in nature, for this reason they are not displayed in the Blocking Tree but instances of deadlocks are shown in the Blocking Chart. In addition, self blocking sessions are not counted as either blocked or blocking sessions.

**What happens when I trace a session?**
When you click **Trace Session**, the Session Trace window opens. The Session Trace provides session statistics and information including the following:

- Overview of the session, including the CPU Time, Row Count, Lock Wait Timeout, Physical Reads and Write metrics
- Last Command executed
- All the SQL Server statements

**How do I use the Blocking chart?**
The **Blocking chart** displays the number of blocked sessions, lead blockers, and total deadlocks at a given point in time. The total deadlocks represent the total number of deadlocks that have occurred since the last time the pane was refreshed.

You can right-click the chart and either print, save as image, or export this data to Microsoft Excel. In addition, you can select **Toolbar** for advanced customization options, such as changing the chart color scheme and the type of chart shown.
Monitor query performance

Query Monitor is a useful SQL Server tool that allows you to log and analyze queries. Query Monitor allows you to establish criteria for the queries you wish to capture, and filter those results to analyze the queries returned.

**Why would I start a Query Monitor?**

The Query Monitor is a very useful tool when diagnosing poor query performance on your SQL Server instance.

**When would I start a Query Monitor?**

You should monitor queries if your SQL Server instance performance is poor, as a way to diagnose potential query issues.

The Query Monitor can be enabled and configured on the Monitored SQL Server Properties window. To open the Monitored SQL Server Properties window, right-click any server in the Server tree and select **Properties**.

**What information can I view on the Queries tab?**

The Queries tab lists each event that occurs in the Query Monitor. For each event, among other details, SQLdm lists the SQL Text associated, start and end times, application, user information, and client computer associated with the event.

The **Event Occurrences** chart at the bottom of the view displays the number of T-SQL statements, stored procedures, and SQL batches captured in the trace.

**TIP** Right-click any column heading to pick additional columns to display.

**What is the difference between Signature Mode and Statement Mode?**

Signature and Statement modes are two different ways to group and view your queries. You can toggle between each mode to see the differences.

**Signature Mode (default Queries view)**

All active queries will be automatically reduced to the signatures (parameters will be stripped) and grouped accordingly.

**Statement Mode**

All queries are presented exactly as they are collected by the Query Monitor Trace.

**How can I use Filter Settings to customize my view?**

Use the Filters section of the Signature Mode, Statement Mode, and Query History views to pick the time range, what to include in the table, and even specific applications, client computers, databases, SQL text, and users to view.

You can access advanced filters by clicking the Filter button. These options include the ability to filter out specific applications, databases and SQL text from your query collection.

**TIP** The Filter and List areas of the Query Monitor views can be hidden and unhidden in the ribbon.

**How are my queries named in SQLdm?**

SQLdm automatically assigns a name to each of your queries starting with Query 1. The name provides context to your query when switching from signature mode, statement mode, and query history views.
Query names are not persistent. The names will be different on different desktops and in different sessions on the same desktop.

What does the Sigma (Σ) character mean on a row?

The Sigma character indicates that the data in this row is at least partially composed of aggregated data. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed. Rows on this view with the Sigma character may include a mix of aggregated and non-aggregated data.

How can I see all the SQL text in a particular field?

Fields with SQL text hold 255 characters of text. To view the full SQL text, right-click the field and select the appropriate option.
The Query Monitor Signature Mode view includes all active query information, automatically reduced to the signatures and grouped accordingly. Note that the parameters are stripped.

**TIP** If you want to prevent certain queries from appearing on the Query Monitor views, click Configure > More Options and enter the applications, databases, or SQL text you want to exclude from the Query Monitor collection.

**How do I use the Query Monitor Signature Mode view?**

The Query Monitor Signature Mode view is the place to start when investigating query performance. It broadly defines queries and trends with a less overwhelming amount of data and allows you to diagnose a query in a general sense. Once you identify a potential problem, you may want to view the statement mode view to see why a query is experiencing performance problems for a particular user, and then to the query history to see how the query performs over time and the associated trends.

The primary focus of the Signature Mode view is in the charts. The chart type drop-down allows you to pick a chart type to view, while the chart category drop-down allows you to choose to view the data by SQL Text, Application, or Database.

Use Query filters to narrow the results to only those that most interest you. Do this by selecting the date and time range, the application, user, database, workstation, and the SQL text you want to include or exclude from your results.

Event Occurrences is a listing of each occurrence of the query and the associated statistics, such as the duration, CPU time, reads, writes, and associated SQL text.

**What does the Sigma (Σ) character mean on a row?**

The Sigma character indicates that the data in this row is at least partially composed of aggregated data. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed. Rows on this view with the Sigma character may include a mix of aggregated and non-aggregated data.

**What do the colors in the cells mean?**

Red in the column indicates that the data in that row represents 20 percent or more of the total data displayed in the list, while yellow indicates that the data in the row represents 5 percent or more of the total data displayed in the list.
View the query monitor statement mode

The Statement Mode view includes queries that are presented exactly as they are collected by the Query Monitor Trace.

**TIP** If queries appear on the Query Monitor views that you never want to appear, click Configure > More Options and enter the applications, databases or SQL text you want to exclude from the Query Monitor collection.

**How do I use the Query Monitor Statement Mode view?**

The Statement Mode view is where you go to drill-down into a specific execution of a particular query. Whereas the signature mode view provides a filtered look at your queries, the Statement Mode view provides all the detail you may need to diagnose a specific problem with a query. Once you find the problem query, History mode allows you to track the query over time and view trends.

The primary focus of the Statement Mode view is in the charts. The chart type drop-down allows you to pick a chart type to display, while the chart category drop-down allows you to choose to view the data by SQL Text, Application, Database, User, or Host. Click a query in a chart to open up the query history for that query.

Use Query filters to narrow the results to only those that most interest you. Do this by selecting the data and time range, the application, user, database, workstation and even the SQL text you want to include or exclude from your results.

Event Occurrences is a listing of each occurrence of the query and the associated statistics, such as the duration, CPU time, reads, writes, and associated SQL text.

**What does the Sigma (Σ) character mean on a row?**

The Sigma character indicates that the data in this row is at least partially composed of aggregated data. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed. Rows on this view with the Sigma character may include a mix of aggregated and non-aggregated data.

**What do the colors in the cells mean?**

Red in the column indicates that the data in that row represents 20 percent or more of the total data displayed in the list, while yellow indicates that the data in the row represents five percent or more of the total data displayed in the list.
Query History allows you to track the performance of a query over time. To access the Query History view, select the query you want to see on the query statement mode or signature mode view in the graph. You can also access the Query History view by right-clicking a session on the Session Details view or a Statement on the Query Waits view and selecting Show Query History.

**How can I use the Query History view?**

1. Use the drop-down menu on each chart to view the history of each of the metrics associated with the query.
2. Click View SQL Text to see the associated query text.
3. Click Keep Detailed History and SQLdm keeps a detailed history of the query going forward.
4. Click Aggregate History to keep an aggregated history for the selected query.

Use Query filters to narrow the results to only those that most interest you. Do this by selecting the data and time range, the application, user, database, workstation and even the SQL text you want to include or exclude from your results.

Event Occurrences is a list of each occurrence of the query and the associated statistics, such as the duration, CPU time, reads, writes, and associated SQL text. Click the Maximize window icon to view a larger version of this list.

**What does the Sigma (Σ) character mean on a row?**

The Sigma character indicates that the data in this row is aggregated. This means that the data is condensed into a day-length summary and some precision about the individual statements, client computers, and users is suppressed.

**What do the colors in the cells mean?**

Red in the column indicates that the data in that row represents 20 percent or more of the total data displayed in the list, while yellow indicates that the data in the row represents five percent or more of the total data displayed in the list.
The Resources tab allows you to monitor the system resources on the computer hosting the SQL Server instance as well as what used by SQL Server.

Resource bottlenecks are often the cause of SQL Server performance problems. These bottlenecks can be the result of poor database application design or point to other system processes using valuable resources needed by SQL Server.

What alerts are associated with the Resources tab?

The following alerts are associated with the Resources tab:

**SQL Server CPU Usage (Percent)**
- Indicates the percentage of the total CPU used by the SQL Server instance.

**SQL Server Memory Usage (Percent)**
- Indicates the percentage of total memory used by SQL Server.

**OS Used Memory (Percent)**
- Indicates the percentage of total memory used on the SQL Server computer.

**OS Paging (Per Second)**
- Indicates the page fault rate for all processes on the computer hosting the SQL Server instance.

**OS Processor Time (Percent)**
- Indicates the percentage of CPU time used by all processes on the computer hosting the SQL Server instance.

**OS Paging (Per Second)**
- Indicates the page fault rate for all processes on the SQL Server computer.

**OS Privileged Time (Percent)**
- Indicates the percentage of CPU time used by all processes on the computer hosting the SQL Server instance, executed in Privileged mode.

**OS User Time (Percent)**
- Indicates the percentage of CPU time spent by all processes, on the computer hosting the SQL Server instance, executing in User mode.

**OS Processor Queue Length**
- Is the number of ready threads in the processor queue on the computer hosting the SQL Server instance.

**OS Disk Time (Percent)**
- Indicates the percentage of elapsed time that all the disks were busy servicing read and write requests on the SQL Server computer.

**OS Average Disk Queue Length**
- Is the average number of both read and write requests that were queued for all the disks on the computer hosting the SQL Server instance.
Procedure Cache Hit Ratio (Percent)

The ratio of procedure cache hits to procedure cache lookups. This indicates when an execution plan is reused from memory as opposed to compiled from disk.

Page Life Expectancy

The number of seconds a page stays in the buffer pool without references.

Disk Reads / Seconds Per Disk

The number of disk reads per second, per disk

Disk Writes / Seconds Per Disk

The number of disk writes per second, per disk

Disk Transfers / Seconds Per Disk

The number of disk transfers per second, per disk
The Resources Summary tab displays real-time charts that allow you to quickly view key diagnostic statistics for monitoring the resources for the selected SQL Server instance and computer on which it resides. The Resources Summary tab is especially useful in determining resource bottlenecks. For more details, click any of the charts to navigate to the associated tab.

How do I use the CPU Usage chart?

The CPU Usage chart can be used to determine the amount of strain your SQL Server instance is putting upon the CPU. This is an important factor in analyzing poorly performing SQL Server instances. Click this chart to open the CPU, which contains more charts and information on your CPU resources.

TIP SQL Server should be the primary process consuming CPU on the computer hosting your instance. If this is the case and the total CPU usage is consistently higher than 80 percent, attention is required. This could indicate the need to either upgrade your CPU or add more processors.

How do I use the Call Rates chart?

The Call Rates chart can help you determine why your CPU usage is high. The following metrics are displayed in the Call Rates chart:

- **Batches**: Displays the rate at which SQL Statement batches are issued by clients on the SQL Server.

- **Compiles**: Displays the rate at which SQL Statements are compiled into the Procedure Cache.

- **Re-Compiles**: Displays the rate SQL Server is recompiling stored SQL Statements into the Procedure Cache.

TIP If the number of re-compiles is consistently higher than 10, or the number of compiles is high, you may need to tune your queries to be less CPU intensive. Consider running the Query Monitor as a way of diagnosing query performance issues. The Query Monitor can be enabled on the Query Monitor tab of the Monitored SQL Server Properties window.

How do I use the Memory Usage chart?

The Memory Usage chart allows you to analyze the way in which memory is used on your SQL Server instance. The chart provides the amount of memory allocated to SQL Server, the memory currently used by SQL Server, and the total amount used on the host computer.

TIP If the SQL Used metric is consistently high compared to the amount of physical memory on the computer where the SQL Server instance resides, it may indicate that more physical memory is required.

How do I use the Cache Hit Ratios chart?

The Cache Hit Ratios chart allows you to view how often a request is serviced by the buffer and procedure cache.

TIP Your Buffer Cache and Procedure Cache percentage should always be at 90 percent or higher. If this number is consistently lower, you should analyze your memory settings and add more memory if necessary.
How do I use the Disk Busy chart?

The Disk Busy chart can be used to analyze bottlenecks in your file system.

**TIP**
If your Disk Busy is consistently over 90 percent you may need to either use a faster disk drive, move files to an additional disk or server, or add disks to a RAID array to improve your SQL Server performance.

How do I use the SQL Server Physical I/O chart?

The SQL Server Physical I/O chart allows you to view the number of pages read from, and written to, the physical disk.

The following reads and writes are displayed on the SQL Server Physical I/O chart:

**Checkpoint Writes**
Saves time during recovery by creating a point at which all dirty pages are guaranteed to have been written to disk.

**Lazy Writer Writes**
Clears unnecessary buffers to create more available buffers to be used by the system.

**Read Ahead Reads**
Are a part of a process in which SQL Server predicts what pages a session will require next and writes them to memory in advance.

**Synchronous Reads**
Are disk writes where the requesting session must wait for the disk write operation to finish.

**Synchronous Writes**
Are disks writes where the requesting session must wait for certain operations to finish.

**TIP**
If the number of reads and writes approaches the limits of the I/O capacity of the computer the SQL Server instance resides, consider reducing the values by tuning your application or database, increasing the I/O capacity of your hardware, or adding memory.
The CPU tab displays key CPU statistics that are updated according to your refresh interval. These statistics can help you determine the cause to any performance bottlenecks that are occurring on your SQL Server instance.

**What charts are on the CPU tab and what do they represent?**

The following charts are available in the CPU tab:

- **CPU Usage**
  
  Displays the percentage of a CPU consumed by SQL Server. On multi-processor computers, CPU time used by SQL Server is derived from the processors available for use by SQL Server.

- **Processor Queue Length**
  
  Displays the number of threads in queue for processor time on the computer that hosts the monitored SQL Server instance.

- **Processor Time**
  
  Displays the percentage of elapsed time that is used by processor executing code in privileged mode.

- **Call Rates**
  
  Displays the rate of SQL Server batch statements, SQL compiles into procedure cache, and SQL Server recompiles of statements stored in the procedure cache.
The Memory tab displays the dynamic memory currently in use by SQL Server as well as how that memory satisfies the requirements of SQL Server components. For example, you can see what percentage of the total allocated memory is currently used by the procedure cache. You can control the amount of memory SQL Server uses by either setting a fixed memory value or setting minimum and maximum memory usage values.

Static structures, like procedure headers, use the amount of memory derived from subtracting the sum of the graphed sections displayed on the Memory tab from the total allocated memory, also provided on the Memory tab.

**What charts are on the Memory tab and what do they represent?**

The following charts are available on the Memory tab:

- **Memory Usage**
  - Displays the SQL Memory Usage by what is used by SQL Server compared to what is allocated and the total used.

- **Paging**
  - Displays the memory pages per second.

- **Memory Areas**
  - Displays the memory usage of the Buffer Cache, Procedure Cache, Connections, Locks, Database, and Optimizer.

- **Cache Hit Rates**
  - Displays the hit rates of the Buffer and Procedure cache.

- **Buffer Cache**
  - Displays the buffer cache used by Free and Active pages.

- **Procedure Cache**
  - Provides the amount of allocated memory used by dynamic and static stored procedures. Included in this value is the memory used by server side cursors and plans. From the total memory allocated to SQL Server, the pinned tables, open objects, connections, locks, and free buffer values are subtracted. The remainder is divided between procedure and data cache.
Get disk performance details

The Disk tab allows you to view key statistics on the way your disks are used on the computer hosting your SQL Server instance. This allows you to analyze your disk performance for any performance bottlenecks that may be occurring.

Disk information is displayed in four charts that display the drives the computer on which the SQL Server instance is located. You can pick which four charts you view by selecting from the drop-down list at the top of each chart. You can select from the following charts:

- **Disk Busy Total**
  Displays the percentage of time during the sample interval that the disk was not idle for all the disks (combined) on the server that the SQL Server instance resides.

- **Disk Busy Per Disk**
  Displays the percentage of time during the sample interval that each disk was not idle.

- **Average Disk Queue Length Total**
  Displays the average length of the disk queue during the sample interval for all disks (combined) on the server that the SQL Server instance resides.

- **Average Disk Queue Length Per Disk**
  Displays the average length of the disk queue during the sample interval for all disks (individual) on the server that the SQL Server instance resides.

- **SQL Server Physical I/O**
  Displays the number of read and write requests sent to the physical disk the SQL Server instance resides.

- **SQL Server Physical I/O Errors**
  Displays the number of errors sent from the physical disk the SQL Server instance resides.

- **Disk Milliseconds / Reads Per Disk**
  Displays the average length of time, in milliseconds, for a disk to read during the sample interval.

- **Disk Milliseconds / Writes Per Disk**
  Displays the average length of time, in milliseconds, for a disk write during the sample interval.

- **Disk Milliseconds / Transfers Per Disk**
  Displays the average length of time, in milliseconds, for a disk transfer during the sample interval.

- **Disk Reads / Seconds Per Disk**
  Displays the number of reads per disk per second during the sample interval.

- **Disk Writes / Seconds Per Disk**
  Displays the number of writes per disk per second during the sample interval.

- **Disk Transfers / Seconds Per Disk**
  Displays the number of transfers per disk per second during the sample interval.

**TIP** OLE Automation must be enabled to show the information in the charts on the Disks tab.
The Procedure Cache view allows you to view each item cached, the size, and the SQL text associated. Graphs below the cache listing provide you with a breakdown of the key Procedure Cache statistics that can help you analyze performance bottlenecks on your SQL Server instance.

The Usage column displays the number of times the associated cache object has been used since inception. The References column displays the number of other cache objects referencing the associated cache object, with "1" as the base.

What are the charts on the Procedure Cache tab and what do they represent?

The following charts are available on the Procedure Cache tab:

**Cache Size by Object Type**

Displays the amount of the procedure cache used by various objects including:

- Adhoc Query
- Check Constraint
- Default Constraint
- Extended Procedure
- Prepared Statement
- View
- Stored Procedure

**Object Types**

- Use the drop-down button to select either Hit Ratio, Object Count, or Use Rate.
- Hit Ratio offers a chart of the cache hit ratios of the individual object types in memory through the last 25 refreshes. For example, you can view the hit ratio of the extended procedure object.
- Object Count displays the number of objects using the procedure cache. Use Rate displays the rate at which each of the object types is used in the procedure cache.
The File Activity allows you to view the relative activity of your database files, on a file-by-file basis. Each database file is listed in a column that represents the disk or mount point the files reside on. At the top of each column the overall Reads, Writes, and Transfers per second are listed. Click on an individual file to expand it and view a graph and the activity data specific to that file.

For each database file, the following information is provided:

**Reads and Writes Per Second**
- The numbers listed are the average number of reads and write operations per second over the period between collections. Collection times are configured on the Console Options window (Tools > Console Options).

**Activity lights**
- Read (red) and Write (blue) lights flash with activity.

**Activity graph**
- A bar graph under the file name displays the recent trend of reads and writes per second. This graph expands when you double-click the file. The expanded Activity graph displays the activity trend for the amount of time specified on the Console Options window (Tools > Console Options).

**Activity data**
- The arrows and the associated data represent the amount of data activity occurring on the file and whether the amount of activity is rising or falling relative to the previous data point.

**TIP**
- Right-click a disk column to Expand/Collapse all files for a database, Expand/Collapse all files on a disk, Navigate to Disk view, or Navigate to the Disk Files view.

**How do I access the File Activity window?**
- To access the File Activity window, select File Activity under Resources in the Servers tree.

**What are Database Files + Remaining Disk Files?**
- Database Files + Remaining Disk Files is a chart that depicts the activity of all the non-SQL database files and files not included in the File Activity window.

**How do I use the filter to display the database activity that is important to me?**
- The filters at the top of the File Activity window allow you to limit the amount of data to that data which is most important to you.

**TIP**
- For an accurate filter, use the percent (%) wildcard character in the File Path Like or File Name Like field if you do not include the entire name.

**Filter Drive**
- Use the Drive drop-down to select the drives to display on the File Activity window. You can also enter a file path in the File Path Like field to select a subset of drives. This field accepts the percent (%) wildcard character.

**Filter Database**
- Use the Database drop-down to select the databases to display on the File Activity window. You can also enter a file name in the File Name Like field to select a subset of databases. This field accepts the percent (%) wildcard character.
File Type

Check the boxes of the file types to display on the File Activity window.

Sort By

Select to sort by File Name, Reads Per Second, Sort Direction, or Writes Per Second.
A wait is what happens when a SQL query is able to run but is waiting on another resource. By examining the wait statistics on your SQL Server instance, you can narrow down the possible causes of the waits and identify the best way to increase the performance of your system.

SQLdm includes the following two windows that allow you to view your server waits:

- Server waits
- Query waits
The Server Waits view allows you to see all the waits that are affecting your server’s performance. By correlating the areas of the SQL Server instance that are waiting with other statistics, it is possible to narrow down possible causes of the waits. All waits occurring on your system are displayed in a graph that shows the time the wait happened and the length of time it took for the wait to clear. The graph can be filtered by the wait type to allow you to view how a particular wait is affecting your server. A table appears below the graph that displays the wait category, wait type, length of time the wait took to clear and a definition of the wait type.

**How often are waits collected?**

Server waits are always collected. History mode for this view is available.

**Which wait types can I pick for the Server Waits view?**

You can filter the view by Backup, I/O, Lock, Memory, Non-I/O Page Latch, Non-Page Latch, Transaction Log, and Other categories.

**Which categories of waits can I choose to limit the view?**

You can select from the following wait categories:

- **Total Waits**
  - Total waits include all the types of waits collected.

- **Signal Waits**
  - Signal waits occur when a resource is now available (after a resource wait) and the task is waiting to run again.

- **Resource Waits**
  - Resource waits occur when access to a resource is denied because the resource is in use or not available.
The Query Waits view displays the queries on your SQL Server instance with the longest wait times. By default, the chart at the bottom of the view displays the query waits over time, and allows you to further investigate based on Statements, Applications, Databases, Clients, Sessions, and Users.

By analyzing these waits you can better determine where your biggest bottlenecks are occurring and what changes could potentially have the biggest performance boost on your SQL Server instance.

How can I use the Query Waits chart?

You can select to view query waits as the waits appear over time or by duration of the wait, color-coded by query. The chart includes each of the following wait type categories: Backup, I/O, Lock, Memory, Non-I/O Page Latch, Non-Page Latch, Transaction Log, and Other.
SQLdm allows you to monitor your database backups, restores, table sizes, table row counts, configurable options, and replication status. SQLdm provides a graphical analysis of the used, unused, and index disk space of the database. You can also view the size and structure of database tables. SQLdm provides a list of data size, text or image size, index page size, and the number of table rows, making reorganization and index creation decisions easy.

What common database concepts are important to understand?

The following concepts are important to understand when using the database tab:

**Percentage of Database/Log Full**

The % Data Full and %Log Full columns on the Database Summary and Database Files views refer to the current space used as a percentage of the potential maximum available space. The larger the number the closer you are to running out of space. These columns can have associated alert thresholds.

**Potential Maximum Available Space**

The Data Potential Growth and Log Potential Growth columns on the Database Summary and Database Files views refer to the currently allocated space plus the available expansion space.

**Available Expansion Space**

Available Expansion Space refers to the available disk space that files can grow into as determined by the growth settings for the files, the location of the files, and the free space on those disks.

Multiple database files as well as files from other databases can potentially compete for space on the same disk. For example, if the data and log files for several databases are on the same disk and the data file of one database expands, the available expansion space for all database files on the disk is immediately reduced.

What alerts are associated with the Database tab?

**Table Fragmentation (Percent)**

Percentage of pages that are fragmented in each table in the database.

**Database Full (Percent)**

Percentage of the allowable disk space for the database currently used by the sum of data, text, and index. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings, and available disk space.

**Log Full (Percent)**

Percentage of the allowable disk space for the database currently used by the transaction log. Allowable disk space for the database is calculated by taking into the current allocated space, auto-growth settings and available disk space.

**Database Read/Write Error Occurred**

Failure to read from or write to the hard disk.

**Database Status**

Multiple types of Database Status alerts exist.

**Index Row hits (Percent)**

Number of row hits that part of a clustered index or an entire non-clustered index returns, providing an indication of index effectiveness.
Get the database status summary

The Database Summary tab allows you to view the status of all the databases on the selected SQL Server instance. Along with the database status table, the Summary tab includes charts that provide options for viewing the capacity usage and recent trends of your databases. You can select multiple databases in the list to compare several databases.

**TIP** The status column can help you quickly identify databases that have associated alerts and need your attention.

**What data is available on the Database Summary tab?**

The Database Summary tab lists each database on the monitored SQL Server instance. The Database Summary tab displays a large variety of database statistics and is an important way to get an overview of the health of your databases. This information gives you a good indication as to how the data is configured in your database, such as comparing the volume of your table data compared to your indexes, or the size of your log files.

**What charts can I view on the Database Summary tab?**

The Database Summary tab includes charts that display capacity usage and recent trend information for your database.

**Capacity Usage**

You can choose to view how data is used on your databases, displayed in megabytes. You can also view how the log is used, viewed in megabytes or percentages. For more information, click inside the chart to open the Database Files tab.

**Recent Trends**

You can choose to view how your database or group of databases are used currently by selecting either active sessions, transactions per second, data size, log size, or log flushes from the drop-down list.
The Tempdb Summary tab allows you to view the status of your tempdb database on the selected SQL Server instance. These charts include options for viewing your current capacity usage and recent trends of your files over time. These statistics appear in the Database Statistics report.

**What data is available on the Tempdb Summary tab?**

The Tempdb Summary tab displays a list of sessions currently using tempdb along with their cumulative usage and tempdb space. These statistics help you get an overview of the health of your tempdb database and gives you a good indication whether you have enough space allocated or whether tempdb issues are causing a performance problem on your server.

**What charts can I view on the Tempdb Summary tab?**

The Tempdb Summary tab includes charts that display capacity usage and recent trend information for your tempdb database.

Toggle between the Version Store Cleanup and Tempdb Contention charts by using the drop-down list available to the right of the chart title.

**Tempdb Space Used by File**

You can choose to view how each file is using space on your tempdb database, displayed in megabytes.

**Tempdb Space Used Over Time**

You can choose to view how your database is used over time based on object type by comparing charts that offer metrics for the Space Used and Space Allocated.

**Version Store Cleanup Rate**

The tempdb database version store stores data rows necessary to support snapshot isolation. This chart helps you see the current cleanup rate to avoid filling up tempdb.

**Tempdb Contention**

The Tempdb Contention chart displays latch wait time (in milliseconds) for the allocation pages of tempdb. The three tracked allocation page types are page free space (PFS), global allocation map (GAM), and shared global allocation map (SGAM). Latch contention of this sort is usually an indication that more tempdb data files need to be created. In some situations, using Trace Flag 1118 may also alleviate tempdb contention.
SQLdm allows you to monitor your tempdb database with a number of charts and alerts to help you avoid costly performance issues caused by a full tempdb. The tempdb system database is a workspace used to hold temporary user objects, results created through queries and sorts, and other objects created by the SQL Server Database Engine. Because of the large amount of data stored in tempdb, users can run out of disk space, which causes the entire server and all of its databases to become unresponsive. SQLdm seeks to resolve some of these issues by providing a monitoring solution that features a series of tempdb-specific charts, views, and alerts.

The tempdb system database, along with the Master, Model, and MSDB databases, is provided by default with SQL Server and is a shared resource available to all users connected to a single SQL Server instance. Each time you start SQL Server, it re-creates tempdb based on the Model database. Tempdb can fill up quickly when you are low on disk space or have a low maximum size for database growth. Certain workloads may cause excessive space usage or create contention in tempdb, which can affect performance on the entire server.

**TIP**
Tempdb monitoring is supported for SQL Server 2005 and above users only.

**What is the tempdb version store?**

The tempdb version store collects the data necessary to support row versioning. Each time a data value changes, a new version of the row is created and stored for as long as the oldest active transaction needs to access it. Once the row version is no longer needed it is removed from tempdb by a cleanup job which runs once per minute. As a result, long-running transactions prevent cleanup of older entries into the tempdb version store, causing growth which can affect performance and cause tempdb to run out of space.

**What is tempdb contention?**

Tempdb resource contention or waits is usually the result of heavy use on too few tempdb files and occurs when the system attempts to allocate pages. The tempdb contention chart displays latch wait time (in milliseconds) for the allocation pages of tempdb. The three tracked allocation page types are page free space (PFS), global allocation map (GAM), and shared global allocation map (SGAM). Latch contention of this sort is usually an indication that more tempdb data files need to be created. In some situations, using Trace Flag 1118 may also alleviate tempdb contention.

**What alerts are available for me to monitor my tempdb?**

Tempdb includes the following specific alerts:

- Data File Autogrow
- Log File Autogrow
- Longest Running Version Store Transaction (Minutes)
- Session Tempdb Space Usage (MB)
- Tempdb Contention (ms)
- Version Store Generation Ratio
- Version Store Size (MB)
Monitor database configuration settings

The Database Configuration view allows you to view the configuration settings of the databases on the selected SQL Server instance. The Database Configuration view contains a single grid that includes all the configuration settings for the databases on the selected SQL Server instance.

**TIP** The columns displayed on the Database Configuration view differ between SQL Server 2005 and SQL Server 2008 instances.

**What can I use the Database Configuration view for?**

The Database Configuration view can be used to analyze configuration differences between databases or databases with incorrect configuration settings.

**TIP** If you are having database issues, the Database Configuration view can help you determine if the problem is caused by configuration errors.

**What causes the Collation and Version columns to report null values?**

If the database is not started and `AUTO_CLOSE` is set to `ON`, these columns will be set to null until the database is started.
View Database Status

SQLdm allows multiple simultaneous status indicators for each individual database, very similar to what you see in Microsoft SQL Server Management Studio. You can see the status of a database in the Servers tree by expanding the associated Server tree. The status displayed matches what is shown in Microsoft SQL Server Management Studio in all cases where Management Studio displays a status.

What are the possible alerts or statuses for my database?

The following alerts and statuses are associated with Database status:

<table>
<thead>
<tr>
<th>Default Status Level</th>
<th>Alert/Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Normal</td>
<td>Everything is okay and the database is operating normally</td>
</tr>
<tr>
<td>OK</td>
<td>Standby</td>
<td>Database is in standby mode</td>
</tr>
<tr>
<td>OK</td>
<td>Read Only</td>
<td>Database is in read-only mode</td>
</tr>
<tr>
<td>OK</td>
<td>Single User</td>
<td>Database is placed in single-user mode</td>
</tr>
<tr>
<td>OK</td>
<td>Restoring Mirror</td>
<td>Restoring database mirror</td>
</tr>
<tr>
<td>OK</td>
<td>Undetermined</td>
<td>No information is available on the database</td>
</tr>
<tr>
<td>OK</td>
<td>Cleanly Shut-down</td>
<td>Database is shutdown and does not need recovery</td>
</tr>
<tr>
<td>OK</td>
<td>DBO Use Only</td>
<td>Database is in a dbo-use mode (restricted_user)</td>
</tr>
<tr>
<td>Warning</td>
<td>Restoring</td>
<td>Restoring database</td>
</tr>
<tr>
<td>Warning</td>
<td>Recovering</td>
<td>Recovering database</td>
</tr>
<tr>
<td>Warning</td>
<td>Recovery Pending</td>
<td>Recovery has not yet begun</td>
</tr>
<tr>
<td>Warning</td>
<td>Offline</td>
<td>Database is offline</td>
</tr>
<tr>
<td>Critical</td>
<td>Suspect</td>
<td>Database integrity is suspect</td>
</tr>
<tr>
<td>Critical</td>
<td>Emergency Mode</td>
<td>Database is in Emergency Mode</td>
</tr>
<tr>
<td>Critical</td>
<td>Inaccessible</td>
<td>SQLdm monitoring service is unable to open the database. This could be caused by a serious issue such as disk failure.</td>
</tr>
</tbody>
</table>

How do I change the default alert status level?

Alerting is based on the most critical status associated with a database at the time of collection.
TIP Note that for certain metrics, using the informational alert means that you will no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.

To change the default alert status level:

1. Right-click any server in the Servers tree.
2. Select Configure Alerts.
3. Select Database Status from the list of alerts.
4. Double-click any of the values on the Configuration tab.
5. Edit each alert to associate it with the desired alert status level.
Analyze database files

The Files view allows you to see the file statistics for each of your databases on the selected SQL Server instance. Database capacity and growth are common causes of SQL Server performance bottlenecks.

**TIP** Database growth statistics are collected every scheduled refresh.

**How can I use the Files view?**

The Files view is an important tool that can be used to see how much disk space you have available to your databases, how much disk space they are using, and the percentage of space used by databases and logs. Database capacity and growth are common causes to SQL Server performance bottlenecks.

For Example: If the percent data full on a database is over a threshold it could mean that either the file doesn't have the file auto-growth setting turned on, is about to reach the maximum allowable size, or that the disk the database is on is about to run out of space.

Another common use of the Files view is to identify the fragmentation percentage on an index. Once identified, you can click Rebuild Indexes to reduce the fragmentation on the selected index.
Get file statistics

Select any database in the list to display the associated file statistics. Click any file in the list to populate the information displayed in the File Usage chart.

Select the drop-down box above the chart to choose a chart type:

- Current Megabytes
- Current Percent
- Potential Megabytes
- Potential Percent

**TIP** You can select multiple databases as well as multiple files to compare file statistics.
The File Groups tab displays the selected database. Click a group on the File Groups tab to populate the File Group Usage chart. Select the drop-down list above the chart to choose a chart type:

- Current Megabytes
- Current Percent
- Potential Megabytes
- Potential Percent

**TIP** Select multiple databases and multiple file groups to compare how they are used. This can help you quickly spot issues with your file usage.
The Transaction Logs tab allows you to view the transaction log files for the selected databases. Select the drop-down list above the chart to choose from the following chart types:

- Current Megabytes
- Current Percent
- Potential Megabytes
- Potential Percent

**TIP** Log files in high transaction rate environments can occupy multiple disks. The Transaction Logs tab allows you to easily monitor these configurations.
The Disks tab allows you to view four charts simultaneously, that display the drives the computer on which the SQL Server instance is located. You can pick which four charts you view by selecting from the drop-down list at the top of each chart. You can select from the following charts:

- Disk Busy Total
- Disk Busy Per Disk
- Average Disk Queue Length Total
- Average Disk Queue Length Per Disk
- SQL Server Physical I/O
- SQL Server Physical I/O Errors
- Disk Milliseconds / Read Per Disk
- Disk Milliseconds / Write Per Disk
- Disk Milliseconds / Transfer Per Disk
- Disk Reads / Second Per Disk
- Disk Writes / Second Per Disk
- Disk Transfers / Second Per Disk

**TIP** OLE Automation must be enabled to show the information in the charts on the Disks tab.

What do I need to do if I have configured OLE Automation to run out-of-proc?

You will need to edit specific registry settings to allow WMI to run out-of-proc.

To edit these registry settings, perform the following steps:

1. In the cd image of the SQLdm install, open the Scripts folder.
2. Copy the SQLdmOutOfProcOleAutomation.reg file onto each of your monitored SQL Server instances.
3. Run this file on each of your monitored SQL Server instances.
The Backups & Restores tab allows you to view the backup and restore history of each database on the selected SQL Server instance. You can view the backup and restore history of a database or a group of databases. The history displays in the Backup/Restore History section of the tab. The Backup/Restore section shows the date and time of the backup or restore, the user that initiated it, and the size and path of the backup or restore file.

How can I use the Backups and Restores view?

The Backups and Restores view is a useful way to view the last successful backup, the recovery model used, and the backup history for each of the databases on the SQL Server instance. To see your backup and restore history, click one or multiple databases to populate the Backup and Restore History table.

TIP The default filter setting shows a 31-day history. For less frequently run jobs such as quarterly or annual backups, it may be necessary to increase the Days To Show setting in the Filters window.
The Tables & Indexes tab allows you to view statistics for all the tables and indexes on a particular database. To see all the tables and indexes for a database, use the database drop-down list to select your database.

Along with displaying statistics, the Tables & Indexes tab includes the ability to update your statistics and rebuild your indexes on the selected table. These options are available by right-clicking a table and selecting the appropriate option from the context menu. To perform these actions on multiple tables, use the buttons on the ribbon.

**TIP**
Table growth statistics are collected on the schedule configured on the Table Statistics tab of the Monitored SQL Server Properties window. This includes the table fragmentation and size statistics.

### What can I do on the Tables and Indexes view?

The Tables and Indexes view provides a good way to view the overall health of your tables. In addition, the Tables and Indexes view allows you to rebuild the index for any of your tables that have a high Fragmentation percentage.

**TIP**
The Rebuild Indexes function is enabled for User tables. System tables cannot have indexes rebuilt.

### How can I use the Table Details section?

The Table Details section of the Table and Indexes view allows you to view various details on the selected table. To update the statistics for your tables, click Update Statistics at the top of the view. The following tables are available in the Table Details section:

- **Size**
  - Displays the size of the data, text, and indexes associated with the selected table.

- **Dependencies**
  - Lists other database objects that the selected table contains references to and other database objects that reference the selected table. In addition to the object name, the type of object and action associated is listed.

- **Indexes**
  - Lists the associated indexes and index information for the selected table.

- **Index Statistics**
  - Lists the number of columns, average length, average row hits, and row hits percentage for each column in an index associated with the selected table. It will also list the data distribution data values and occurrences of each value.

### What information does the Table Size Details chart contain?

The Table Size Details chart displays the size of each component in a table. To view the Table Size Details chart, select a table from the list and click the Size tab in the Table Details section.

### How can I analyze table dependencies?

The Table Dependencies tab displays the objects the selected table references and is referenced by, the type of dependency, and the action of the dependency. This allows you to quickly see how the table is utilized and how it is affected by and affects other tables and objects.
What information is on the Indexes tab?

The Indexes tab displays all indexes defined for the selected table and an overview of each index.

How can I analyze index statistics?

The Tables and Indexes view allows you to see general table and index statistics for the database selected. You can also rebuild indexes for highly fragmented tables and update your statistics, which are displayed in the chart and tables in the Table Details section of the view. For more information, see SQL Server Books Online.

TIP

Fragmentation data for a particular table may not be collected for any of the following reasons:

- Table statistics collection has not occurred
- The database hosting the table has been excluded from collection
- The table size does not meet the minimum size threshold
- The database hosting the table is not accessible
- The table is locked
The Update Statistics option on the Tables and Indexes view of the Databases tab allows you to update your Table Statistics information. When you click Update Statistics from the SQLdm Console, the Select Tables window opens and allows you to choose the tables to update.
The Rebuild Indexes option on the Tables and Indexes view allows you to select tables to rebuild. Rebuilding your tables allows you to decrease the amount of fragmentation on a table. High table fragmentation can lead to overall performance bottlenecks on your SQL Server instance.

When you click Rebuild Indexes from the SQLdm Console, the Select Tables window opens and allows you to choose the table indexes to rebuild.

**TIP** If you are rebuilding indexes on a very large table the rebuild process could take several minutes to complete. While the rebuild is in process, the table is locked and users are unable to access it.

Once the indexes are rebuilt, the % Fragmentation (Current) column displays showing you the results.

**TIP** If you are rebuilding a very large index, the SQLdm Console will check for the rebuild status for 90 seconds. If the rebuild is not complete after 90 seconds, the % Fragmentation (Current) column does not display the new value until the next table statistics collection occurs.
Monitor database mirroring

The Database Mirroring view allows you to monitor the mirrored databases set up on monitored SQL Server instances. You can view the mirrored databases by clicking Mirroring on the Databases tab.

**TIP** If you have enabled database mirroring on your SQL Server, the SQLdm Database Mirror Monitoring view is enabled by default.

**What is database mirroring?**

Database mirroring is a process available in SQL Server 2005 and 2008 that moves database transactions from one SQL Server database to another SQL Server database on a different instance of SQL Server. Database mirroring allows you to configure a duplicate of your important databases for availability and redundancy.

**What database mirroring statistics can I monitor in the Database Mirror Monitoring view?**

In this view, you can see all the mirrored databases on all the monitored SQL Server instances, their current status, and history.

The database table provides the following information:

- **Database Name**
  - The name of the database.

- **Server Instance**
  - The SQL Server instance hosting the mirrored database.

- **Current Role**
  - Describes whether the database is the principal or the mirrored database.

- **Mirroring State**
  - The status of the database mirror, color-coded based on the configured alert level.

- **Witness Connection**
  - The status of the connection, color-coded based on the configured alert level.

- **Operational State**
  - Describes whether the database mirroring session is operating in a preferred configuration based on the user’s configuration of SQLdm.

- **Operating Mode (Hidden)**
  - Shows the current operating mode: High availability with automatic fail-over (has a witness, synchronous); High availability without automatic fail-over (no witness, synchronous); High performance (no witness, asynchronous)

**What actions can I take in the Database Mirror Monitoring view?**

Each row of the mirrored databases table provides a menu of actions you can perform:

- **Failover**
  - Allows you to fail over the principal role between the two partner databases in the mirroring session.
The duration of a mirroring failover operation varies according to the amount of log in the redo queue on the mirror. All transactions in the redo queue must roll forward before the mirror can assume the role of Principal. After the fail over has been initiated, the Principal changes to PENDING_FAILOVER while the mirror synchronizes. Once complete, there is a brief period where user connections are terminated and while roles are switching the mirroring status is undetermined. If a refresh occurs at this time, your mirrored database does not show in the Mirrored Databases grid. This lasts for a moment and your next refresh reflects the switched roles.

For more information about the failover function, see SQL Server Books Online.

**Suspend/Resume Mirroring**

Allows you to pause all mirroring operations. While mirroring is paused in "High Availability" mode, all transactions opened against the principal are queued for transmission to the mirror partner once the session resumes.

**Set mirror relationship as "Normal" or "Failed Over"**

Allows you to define the status of a mirror relationship. Once you have set a preferred operational state, you will be alerted if the configuration differs from this.

**Clear the role preference for this mirroring session**

Allows you to clear the preferred role for the session and stop receiving alerts for differences between the current configuration and the Normal status.

What alerts are available for mirrored databases?

The following alerts can be configured for mirrored databases.

**Mirror Commit Overhead**

Average delay in milliseconds on the principal server instance for each transaction due to database mirroring.

**Mirrored Server Role Change**

A change in role has taken place on a mirrored database on the monitored server.

**Mirroring Oldest Unsent**

The age of the oldest unsent transaction in minutes on the principal. This is only meaningful on the principal server instance.

**Mirroring Preferred Configuration**

A change has taken place in terms of the preferred operational status.

**Mirror Status**

Status of the mirrored database such as synchronized, suspended or disconnected.

**Redo Queue Length**

Size of the redo queue on the mirror in kilobytes. The redo queue is the size in kilobytes of the outstanding transactions on the mirror that have yet to be applied.

**Unsent Queue Length**

Size of the unsent log in the send queue on the principal in kilobytes.

**Mirroring Witness Connection**

Is the witness connected or not (if there is a witness configured).
Monitor services

The Services tab allows you to view the status and manage your SQL Server services including the following services:

- SQL Server Agent
- Full-Text Search. SQL Server 2008 no longer has a separate full-text search service. If the monitored instance is SQL Server 2008, the full text search does not appear on the grid or graph.
- DTC (Distributed Transaction Coordinator service)
- Replication services

**TIP**
The Start and Stop service actions are not supported for virtual SQL Server instances (instances located on a Windows cluster node). To start or stop a service on a virtual SQL Server 2008 instance, use the Microsoft Failover Cluster Management tool. For SQL Server 2005 or earlier, use the Microsoft Cluster Administrator tool to manage services.

What alerts are associated with the Services tab?

- **Non-Distributed Transactions (Count)**
  The number of transactions not written to the Distribution database.

- **Last Full-Text Catalog Update (Hours)**
  The number of hours beyond which the Full-Text Catalog statistics data is considered outdated.

- **Unsubscribed Transactions (Count)**
  The number of transactions written to the Distribution database that are not subscribed.

- **Unsubscribed Transactions (Seconds)**
  The number of seconds a transaction written to the database waits before subscription.

- **DTC Status**
  Displays the status of the Distributed Transaction Coordinator service.

- **Full-Text Search Status**
  Displays the status of the Full-Text Search service.

- **SQL Server Agent Status**
  Displays the status of the SQL Server Agent service.

- **SQL Server Status**
  Displays the status of the SQL Server service.

- **SQL Server Agent Long Running Job**
  Indicates the percentage of time a scheduled job is taking over what it normally takes to complete the job.

- **SQL Server Agent Job Failure**
  Indicates that a scheduled job processed by the SQL Server Agent service ended abnormally.

- **CLR Enabled**
  Indicates whether the Common Language Runtime (CLR) feature enabled on the SQL Server.
**Cluster Active Node**

The active node of the cluster is a non-preferred node. The preferred node may be configured under Server Properties > Cluster Settings.

**Cluster Failover**

There has been a change in the active node. This alert remains active for the amount of time specified in the alert's advanced settings.

**Distribution Latency**

Time in seconds that a replication transaction has been holding at the Publisher and waiting to be received by a Distributor.
Get an overview of your services performance

The Services Summary tab includes the status of your Services. You can start and stop your DTC, or Full-Text Search by selecting the appropriate button.

**TIP** SQL Server 2008 no longer has a separate full-text search service. If the monitored instance is SQL Server 2008, the full text search does not show on the grid or graph.

**How do I use the Service Availability chart?**

The Service Availability chart displays the DTC (Distributed Transaction Coordinator), Full-Text Search, SQL Server, and SQL Server Agent status.

**TIP** An Unavailable status in the chart is caused by a pending status change such as starting, stopping, pausing, and continuing.
Monitor SQL Agent jobs

The SQL Agent Jobs view allows you to view all your SQL Agent Jobs for the selected SQL Server instance. You can view the Job History of SQL Agent Jobs in the Job History list and Job Steps by clicking the + button next to the Job Name.

How do I access the SQL Agent job list?

To access the list of SQL Agent jobs associated with a server, on the Servers navigation page, click the SQL Server instance you want to view, and then click Services > SQL Agent Jobs.

What do the colors in the Last Run Outcome mean?

The Last Run Outcome column of the SQL Agent Jobs page is color coded to quickly alert you of issues needing your attention. The colors used in this column depend on the use of the SQL Agent Job Completion metric in the Alert Configuration dialog.

If this metric is enabled, the Last Run Outcome column is colored based on the correlation of the outcome of the job or job step and the state that outcome corresponds to in the SQL Agent Job Completion metric. If this metric is not enabled, then SQLdm displays results in red if the outcome is Failed or Canceled, and green if the outcome is Succeeded. All other outcome states remain the default color.

To customize the Last Run Outcome settings for your SQL Agent jobs:

1. Right-click a SQL Server instance in the Servers tree.
2. Select Configure Alerts.
3. Select the SQL Server Agent Job Completion metric.
4. Click the Edit button.
5. Make the necessary changes to customize your alert states.
6. Click OK, and then click OK.

How do I view message information about a specific job?

Use the Job History list to see more information about a specific job. You can review the full message by right-clicking the appropriate job name, and then selecting View Message. A copy feature allows you to cut and paste the message detail into another format for additional use.
Monitor full text searches

The Full-Text Search view can be used to monitor and diagnose problems with the SQL Server Full-Text Search service. The view is composed of a single list that displays information about each Full-Text catalog maintained on a monitored SQL Server. You can view Catalog details by selecting a Full-Text Catalog from the list.

The Full-Text Search view lists all the full-text search catalogs that are hosted on the SQL Server instance as well as their statuses. You can also start or stop the Full-Text Search service, optimize or rebuild the catalogs.

For each catalog listed, the database, status, size, and last population date is listed. In addition, selecting any catalog from the list will populate the Catalog Details section. This section includes information on the tables, rows, columns, catalogs, language, and column data types included in the Full-Text search.

**What happens when I Optimize or Rebuild my Full-Text Catalog?**

**Optimize**

When you select the Optimize button, SQL Server will optimize the space utilization of the catalog and improve query performance as well as the accuracy of relevance rankings of search results.

**Rebuild**

When you select the Rebuild button, SQL Server repopulates the full-text search index.
Analyze replication information

The Replication view allows you to analyze the replication information on the selected SQL Server instance. The view displays the topology and key metrics that detail the current state of active subscriptions and non-subscribed subscriptions on all SQL Server instances related to replication on the monitored SQL Server.

For more information about Replication, see SQL Server Books Online.

**TIP** Collecting replication statistics can have performance impact on the monitored SQL Server instance.

**Replication Topology**

The Replication topology grid displays all replication sessions in which the selected server is participating, whether it is a publisher, distributor, or subscriber. For merge and snapshot replication, SQLdm must monitor the distributor. For transactional replication, it is recommended that you monitor the distributor, publisher, and subscriber for a complete set of metrics. Note that SQLdm displays a subset of data for each monitored participant. Right-click a row or session to navigate to participants for more information.

**Detailed Overview**

The Detailed Overview tab provides an overview of transactional replication specific to the publisher, distributor, and subscriber. In addition, SQLdm also displays an overview of the selected merge replication session gathered from the distributor.

**TIP** Times displayed in the subscriber grid are local to the subscriber.

**Non-Subscribed Queue**

The Non-Subscribed Queue tab contains the Entry Date, Subscription Database, Wait Time, and Command contained in the queue of active non-subscribed commands. Once a command exceeds the retention period, SQLdm switches the command to inactive and no longer displays the detail.
Monitor logs

The Logs tab allows you view SQL Server related logs. These logs can be used to diagnose SQL Server problems where a history of events is necessary to form context.

By default the current log is selected; you can select archived or even multiple archived logs to view or search. Filter and Search options allow you to locate the areas of the logs that are most important.

**TIP** The filter option will only filter logs once the logs alerts are enabled.

The archive number is the file number associated with the log in SQL Server while the date is the last modified date of the file on disk. If the archive numbers seem out of order, you should check the error log.

**What can I use the Logs tab for?**

The Logs tab allows you to look at all the events taking place on your SQL Server instance. Each event, when selected, contains detailed information in the Details section of the tab which allows you to diagnose specific issues that occur.

For Example: if the one of your SQL Server instances restarts, the event log can tell you the time of the restart. This helps you diagnose issues that occur on your SQL Server instances.
The Filter Settings window allows you to select:

- The time range to filter events, such as Start Date and End Date
- The source of the log entry, such as the session ID number
- The message severity, such as OK, Warning, or Critical

If the alert is disabled, all messages are informational.

**To Filter the Server Log:**

1. Click the Filter button on the Logs tab.
2. Enter an End Date and Start Date in the General section of the Filter Settings window.
3. Select the Message Severity.
4. Click OK.
The Find window, accessed by clicking Search, allows you search for text in the event messages.

To search for text in your Server logs:

1. Click Search on the Logs tab.
2. Enter the search term in the Find what field.
3. Select the field you want to search (Message, Msg #, or Source) from the Look in drop-down list.
4. Select your Find options.
5. Click Find Next.
6. Keep clicking Find Next to scroll through the matches to locate the specific event.
Click the Cycle Server Log button to archive what is in the most recent (current) log and start a new log.

**To cycle the Server Logs:**

1. Click Cycle Server Log.
2. Select the log types you want to cycle (SQL Server or SQL Server Agent).
3. Click OK to cycle the logs.
The Configure Logs window allows you to determine the number of SQL Server Logs retained by SQL Server. You can access this window by clicking Configure on the Logs tab.

Check the Unlimited check box to retain all Event Logs or enter the number to keep only a number of the most current logs. SQLdm deletes older logs.

**TIP** The number indicated on the Configure Server Logs window changes the number of SQL Server Logs stored by SQL Server, but not the number of SQL Server Agent logs stored.
SQLdm allows you to configure alerts to inform and warn you about approaching issues with your SQL Server instances. You can view these alerts using the SQLdm Console, the Idera Newsfeed, or SQLdm Mobile.

When an alert threshold is reached, SQLdm can:

- Send the alert message to the Idera Newsfeed Action Provider, which publishes the alert message as a status update on the server wall in the Idera Newsfeed interface and on the Active Alerts view in SQLdm Mobile
- Send an email notification
- Pop up an alert message in your Windows taskbar
- Write an event to the Windows Event log
- Create a new To Do item (Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.)

In response to a set threshold entering the error level, SQLdm can perform one or more of the following actions:

- Send the alert message to the Idera Newsfeed Action Provider, which publishes the alert message as a status update on the server wall in the Idera Newsfeed interface and on the Active Alerts view in SQLdm Mobile
- Send an email notification
- Write an event to the Windows Event log
- Create a new To Do item (Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.)

Once you correct the situation triggering the alert, SQLdm alerts again you if the situation recurs.

**TIP**

On each SQLdm view, columns containing associated alerts are highlighted with their current status color. You can right-click many of these columns and select Configure Alert to change the associated thresholds.

**What is an alert template?**

SQLdm allows you to configure generic alert settings as a template that you can apply to servers and groups of servers in your organization. Click Tools > Alert Configuration Template and complete the required fields to configure an alert template.

**What is an informational alert?**

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQLdm. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

**TIP**

Note that for certain metrics, using the informational alert means that you will no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.

**What is the Idera Newsfeed?**

The Idera Newsfeed is a revolutionary new way for DBAs and managers to collaborate, share knowledge, and keep close tabs on your most critical SQL Server issues. With the Idera Newsfeed, you can quickly share information to stay informed, be more productive and shorten the time to problem resolution.

For more information, see the Idera Newsfeed Help.
What is SQLdm Mobile?

SQLdm Mobile includes the Idera Newsfeed as well as several dashboard views that define your server health and expose critical alerts. The SQLdm Mobile Web application can be accessed from most smart phones and mobile devices.
The Alert Configuration window allows you to set up alerts for specific SQL Server instances. To open the Alert Configuration window with a SQL Server instance selected, right-click the instance and select Configure Alerts. You can set the acceptable thresholds using the following alert type:

<table>
<thead>
<tr>
<th>Alert Type</th>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Green</td>
<td>Acceptable threshold where SQLdm does not generate an alert.</td>
</tr>
<tr>
<td>Informational</td>
<td>Blue</td>
<td>Informational threshold where SQLdm generates an informational alert.</td>
</tr>
<tr>
<td>Warning</td>
<td>Yellow</td>
<td>Warning threshold where SQLdm generates a warning alert.</td>
</tr>
<tr>
<td>Critical</td>
<td>Red</td>
<td>Critical threshold where SQLdm generates a critical alert.</td>
</tr>
</tbody>
</table>

In addition to setting the thresholds for specific alerts, you can customize your alerts by adding comments to each alert. Comments allow you to include instructions to users when a level reaches a particular threshold or to provide additional information about the alert. You can also customize the levels at which SQLdm provides alert notifications by changing the informational, warning, and critical values on the Server Thresholds tab of the Alert Settings window.

You can also base your alerts on the past performance of the metrics collected by SQLdm. This is a powerful and effective way to ensure that the alerts you receive are outside of your typical metric ranges.

**TIP**

For each alert metric, you can enter the specific criteria by double-clicking the values on the Configuration tab.

**To configure an alert:**
1. Right-click a SQL Server instance in the Servers tree.
2. Select Configure Alerts.
3. Select the metric you want to edit from the list in the Alert Configuration window.
4. Check the boxes next to Informational, Warning, and Critical to include alerts for these states.
5. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level.
6. Select the Comments tab and enter information you want displayed in the alert message for this metric.
7. Click Apply.
8. If you want the edits you made to be replicated to every SQL Server instance SQLdm is monitoring, click Yes.
9. Click OK to accept your changes.

**Why are my new Table Fragmentation alert thresholds not immediately effective?**

SQLdm updates Table Fragmentation alerts with new threshold information only when new data is generated during the table statistics collection interval. It is preferred that you set your table statistics collection interval to occur when your server is not performing any production tasks.

Because this data collection occurs at most once per day and you can configure it to run as infrequently as once per week, there may be a significant delay before new alerts are generated with your new alert threshold. You can configure your wait monitoring using the Monitored SQL Server Properties window.

**What is an alert template?**
SQLdm allows you to configure generic alert settings as a template that you can apply to servers and groups of servers in your organization. Click Tools > Alert Configuration Template and complete the required fields to configure an alert template.

**What is an informational alert?**

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQLdm. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

**TIP**

Note that for certain metrics, using the informational alert means that you no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.
Configure alert templates

The Alert Templates window allows you to configure generic alert settings that you can apply to servers and groups of servers in your organization. An event that causes a metric to no longer meet a condition or to exceed a threshold signals SQLdm to raise an alert. Alert templates save you time and help you to avoid mistakes by simplifying alert configuration across your environment. SQLdm displays the default template name and description in bold text.

Applying an alert template to a monitored server overwrites all existing alert configurations. You can make any custom changes to the specific server's alert configuration once the template is applied. If you make a change to a template that is applied to a server, you must also reapply the template to that server using the Apply To button.

**TIP** Servers are not permanently attached to an applied alert template. SQLdm does not propagate to the monitored server any changes made to the template unless explicitly applied.

**To create a new alert template:**

1. Select Tools > Alert Configuration Templates.
2. Click New.
3. Type a unique name and description.
4. If you want to create a new template based on an existing template, select the name of the existing template from the Copy from list.
5. If you want this new template to be your default template, check Default Template.
6. Click OK when done.
7. If you want to apply this new template to your existing SQL Server instances, click Yes in the confirmation message box. If you do not want to apply your new template to any existing SQL Server instances at this point, click No.
8. Configure your alert settings, and then click OK.

**When should I make ad hoc changes and when should I use a template?**

A template allows you to apply the same alert configurations to more than one SQL Server instance. It helps you avoid simple mistakes such as transposing characters or leaving out metric details that could occur when you manually configure each server. Alert templates also help you roll out changes by making the updates in a single location and then rolling out the changes across your environment. If you are more interested in ad hoc alert configuration, remember that you can use a template to return a server to default settings if you make any changes that you later want to revert.

**Can I make changes to an existing template?**

Yes, SQLdm allows you to make changes to an existing alert template. Select the template from the list, and then click View/Edit. If necessary, you can delete any existing template except for the Default Template. To delete a template, select the template you want to delete, and then click Delete.

**How do I apply template changes to a server?**

Once you create a new template or make changes to an existing template, you should apply the updates to your servers. Select the template from the list, click Apply To, select one or more server to which you want to apply the alert template, and then click OK. Note that if the template you are making changes to is already associated with a server, these changes are not automatically applied when you change the template. You must use the Apply To button and select the appropriate servers.

**How do I set a template to use as the default?**
You can set an existing template as the default for SQLdm to apply to all new servers added without specifying an alert template. To designate a template as the default, click the template in the list, click Set Default, and then click OK. If there is only one template in your list, SQLdm uses this template as the default.
Configure alert template settings

The Alert Template Configuration window allows you to set up the alerts templates for multiple or all SQL Server instances.

TIP Be sure to review a list of the default alert IDs, their descriptions, and associated events before configuring your alerts.

You can set the acceptable thresholds using the following alert types:

<table>
<thead>
<tr>
<th>Alert Type</th>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Green</td>
<td>Acceptable threshold where SQLdm does not generate an alert.</td>
</tr>
<tr>
<td>Informational</td>
<td>Blue</td>
<td>Informational threshold where SQLdm generates an informational alert.</td>
</tr>
<tr>
<td>Warning</td>
<td>Yellow</td>
<td>Warning threshold where SQLdm generates a warning alert.</td>
</tr>
<tr>
<td>Critical</td>
<td>Red</td>
<td>Critical threshold where SQLdm generates a critical alert.</td>
</tr>
</tbody>
</table>

In addition to setting the thresholds for specific alerts, you can customize your alerts by adding comments to each alert. Comments allow you to include instructions to users when a level reaches a particular threshold or to provide additional information about the alert. You can also customize the levels at which SQLdm provides alert notifications by changing the informational, warning, and critical values on the Server Thresholds tab of the Alert Settings window.

You can also base your alerts on the past performance of the metrics collected by SQLdm. This is a powerful and effective way to ensure that the alerts you receive are outside of your typical metric ranges.

TIP For each alert metric, you can enter the specific criteria by double-clicking the values on the Configuration tab.

To configure the settings for an alert template:

1. Select the metric you want to edit from the list in the Alert Template Configuration window.
2. Check the boxes next to Informational, Warning, and Critical to include alerts for these states.
3. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level.
4. Select the Comments tab and enter information you want displayed in the alert message for this metric.
5. Click Apply.
6. If you want the edits you made to be replicated to every SQL Server instance SQLdm is monitoring, click Yes.
7. Click OK to accept your changes.

What is an informational alert?

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQLdm. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

TIP Note that for certain metrics, using the informational alert means that you will no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.
Can I add more metrics to my alert configuration?

You can add custom metrics to your alert configuration by using the Add Custom Counter wizard on the Administration window. This dialog allows you to use custom counters to track metrics that SQLdm begins to monitor. You can select metrics in PerfMon, SQL Server, and even metrics monitored via custom T-SQL script.
Set advanced alert options

The Advanced Alert Settings window allows you to control how and when alerts are raised by limiting the number of alerts and notifications that are generated.

**Alert Duration**

Alert duration allows you to select the total time an alert is raised for an event.

**Alert Suppression**

Alert suppression allows you to select the amount of time SQLdm waits before alerting on a metric that has exceeded threshold limits.

**Alert Filters**

Alert filters allows you to select:

- Disk drive exclusions to exclude one or more disk drives from triggering an alert
- Text and regular expressions to exclude certain words and expressions from triggering an alert

**Autogrow Settings**

Autogrow settings allows you to select whether you want SQLdm to take autogrowth into account when calculating the remaining space available.

**Database Exclusions**

Database exclusions allows you to select the databases you want to exclude from database alerts.

**Job Exclusions**

Job exclusions allows you to select both the jobs and job categories you want to exclude from job alerts.

**Sessions Exclusions**

Sessions exclusions allows you to select the applications, host servers, and users you want to exclude from job alerts.

**Custom Counters**

Custom counters allows you to select whether SQLdm generates an alert when a custom counter is not collected.

**How do I configure a custom response when an alert is generated by SQLdm?**

SQLdm allows you to configure when and how alert responses are generated. You can select from the following alert responses (providers):

- Send an email to a person or group of people
- Log an event in the Windows Event Log
- Enable the Query Monitor
- Execute a program or utility
- Send an event to your Network Management tool
- Execute a SQL Agent Job
- Execute a Script
- Create a To Do (Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.)
To configure a custom response when an alert is generated, select Tools > Alert Actions and Responses from the console toolbar.
Configure alert duration

SQLdm allows you to determine the total time an alert is raised for an event occurring on a specified SQL Server instance. The Alert Duration tab of the Advanced Alert Configuration window provides a page for you to enter the duration of the alert in hours and minutes. For the SQL Server Agent Job Failure metric, you can specify whether you want SQLdm to trigger the alert each time any job fails or to raise an alert only upon the failure of the most recently-executed job.

The metrics that allow for specifying alert duration include the following:
  - SQL Server Agent Job Completion
  - SQL Server Agent Job Failure
  - SQL Server Agent Long Running Job (Minutes)
  - SQL Server Agent Long Running Job (Percent)

Use ad hoc Alert Configuration to enter how long a job alert remains active:

1. In the Navigation page, click Servers.
2. Right-click the appropriate SQL server instance, and then select Configure Alerts.
3. Select the metric for which you want to configure an alert.
4. Click Advanced.
5. Click Alert Duration.
6. Type or use the selection window to enter the alert duration in hours and minutes.
7. Click OK.
8. To accept these changes, click OK.

Use Alert Configuration templates to enter how long a job alert remains active:

1. Click Tools > Alert Configuration Templates.
2. Click New or select an existing template, and then click View/Edit.
3. If you clicked New, type a name, and then click OK.
4. If you clicked View/Edit, click OK.
5. Select the metric for which you want to configure an alert.
6. Click Advanced.
7. Click Alert Duration.
8. If you selected the SQL Server Agent Job Completion, SQL Server Agent Long Running Job (Minutes), or SQL Server Agent Long Running Job (Percent) metric, continue with the next step. If you selected the SQL Server Agent Job Failure metric, select Anytime a job failure occurs to trigger an alert each time a job failure occurs or select Only if the most recent job execution failed if you want to raise an alert only when the most recently-executed job fails.
9. Type or use the selection window to enter the alert duration in hours and minutes.
10. Click OK.
11. If you want the edits you made to be replicated to every SQL Server instance SQLdm is monitoring, click Yes.
12. To accept these changes, click OK.
Configure alert suppression

In certain environments, it is acceptable for some metrics to exceed the configured alert thresholds for very short periods of time, for example one refresh. Typically these exceptions last less than a minute. If you are monitoring many SQL Server instances or if the amount of data collected is very high, this could produce a significant number of alerts. The Advanced Alert Configuration window allows you to limit the number of these alerts that are generated by letting you input the number of minutes a threshold violation occurs before an alert is raised, reducing the amount of alert "noise" you receive.

**How many minutes should I enter on the Advanced Configuration window?**

The amount of time you enter here depends on how critical the metric is in your environment and the amount of time SQLdm waits between refreshes. If you enter a time less than that of the refresh, the second consecutive refresh where the alert is outside the acceptable threshold range will prompt the alert.

The following table displays the length of time SQLdm waits before an alert is raised when the scheduled refresh occurs every six minutes and various values are entered:

<table>
<thead>
<tr>
<th>Time entered on the Advanced Alert Configuration window</th>
<th>Alert is generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5 minutes</td>
<td>Second Refresh occurs six minutes after the first time the problem is encountered</td>
</tr>
<tr>
<td>6-11 minutes</td>
<td>Third Refresh occurs 18 minutes after the first time the problem is encountered</td>
</tr>
<tr>
<td>12-17 minutes</td>
<td>Fourth Refresh occurs 24 minutes after the first time the problem is encountered</td>
</tr>
</tbody>
</table>
Configure alert filters - disk drive exclusions

By default, certain metrics include all drives within a specific SQL Server instance when monitoring for events that can trigger an alert. The Alert Filters tab of the Advanced Alert Configuration window allows you to select one or more disk drives to exclude from alerting on the selected metric.

The metrics that allow for disk drive exclusions include the following:

- Average Disk Milliseconds Per Read
- Average Disk Milliseconds Per Transfer
- Average Disk Milliseconds Per Write
- Disk Reads Per Second
- Disk Transfers Per Second
- Disk Writes Per Second
- OS Average Disk Queue Length Per Disk (Count)
- OS Disk Full (Percent)
- OS Disk Time Per Disk (Percent)

**Use ad hoc Alert Configuration to exclude disk drives from an alert:**

1. In the Navigation page, click Servers.
2. Right-click the appropriate SQL server instance, and then select Configure Alerts.
3. Select the metric for which you want to configure an alert.
4. Click Advanced.
5. Click Alert Filters.
6. Type the name of the disk drives you want to exclude from alerts. Use semicolons to separate names and do not use wildcard characters.
7. Click OK.
8. To accept these changes, click OK.

**Use Alert Configuration Templates to exclude disk drives from an alert:**

1. Click Tools > Alert Configuration Templates.
2. Click New or select an existing template, and then click View/Edit.
3. If you clicked New, type a name, and then click OK.
4. If you clicked View/Edit, click OK.
5. Select the metric for which you want to configure an alert.
6. Click Advanced.
7. Click Alert Filters.
8. Type the name of the disk drives you want to exclude from alerts. Use semicolons to separate names and do not use wildcard characters.
9. Click OK.
10. If you want the edits you made to be replicated to every SQL Server instance SQLdm is monitoring, click Yes.
11. To accept these changes, click OK.
Configure text and expression alerts

SQL Server Agent Log and SQL Server Error Log metrics allow users to trigger an alert based on certain text or regular expressions. SQLdm provides different levels of alerts so you can select whether you want to generate a critical, warning, or informational alert based on your entered data.

Use ad hoc Alert Configuration to enter text or a regular expression to trigger an alert:

1. In the Navigation page, click Servers.
2. Right-click the appropriate SQL server instance, and then select Configure Alerts.
3. Select the metric for which you want to configure an alert.
4. Click Advanced.
5. If you want to trigger an alert for certain text, type the text in the appropriate field. Use semicolons to separate names and a percent (%) character as a wildcard.
6. If you want to trigger an alert for a certain regular expression, type the expression in the appropriate field. Type only one regular expression per line.
7. Click OK.
8. To accept these changes, click OK.

Use Alert Configuration Templates to enter text or a regular expression to trigger an alert:

1. Click Tools > Alert Configuration Templates.
2. Click New or select an existing template, and then click View/Edit.
3. If you clicked New, type a name, and then click OK.
4. If you clicked View/Edit, click OK.
5. Select the metric for which you want to configure an alert.
6. Click Advanced.
7. If you want to trigger an alert for certain text, type the text in the appropriate field. Use semicolons to separate names and a percent (%) character as a wildcard.
8. If you want to trigger an alert for a certain regular expression, type the expression in the appropriate field. Type only one regular expression per line.
9. Click OK.
10. If you want the edits you made to be replicated to every SQL Server instance SQLdm is monitoring, click Yes.
11. To accept these changes, click OK.

What is an informational alert?

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQLdm. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

Note that for certain metrics, using the informational alert means that you no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.
SQL Server can automatically expand a database or log when necessary based on a set of parameters. The Autogrow Settings tab of the Advanced Alert Configuration window allows you to specify whether you want to take this autogrowth into account when calculating the remaining space available.

SQLdm allows you to select whether to trigger an alert based on the current database or log used size divided by the maximum possible size. Click Yes, alert on the current used size divided by the maximum possible size on the Autogrow Settings page to set the alert to include autogrow or click No, alert on the current used size divided by the current file size if you do not want to take SQL Server autogrowth into consideration.
The Alert Filters tab of the Advanced Alert Configuration window allows you to select databases to exclude from alerting on the selected metric. You can limit the amount of alert noise generated for very small databases, which tend to report a high fragmentation rate, or databases that are in single user mode and are usually in use.

The metrics that allow for database exclusions include the following:

- Database Full (Percent)
- Database Status
- Log Full (Percent)
- Table Fragmentation (Percent)

To select databases to exclude from a specific alert:

1. Open the Alert Configuration window and select the alert you want to configure.
2. Click Advanced.
3. Click Exclude.
4. Select one or more multiple databases (by holding down Shift) from the list and click Exclude.
5. Click OK.
6. To accept these changes, click OK.
Configure job exclusions

The Alert Filters tab of the Advanced Alert Configuration window allows you to select jobs or job categories to include or exclude from alerting for the specified metric. Filtering allows you to limit the amount of alert noise generated for SQL Server Agent Jobs or Job categories that, for example, are known to run for a long time.

**TIP**

By default, SQLdm alerts on all Jobs. If you check the Alert on individual job steps box, all job steps generate alerts. Using the Include filter limits job alerts to only those SQL Server Agent Job steps that apply to the filter. The Exclude filter limits the job alerts to every SQL Server Agent Job step except for those included in the Exclude filter.

The metrics that allow for SQL Server Agent Job exclusions include the following:

- SQL Server Agent Job Completion
- SQL Server Agent Job Failure
- SQL Server Agent Long Running Job (Minutes)
- SQL Server Agent Long Running Job (Percent)

To filter SQL Server Agent Job alerts:

1. Open the Alert Configuration window and select the alert you want to configure.
2. Click Advanced.
3. Select the Job Filters tab.
4. Check the Alert on individual job steps box if you would like an alert for each step in each job.
5. Click the Add button under the Include or Exclude filters.
6. Enter the Job Name or Job Categories to include or exclude and click OK. Use the percent symbol (%) as a wildcard.
7. Click OK to apply your changes.
The Advanced Alert Configuration window allows you to exclude applications, host servers, and users from the selected alert. You can limit the amount of alert noise generated by known applications, power users, or a even whole list of power users, such as a development team, that commonly exceed your alert thresholds.

The metrics that allow for session exclusions include the following:

- Blocking Session Wait Time (Seconds)
- Oldest Open Transaction (Minutes)
- Session CPU Time (Seconds)

**To configure session exclusions:**

1. Open the Alert Configuration window and select the alert you want to configure.
2. Click Advanced.
3. Enter the applications, host servers, and users you want to exclude from the alert using % for wildcards and semicolons to separate application names.
4. Click OK to save your configuration.
The Advanced Alert Configuration window allows you to select whether you want an alert to be created if SQLdm is unable to collect a custom counter. While selecting to not generate an alert if the custom counter cannot be collected eliminates any potential noise the alert could generate.

**TIP** If a custom counter fails to be collected, it is included in a critical summary alert that lists all the custom counters that were not able to be collected.
The Add Job Filter or Edit Job Filter window allows you to create or edit a filter to determine when you want a SQL Server Agent job condition to raise an alert.

**How do I access the add or edit job filters?**

To access the Add Job Filter or Edit Job Filter window for a single SQL Server instance:

1. Right-click a SQL Server instance in the Servers tree.
2. Select **Configure Alerts**.
3. Select one of the following alerts:
   - SQL Server Agent Job Completion
   - SQL Server Agent Job Failure
   - SQL Server Agent Long Running Job (Minutes)
   - SQL Server Agent Long Running Job (Percent)
4. Click the Advanced button and select **Job Filters**.
5. Click **Add** or **Edit** in either the Include or Exclude list.

To access the Add Job Filter or Edit Job Filter window for all SQL Server instances:

1. Select **Tools > Default Alert Configuration**.
2. Select one of the following alerts:
   - SQL Server Agent Job Completion
   - SQL Server Agent Job Failure
   - SQL Server Agent Long Running Job (Minutes)
   - SQL Server Agent Long Running Job (Percent)
3. Click the Advanced button and select **Job Filters**.
4. Click **Add** or **Edit** in either the Include or Exclude list.

**How do I use the Job Filters tab?**

The Job Filters tab allows you to include or exclude a job by job category, name, or step name if you check **Alert on individual job steps**.

**To add a job filter:**

1. Select **Like** or **Equals** in the Operator drop-down window for either the Job Category, Job Name, or Step Name field.
2. Enter the value to filter the job for in the Value field or click the **Browse** button to browse for the object you want to filter.
   - **TIP**: Use the percent symbol (%) as a wildcard.
3. Click **OK**.
   - **TIP**: The Browse button is only available when adding or editing job step filters for a single SQL Server instance.
The Value Configuration window allows you to select whether you receive an alert for a particular event based on the selected level for that event. For example, if you select the Critical alert level for Offline for Database Status, if the Database is ever offline you will receive a Critical Level alert.

<table>
<thead>
<tr>
<th>Alert Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅</td>
<td>The selected metric is in an OK state.</td>
</tr>
<tr>
<td>🔄</td>
<td>The selected metric is in an Informational state.</td>
</tr>
<tr>
<td>⚠️</td>
<td>The selected metric is in a Warning state.</td>
</tr>
<tr>
<td>⚥</td>
<td>The selected metric is in a Critical state.</td>
</tr>
</tbody>
</table>

**How do you open the Value Configuration dialog?**

1. Select Tools > Default Alert Configuration to open the Default Alert Configuration window.
2. Select one of the following metrics:
   - Database Status
   - Mirroring Status
   - OS Metrics Collection Status
   - DTC Status
   - Full-Text Search Status
   - SQL Server Agent Status
   - SQL Server Status
   - SQL Server Agent Job Completion
3. Click Edit.
4. Make any necessary changes, and then click OK.

**What is an informational alert?**

Informational alerts allow you to set a threshold that, when generated, triggers a status that does not affect the overall status of the server within SQLdm. You can use informational alerts to notify an administrator of the state of a particular metric for a server or trigger secondary processes that could take action to prevent issue escalation.

**TIP** Note that for certain metrics, using the informational alert means that you will no longer receive a warning or critical alert for events generated by that metric. Please review the situation before setting up an informational alert.
One of the most important steps in setting up SQLdm to successfully monitor your SQL Server instances is the configuration of your alerts so that you only receive alerts on metrics that actually fall outside the typical performance of your environment.

SQLdm allows you to do this by letting you set a period of time where the metric ranges collected for a set of alerts help determine your alert thresholds. After data begins to be collected, SQLdm will give you recommendations for tweaking your alert thresholds for the following metrics:

- OS Memory Usage (Percent)
- Blocked Sessions (Count)
- Client Computers (Count)
- SQL Server CPU Usage (Percent)
- OS Average Disk Queue Length (Count)
- OS Disk Time (Percent)
- OS Privileged Time (Percent)
- OS User Time (Percent)
- OS Processor Time (Percent)
- OS Paging (Per Second)
- OS Processor Queue Length (Count)
- SQL Server Response Time (Milliseconds)
- Custom Counters

To configure your alert thresholds:

1. **Configure the performance baseline.**
2. **View suggested alert thresholds.** Ideally you should wait until the baseline period is over. You will begin to notice blue bars (see the image below) on the Alert Configuration window, 24 hours after collections have begun, which represent your reference ranges.

![Image of alert configuration window]

3. **As your performance baseline period ends,** text appears at the top of your Alert Configuration window letting you know that recommendations are available. Click the Alert Recommendation text to open the Alert Recommendations window for your SQL Server instance. This window allows you to see the thresholds SQLdm recommends based on the data collected during your performance baseline.
4. **Either apply the alert recommendations or tweak your thresholds.**
The Configure Baseline window allows you to select the date range that is used by SQL diagnostic manager to set your alert threshold. As data is collected, your alert recommendations are calculated using your baseline.

**To configure your performance baseline:**

1. Right-click the SQL Server instance from the Servers tree and select Configure Baseline.
2. Select to either have SQLdm automatically configure your baseline or to enter custom date ranges to use:

   **Automatic**
   
   This a moving window that updates your alert baseline on the last 7 days of collected data.

   **Custom**
   
   This is a specific time range that will not change.

3. Select the days and hours SQLdm should use to calculate your baseline.

**TIP**

These fields allow you to select the days and times in which to base your alert baseline calculations. You can uncheck days and times, when your SQL Server instances are not busy, to omit from the calculation so that you receive a more realistic and accurate baseline. The days and hours selected apply to both the Automatic and Custom baseline selections.
SQLdm provides you alert recommendations to help eliminate false alerts, such as alerts for metrics that fit within the typical performance of your server. You can view these recommendations on the Alert Recommendations window. The Alert Recommendations window is accessed by clicking the "Alert recommendations are available for this SQL Server. Click here to view the recommendations now." text at the top of the Alert Configuration window. This text appears after collections have occurred in which alert thresholds fall within the baseline range. The recommendations presented are calculated using your baseline and adding either 20% or 30% over the standard deviation. To edit the percentage over the standard deviation used, select the Options button at the bottom of the Alert Recommendations window.

**To use the Alert Recommendations window:**

1. Select the first metric in the Alert Recommendations list. The current alert configuration appears at the bottom of the window so that you can easily compare that against the recommended warning and recommended critical columns.
2. If you agree with the recommendations and want to apply them to your metric, check the Apply box.
3. Click OK.
Use the Alert Recommendation Options window to change the percentage above your performance baseline that SQLdm uses when recommending alert thresholds. Either type a percentage above the baseline for the Warning and Critical alert thresholds or use the Up and Down arrows.
SQL diagnostic manager allows you to monitor your clustered SQL Server environment including alerts when a failover occurs and when the preferred active cluster node changes.

An important way to track clustering events using SQLdm is through the usage of cluster alerts.

**Cluster Failover Alert**

By default, this alert is enabled and set to critical. When a failover occurs, SQLdm generates an alert and then performs one of the following actions:

- Clears after 12 hours if no additional alerts are generated on the cluster.
- Generates an additional alert if another cluster failover occurs.
- Clears instantly when manually cleared.

**Cluster Active Node Alert**

By default, this alert is enabled and pre-populated with the first node added to SQLdm. You can change this setting on the Cluster Settings tab of the Monitored SQL Server Properties dialog box.

The Cluster Active Node is triggered when SQLdm detects that a clustered SQL Server instance is running on a node other than the configured preferred node.

**What are the best practices for using the Cluster Alerts?**

Choose the alert that best reflects your environment and setup notifications for it. Disable the other alert if it is not necessary.
The Alerts view allows you to view all the alerts for all the servers in your organization. You can organize the alerts in the list using either the Current View or the Filter.

**TIP**
The Change column is a useful way to see the agent history of a metric. This column tracks every status change occurring between OK, informational, warning, and critical states.

### How do I organize your alerts by the Current View?
Organize your alerts by selecting from one of the following views from the Current View section of the Navigation Pane:

- **Active**
  Organizes all your alerts that are currently active.

- **By Severity**
  Organizes all your alerts by their severity (warning or critical).

- **By Server**
  Organizes all your alerts by monitored SQL Server instance.

- **By Metric**
  Organizes all your alerts by each of the metrics affected.

- **Agent Job Failures**
  Lists all the alerts caused by Agent Job failures.

- **Blocked Processes**
  Lists all the alerts caused by blocked processes.

- **Oldest Open Transactions**
  Lists the oldest open transactions that have alerts associated with them.

- **Query Monitor**
  Lists all the alerts associated with the Query Monitor.

- **Table Reorganization**
  Lists all the alerts associated with table reorganization.

- **Custom**
  Allows you to configure the view to show what is most important to you.

### How do I organize my alerts using the Filter?
Click the Show Filter Options toggle in the Navigation Pane to open the Alerts filter. You can select filter options such as the view the alert is associated with, the SQL Server instance, the metric, the severity, and the time range you would like to see.

You can also filter alerts based on the tags that have been assigned to them.
Configure how SQLdm responds to alerts

The Alert Actions and Responses window allows you add, modify, and configure alert responses.

- You can create a To Do item and assign it to specific users when the conditions configured in your notification rule are met. Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.
- Configure your email (SMTP) settings to send out alert notifications using the email server available on your network.
- Configure your network management (SNMP) settings to set up a network management alert response.

How do I get here?

Select Tools > Alert Actions and Responses to open the Alert Actions and Responses window.

When would I want to set up an alert action and provider?

When you first add your SQL Server instance to SQLdm, you should begin setting up your alert thresholds. Once your alert thresholds have been set, you should begin setting up your alert action rules and providers. This allows you to select who receives notifications or is assigned To Do tasks when an alert threshold is met.

What actions are available on the Alert Responses tab?

You can enable or disable an action provider by checking/un-checking the box in the Enabled column.

Add

Click Add to create a new alert response.

Edit

Click Edit to make modifications to an existing alert response.

Copy

Click Copy to make a duplicate of an existing alert response. This can save time when you want to change just one setting on a particular rule.

Remove

Click Remove to delete the selected alert response.

What actions are available on the Action Providers tab?

You can enable or disable an action provider by checking/un-checking the box in the Enabled column.

Add

Click Add to create a duplicate action so you can add multiple instances to a rule. For example, you can create an alert response rule to trigger more than one SQL Server Agent job.

Edit

Click Edit to edit an SMTP or SNMP action provider you have created previously.

Remove

Click Remove to delete an SMTP or SNMP action provider you have created previously.
Use the Alert Response window to create or modify your new alert response rule.

**How do I access the Alert Response window?**

1. Select **Tools > Alert Actions and Responses**.
2. Click **Add** on the **Alert Responses** tab.

**How do I add a new rule?**

After you create a new rule, you can then create a To Do item and assign it to specific users for when the conditions configured in your notification rule are met. Note that beginning with SQLdm 7.0, the To Do feature is available only to users who had To Do activated in an earlier 6.x version. SQLdm 7.1 does not support this feature for new users.

1. Click **Add** on the **Alert Responses** tab of the Alert Actions and Responses dialog box.
2. Enter a **Name** for the rule.
3. Check the boxes for the conditions and destinations for the rule.
4. Click **Enable** to activate the associated action. Note that all actions are disabled by default.
5. **If you enabled an action**, provide any required information for the selected action.
6. Click **Add** to select additional actions to perform in response to the alert. This feature allows you to add multiple instances of certain actions.
7. Click the hyperlink text to select specific conditions for the rule.
8. Click **OK** when finished.

**What conditions are available?**

By default, alert response rules are not limited to specific SQL Server instances, metrics, severity levels, and times. You can use conditions to tailor the triggers of your alert responses. The following conditions help you manage alert response rules.

**Where the SQL Server Instance is in specified list**

Trigger a response when the listed SQL Server instance is included in the displayed list. Click specified list, and then select one or more instances to include in the alert response trigger. Click Select all to include all of the displayed instances.

**Where SQL Server Instance has a tag in specified list**

Trigger a response when the listed SQL Server instance has an existing tag in the displayed list. Click specified list, and then select one or more tags to include in the alert response trigger. Click Select all to include all of the displayed tags.

**Where the metric is in the specified list**

Trigger a response if the metric is included in the displayed list. Click specified list, and then select one or more metrics to include in the alert response trigger. Click Select all to include all of the displayed metrics.

**Where metric severity has changed**

Trigger a response when the metric severity changes regardless of level. Click metric, and then select one or more metrics to include in the alert response trigger. Click Select all to include all of the displayed metrics. Click the displayed severity level, and then select one or more severity levels to include in the alert response trigger.
**Where refresh occurred during a specific time frame**

Trigger a response if the refresh occurred during the displayed time frame. Click during this time, and then select the time and date range to set the alert response trigger time frame.

**What action providers are available?**

Action providers allow you to select actions that you want to occur as part of the alert response. By default, all actions are disabled. To enable an action, click Enable next to the appropriate action. The following actions help you manage alert response rules.

**Idera Newsfeed Action Provider**

Publish the alert message as a status update on the server wall in the Idera Newsfeed interface. For more information, see the Idera Newsfeed Help.

**Email (SMTP) Provider**

When the criteria for the new alert rule is met, SQLdm sends an alert email message to the SMTP server, which then forwards the message to the specified email address. Configure your email settings and network manager settings to receive these alert email messages.

**Network Management Protocol (SNMP) Trap Message Provider**

When the criteria for the new alert rule is met, SQLdm sends an SNMP Trap message to the network management console based on the configuration of your email settings and network management settings.

**Enable Query Monitor Action Provider**

When the criteria for the new alert rule is met, it will trigger the Query Monitor to start so that more information can be collected.

To enable query monitor as an alert response:

1. Add the Enable Query Monitor provider.
2. Click for this time period in the Select Actions section of the Alert Response window.
3. Select either Enable query monitor to enable the query monitor until it is manually disabled or Enable query monitor for a limited time.
4. Click OK.

**EventLog Action Provider**

When the criteria for the new alert rule is met, an entry to the EventLog is created.

**Program Action Provider**

When the criteria for the new alert rule is met, you can specify a program for the SQLdm Management Service to run in response to the alert.

To start a program as an alert response:

1. Add the Program Action provider.
2. Click Program in the Select Actions section of the Alert Response window.
3. Enter a description for the Program action.
4. Enter the full path to the program you want to launch as an alert response. Use the More Options drop-down menu to select from common runtime arguments.

**TIP**

You can only run executable (.exe) files through the Program Action Provider. If you need to run a batch (.bat) file, you will have to use `cmd /c run.bat`. 
5. Enter the full path to the directory you want the program to run in.

**TIP**
The Start in field is useful when you want the directory in which an action begins to be a different directory than that of the program.

6. Click Test to ensure that the SQLdm Management Service can run the program.
7. Click OK.

**SQL Agent Job Action Provider**

When the criteria for the new alert rule is met, you can specify a SQL Agent Job to run in response to the alert.

**To start a SQL Agent Job as an alert response:**
1. Add the SQL Agent Job Action provider.
2. Click the SQL Agent Job text in the Select Actions field.
3. Select the SQL Server instance from the Server drop-down list or select %(Instance) to run the SQL Agent Job on the server that caused the alert.
4. Click Browse next to the Job Name field.
5. On the Select Job window, select the SQL Server instance the SQL Agent Job is located on and click Load.
6. Select the SQL Agent job from the list and click OK.
7. If you want to select a particular Job Step to run, click the Browse button in the Job Step window
   a. In the Select Job Step window click Load.
   b. Select the Job step from the list and click OK.

**SQL Script Action Provider**

When the criteria for the new alert rule is met, you can specify a SQL Server script to run in response to the alert.

**To run a SQL Script Action Provider as an alert response:**
1. Add the SQL Script Action provider.
2. Click the SQL Script text in the Select Actions field.
3. Enter a description for the SQL Script Action in the Description field.
4. Click Browse to select the SQL Server instance or leave it at the default of %(Instance) to run the SQL Script on the server that caused the alert.
5. Either type in the T-SQL you want to run when the alert criteria is met, or paste in the T-SQL from another application.
6. Click OK.

**To Do Action Provider**

When the criteria for the new alert rule is met, a To Do list item is created.

**TIP**
Idera is deprecating the To Do list feature beginning with SQL diagnostic manager 7.0. While users who had To Do activated in an earlier 6.x version can continue to use To Do items, SQLdm 7.1 does not support this feature for new users. Current To Do users can reference Idera Solution for information on how to enable the To Do view.

**To create a To Do as an alert response:**
1. Add the To Do action provider.
2. Click the someone text in the Select Actions field.
3. Enter the users to assign the To Do in the Users field.
4. Enter a custom message in the Message field or accept the default text.
5. Click OK.
How do I change the condition value?

Click the underlined value and SQLdm displays a list of currently defined items for you to select as the new value. Make the necessary updates, and then click OK.
The Add Action Provider dialog box allows you to pick the action provider type you want to use for your alert response. The following choices are available:

**Enable Query Monitor Action Provider**
- Enable the Query Monitor when the criteria of your Alert Response rule is met.

**Email (SMTP) Provider**
- When the criteria for the new alert rule is met, SQLdm sends an alert email message to the SMTP server, which then forwards the message to the specified email address. SQLdm allows you to configure your email settings and configure your network management settings to properly route alert messages.

**Idera Newsfeed Action Provider**
- Publish the alert message as a status update on the server wall in the Idera Newsfeed interface and on the Active Alerts view in SQLdm Mobile. For more information, see the Idera Newsfeed Help.

**Network Management (SNMP) Trap Message Provider**
- When the criteria for the new alert rule is met, SQLdm sends an SNMP Trap message to the specified network management console. SQLdm allows you to configure your email settings and configure your network management settings to properly route alert messages.

**Program Action Provider**
- Start a program when the criteria of your Alert Response rule is met.

**SQL Agent Job Action Provider**
- Execute a SQL Agent job when the criteria of your Alert Response rule is met.

**SQL Script Action Provider**
- Execute a SQL Script when the criteria of your Alert Response rule is met.

**To Do Action Provider**
- Create a To Do Action when the criteria of your Alert Response is met. This option only displays when you have upgraded from SQLdm 6.x and previously enabled the To Do feature. This feature is deprecated in SQLdm 7.0 and later.
The Alert Communications wizard allows you to set up and configure SMTP (email) and SNMP (network management) providers for your alerts. Click Next to continue to the Choose Provider Type window of the wizard.

How do I open the Alert Communication wizard?

1. Select Tools > Alert Actions and Responses from the SQLdm toolbar menu.
2. Select the Action Providers tab and click Add.
The Choose Provider Type window of the Alert Communication wizard allows you to choose which type of provider you want to create. Select either Simple Mail Transfer Protocol (SMTP) or Simple Network Management Protocol (SNMP) from the Provider Type drop-down, give the Provider a name and click Next to configure your provider.

**Simple Mail Transfer Protocol (SMTP)**

This provider allows you to send an email to one or multiple email addresses whenever alerts are generated.

**Simple Network Management Protocol (SNMP)**

This provider allows you to send an event to your Network Management system whenever alerts are generated.

**How do I access this window?**

1. Select **Tools > Alert Actions and Responses** from the SQLdm toolbar menu.
2. Select the Action Providers tab and click Add.
3. Click Next on the Welcome to the Alert Communications wizard window.
Configure email settings

The Alert Communications wizard allows you to configure email settings for sending out alert notifications using the email server available on your network.

**To configure your email settings:**

1. In the Alert Communications wizard, select the Provider Type.
2. Type a unique Provider Name, and then click Next.
3. Enter the SMTP address, port number, and the number of seconds to wait before timing out.
4. If the Server requires authentication, check Server requires authentication, and then type the appropriate login information for the email server.
5. Type the Name and E-mail address you want to appear in the From field in alert notifications.
6. To verify the connection to the SMTP server, click Test. Enter the email address to send the test email, and then click OK.
7. Verify that the test email arrived and click OK if the test is successful.

**How do I access the Alert Communications wizard?**

Select Tools > Alert Actions and Responses. Click the Action Providers tab, and then click Add.
The **Alert Communications wizard** allows you to set up Simple Network Management Protocol (SNMP) for sending out alert events to your Network Management system.

**To set up your SNMP settings:**

1. Enter the SNMP address, port number, and the community name to be used.

   **TIP** It is best to specify a computer name for the address, especially in a DHCP environment. If the manager has a static address, the IPv4 formatted address can be entered instead.

2. Click Test to verify the connection to the SNMP enabled management console.
3. Verify that the test event arrived and click **OK** if the test is successful.
4. Click **Finish**.

**Does Idera provide a MIB file that I can load into my Network Management Console?**

SQLdm includes a MIB that you can use in your Network Management System to format the events that are sent by SQLdm. The MIB file is located in the root directory of your SQLdm installation and is named Idera.MIB.

**What is a MIB file?**

A MIB is a standardized file that defines the data that is sent to the Network Management Console in the SNMP Trap events generated by SQLdm.

**How do I access this window?**

1. Select **Tools > Alert Actions and Responses** from the SQLdm toolbar menu.
2. Select the Action Providers tab and click **Add**.
3. Click Next on the Welcome to the Alert Communications wizard window.
4. Select **Simple Network Management Protocol (SNMP)** from the Provider Type drop-down.
5. Enter a name for your provider.
6. Click **Finish**.
The Reports view allows you to create reports that allow you to analyze current and historical performance and statistical data.

SQLdm provides three types of comprehensive reports: Monitor, Analyze, and Plan along with the ability to create custom reports.

**TIP** Reports look best when printed in the letter size format.

How can I customize my reports?

After you run a report, you can use the Report Toolbar to select the page setup, page width, and print layout to customize your report.

How can I filter my reports?

Click Show Filter Options to open the Filter area of the Reports view. Each report has several filter options, such as which SQL Server instances to display, the period of time to report on, and the time intervals to show in the charts.

How can I deploy reports to Microsoft Reporting Server?

You can use the SQLdm Reports Deployment wizard to deploy specified reports to the Microsoft Reporting Server.
Every report in SQLdm contains filters to allow you to select the information to be shown in your reports.

**What filter fields are available?**

- **Chart Type**
  
  Allows you to select which type of chart you want to report a trend over the specified time.

- **Compare Server**
  
  This optional information allows you to select a different SQL Server instance and period of time for you to compare statistics with the other selected Server and Period.

- **Databases/Database Name**
  
  Allows you to include the databases on which you want to report.

- **Forecast Type/Forecast Units**
  
  Allows you to select which type of forecast and number of units to use for projecting future growth trends based on your expected growth rate and sample size.

- **Drive Name**
  
  The name of the drive within the selected SQL Server instance on which you want to report.

- **Minimum Duration**
  
  Allows you to specify the minimum duration of the queries to display in your reports.

- **Number of Servers**
  
  Allows you to specify the number of SQL Server instances you want to include in your report.

- **Period**
  
  The time range on which you want to report. Select Custom from the Period drop-down list to select the specific days, months, or even days on which you want to report.

- **Sample**
  
  This is period of time between data points on the graphs that appear in your report, and represent the most current information as of that particular point.

- **Search Item and Search Value**
  
  Allows you to select up to four different items and their values to search for in your server inventory. Use a percent (%) character as a wildcard when specifying a Search Value.

- **Server**
  
  The SQL Server instance or instances on which you want to report.

- **Start Time**
  
  The start time range from which you want to report trends. SQLdm includes trends through to the most recent collection time.

- **Tag**
  
  Allows you to filter items in the report based on tags that have been assigned to those items.
Monitor reports give you overview information for your monitored SQL Server instances. Click the links below for more information.

**Enterprise Summary**
- Use the Enterprise Summary report to view the health of your SQL Servers.

**Server Summary**
- Use the Server Summary report to view the health of a single SQL Server.

**Active Alerts**
- Use the Active Alerts report to identify the currently active alerts for all monitored SQL Servers.

**Mirroring Summary**
- Use the Mirroring Summary report to view the health of your mirrored databases.

**Metric Thresholds**
- Use the Metric Thresholds report to view all metric thresholds for a SQL Server instance.
The Enterprise Summary report provides an overview of the health and performance of all SQL Server instances monitored by SQLdm. Click the appropriate instance and SQLdm displays the Server Summary report.

**When should I run this report?**

You can run the Enterprise Summary report to view a quick overview of the health and performance of all monitored instances. This report allows you to see the status of all your SQL Server instances at a glance.

**How does SQLdm calculate the metric values on this report?**

Most values on the Enterprise Summary report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237
The Server Summary report provides a detailed overview of the health and performance of a specific SQL Server, such as response time and CPU usage.

**When should I run this report?**

You should run the Server Summary report to view the detailed performance and health information of a single SQL Server.

**How does SQLdm calculate the metric values on this report?**

Most values on the Server Summary report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
The Active Alerts report lists all outstanding, active alerts that SQLdm has raised against your SQL Server instances. To see more detail about the health of a specific instance, click on that instance and the Server Summary report displays.

When should I run this report?
You should run the Active Alerts report to view all alerts currently active on your monitored instances. You can sort the alerts by severity level, so you can focus on the most critical alerts first.

How does SQLdm calculate the metric values on this report?
Most values on the Active Alerts report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
The Mirroring Summary report allows you to view the current mirroring sessions across all or selected tags or servers.

The data for this report comes from SQL Server system views for database mirroring. The SQLdm alert thresholds are used to highlight existing issues as defined by your configured alert settings.

When should I run this report?

You should run the Mirroring Summary report to view a quick overview of the health of your mirrored databases.

How does SQLdm calculate the metric values on this report?

Most values on the Mirroring Summary report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
The Metric Thresholds report lets you view a list of metric thresholds for a monitored SQL Server instance. To successfully monitor OS metrics, ensure that WMI/OLE Automation is enabled on the monitored instance.

**When should I run this report?**

You should run the Metric Thresholds report to review your current Informational, Warning, and Critical threshold settings for the selected SQL Server instance. This report also lists whether a metric is enabled, helping you decide what action is necessary for you to get the most out of SQLdm.
Analysis reports provide information that allows you to analyze the current effectiveness of your SQL Servers. Click the links below for more information.

**Server Analysis Reports**

- **Top Servers**
  - Use the Top Servers report to identify your worst performing SQL Servers.

- **Server Statistics**
  - Use the Server Statistics report to analyze and compare performance trends across two SQL Servers.

- **Server Inventory**
  - Use the Server Inventory report to find SQL Servers that share the same properties.

- **Query Overview**
  - Use the Query Overview report to identify your worst performing queries.

- **Top Queries**
  - Use the Top Queries report to find queries that are performing poorly or executing too frequently.

- **Alert History**
  - Use the Alert History report to analyze the alert history for a SQL Server.

**Database Analysis Reports**

- **Top Databases**
  - Use the Top Databases report identify your worst performing databases.

- **Database Statistics**
  - Use the Database Statistics report to analyze and compare performance trends across two databases.

- **Top Database Applications**
  - Use the Top Database Application report to find database applications that consume system resources.

- **Mirroring History**
  - Use the Mirroring History report to analyze the event history for a mirrored database.

- **Top Tables by Growth**
  - Use the Top Tables by Growth report identifies the fastest growing tables.

- **Top Tables by Fragmentation**
  - Use the Top Tables by Fragmentation identifies the most fragmented tables.

**Resource Analysis Reports**

- **Session Statistics**
  - Use the Session Statistics report to track key session and network performance metrics over time.
CPU Statistics
   Use the CPU Statistics report to track key CPU performance metrics.

Disk Details
   Use the Disk Details report to track key disk metrics.

Disk Statistics
   Use the Disk Statistics report to track key disk performance metrics.

Replication Statistics
   Use the Replication Statistics report to track key replication performance metrics.

Memory Statistics
   Use the Memory Statistics report to track key memory performance metrics.
Server analysis reports provide information that allows you to analyze the current effectiveness of your SQL Servers. Click the links below for more information.

**Server Analysis Reports**

**Top Servers**
- Use the [Top Servers report](#) to identify your worst performing SQL Servers.

**Server Statistics**
- Use the [Server Statistics report](#) to analyze and compare performance trends across two SQL Servers.

**Server Inventory**
- Use the [Server Inventory report](#) to find SQL Servers that share the same properties.

**Query Overview**
- Use the [Query Overview report](#) to identify your worst performing queries.

**Top Queries**
- Use the [Top Queries report](#) to find queries that are performing poorly or executing too frequently.

**Alert History**
- Use the [Alert History report](#) to analyze the alert history for a SQL Server.
The Top Servers report identifies your worst performing SQL Server instances based on the number of active alerts, response time, CPU usage, memory usage, and disk usage. Use this report to compile a list of instances that need immediate attention.

**TIP**
A value of N/A indicates that data could not be collected due to having OLE Automation disabled or a down WMI server. You must enable OS metrics monitoring to receive results.

**When should I run this report?**
You should run the Top Servers report daily to keep ahead of critical issues that would cause an instance to become unavailable.

**How does SQLdm calculate the metric values on this report?**
Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
The Server Statistics report lets you analyze and compare performance trends across two SQL Server instances. You can focus on a single performance statistic, such as memory usage, and track its value on both instances over a specified period of time.

When should I run this report?

You should run the Server Statistics report when you want to trend key performance metrics over time for one or two specific instances, in response to a known problem or a routine health check. For example, if the same third party application uses two different databases but only one database has reported issues, then comparing their key performance metrics can help you diagnose the issue.

How does SQLdm calculate the metric values on this report?

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
The Server Inventory report lets you compile a list of SQL Server instances that share up to four different properties, such as the SQL Server version or whether the instance is clustered.

**TIP** You can use % as a wildcard in the Search Value field.

**When should I run this report?**

You should run the Server Inventory report when you need to identify which monitored instances require software or hardware upgrades, or simply want to know how many instances of a SQL Server version have been deployed.

**How does SQLdm calculate the metric values on this report?**

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
The Query Overview report identifies the worst performing queries run on a specific SQL Server instance over time, including the call volume of each query. This report requires that you enabled Query Monitoring in the SQLdm Console and have collected sufficient query data for the specified time period.

**When should I run this report?**

You should run the Query Overview report when you want to determine whether SQL Server queries run through Microsoft SQL Server tools or third-party applications are impacting the performance of your monitored instances. This report can help you further diagnose a known performance issue or proactively identify queries whose resource consumption may cause issues in the future.

**How does SQLdm calculate the metric values on this report?**

Most values on the Top Tables by Growth report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
The Top Queries report lets you compile a list of queries based on call frequency, duration of execution, CPU usage, and the number of reads and writes performed on the databases hosted by the specified SQL Server instance. You can define minimum thresholds for each of these performance metrics and then see which queries match or exceed the selected values. This report requires that you enabled Query Monitoring in the SQLdm Console and have collected sufficient query data for the specified time period.

When should I run this report?

You should run the Top Queries report on a routine basis, such as every week, to proactively identify potential performance issues that may be caused by queries. You can also use this report to track historic performance trends and determine the relationship between query usage and your database performance.

How does SQLdm calculate the metric values on this report?

Most values on the Top Queries report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the Console and Reports represent? - Solution #00002237
The Alert History report lets you review the alerts that SQLdm has raised for the selected SQL Server instance over a specified period of time. You can also review alerts for multiple instances that belong to a tag.

**When should I run this report?**

You should run the Alert History report when you want to analyze the historical health of the SQL Server by identifying and tracking critical values for key performance metrics. Each raised alert represents a point in time when a metric reached or exceeded the specified threshold.

By default, SQLdm alerts on many performance-related SQL Server metrics. You can also configure alerts and create custom counters to track specific performance data.

**How does SQLdm calculate the metric values on this report?**

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
Database analysis reports provide information on databases that allows you to analyze the current effectiveness of your SQL Servers. Click the links below for more information.

**Database Analysis Reports**

- **Top Databases**
  - Use the Top Databases report to identify your worst performing databases.

- **Database Statistics**
  - Use the Database Statistics report to analyze and compare performance trends across two databases.

- **Top Database Applications**
  - Use the Top Database Applications report to find database applications that consume system resources.

- **Mirroring History**
  - Use the Mirroring History report to analyze the event history for a mirrored database.

- **Top Tables by Growth**
  - Use the Top Tables by Growth report to identify the fastest growing tables.

- **Top Tables by Fragmentation**
  - Use the Top Tables by Fragmentation report to identify the most fragmented tables.
The Top Databases report identifies your worst performing SQL Server databases based on the size of the database, the growth rate of the database, and the number of reads, writes, and transactions per second performed on the database. Use this report to compile a list of databases that used most often or have the heaviest loads.

**TIP** You can use % as a wildcard in the Search Value field.

**When should I run this report?**

You should run the Top Databases report on a routine basis, such as once a week. This report helps you:

- Plan for upcoming maintenance
- Identify issues that may impact third-party application performance even though the host SQL Server instance remains healthy
- Compare database performance across your enterprise

**How does SQLdm calculate the metric values on this report?**

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the Console and Reports represent? - Solution #00002237
The Database Statistics report lets you analyze and compare performance trends, such as the percentage of growth, across two SQL Server databases. You can also compare performance metrics at different points in time for the same database. This report can also include tempdb database statistics.

**When should I run this report?**

You should run the Database Statistics report when attempting to understand performance differences or trouble-shooting database issues.

**How does SQLdm calculate the metric values on this report?**

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal ([www.idera.com/support](http://www.idera.com/support)) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
The Top Database Applications report lets you find the applications that are consuming the highest amount of your system resources, such as CPU usage, when run on a specific database. You can define minimum thresholds for each of these performance metrics and then see which applications match or exceed the selected values. This report requires that you enabled Query Monitoring in the SQLdm Console and have collected sufficient query data for the specified time period.

**TIP**
You can use % as a wildcard in the Search Value field.

**When should I run this report?**

You should run the Top Database Applications report on a routine basis, such as every week, to proactively identify potential performance issues that may be caused by third-party applications. You can also use this report to track historic performance trends and determine the relationship between third-party application usage and your database performance.

**How does SQLdm calculate the metric values on this report?**

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the Console and Reports represent? - Solution #00002237
The Mirroring History report lets you view the status of select mirroring metrics at the time of each SQLdm scheduled refresh for the specified period.

You can filter the results by time, tag, server, and database, and also choose to view only problems as defined in your alerts. Fields showing metrics corresponding to alerts are color-coded according to the alert.

**When should I run this report?**

You should run the Mirroring History report when you want to analyze the historical health of your mirrored SQL Server databases by identifying and tracking critical values for key performance metrics. Each red or yellow value in the report grid corresponds to a raised alert. These visual warnings represent points in time when the metric reached or exceeded the specified alert threshold.

By default, SQLdm alerts on many performance-related SQL Server metrics. You can also configure alerts and create custom counters to track specific performance data.

**How does SQLdm calculate the metric values on this report?**

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
The Top Tables by Growth report identifies the fastest growing tables in a database. You can select a specific database to analyze or evaluate all databases on a specific SQL Server instance. You can also choose a growth metric and its minimum threshold to see which tables match or exceed the selected value.

When should I run this report?

You should run the Top Tables by Growth report on a routine basis, depending on how heavily your company uses the host SQL Server instance. Use this report to see how rapidly (or slowly) tables are growing over time. If database capacity on a given instance becomes limited, run this report to see which tables that are the top contributors and may need to be moved to another database.

How does SQLdm calculate the metric values on this report?

Most values on the Top Tables by Growth report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the Console and Reports represent? - Solution #00002237
The Top Tables by Fragmentation report identifies the most fragmented tables in a database. You can select a specific database to analyze or evaluate all databases on a specific SQL Server instance. You can also choose a fragmentation metric and its minimum threshold to see which tables match or exceed the selected value.

When should I run this report?

You should run the Top Tables by Fragmentation report on a routine basis, depending on how mission-critical or sensitive the data is and how heavily your company uses the host database. Use this report to identify which tables should be optimized to improve performance.

How does SQLdm calculate the metric values on this report?

Most values on the Top Tables by Growth report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237
Resource analysis reports provide information that focuses on key performance metrics of your SQL Servers. Click the links below for more information.

**Resource Analysis Reports**

**Session Statistics**
- Use the Session Statistics report to track key session and network performance metrics over time.

**CPU Statistics**
- Use the CPU Statistics report to track key CPU performance metrics.

**Disk Details**
- Use the Disk Details report to track key disk metrics.

**Disk Statistics**
- Use the Disk Statistics report to track key disk performance metrics.

**Replication Statistics**
- Use the Replication Statistics report to track key replication performance metrics.

**Memory Statistics**
- Use the Memory Statistics report to track key memory performance metrics.
The Session Statistics report lets you view trends for sessions and network connectivity, such as transactions per minute, on a monitored SQL Server instance over a specified period of time.

When should I run this report?
You should run the Session Statistics report to analyze the sessions and network connectivity for a monitored instance. High logins or transactions per minute may indicate a need for increased resources.

How does SQLdm calculate the metric values on this report?
Most values on the Session Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
The CPU Statistics report lets you view trends for CPU usage on a monitored SQL Server instance over a specified period of time. To successfully monitor OS metrics, ensure that WMI/OLE Automation is enabled on the monitored instance.

When should I run this report?
You should run the CPU Statistics report to analyze contributors to CPU usage for a monitored instance.

How does SQLdm calculate the metric values on this report?
Most values on the CPU Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
The Disk Details report lets you view key disk metrics of a monitored SQL Server instance and specific disk drive over a specified period of time. To successfully monitor OS metrics, ensure that WMI/OLE Automation is enabled on the monitored instance.

**When should I run this report?**
You should run the Disk Details report to analyze the six key metrics for disk speed, such as disk time per read or disk reads per second.

**How does SQLdm calculate the metric values on this report?**
Most values on the Disk Details report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How to customize existing reports used with Reporting Services? - Solution #00000307
The Disk Statistics report lets you view trends for the disk activity of a monitored SQL Server instance over a specified period of time. To successfully monitor OS metrics, ensure that WMI/OLE Automation is enabled on the monitored instance.

**When should I run this report?**

You should run the Disk Statistics report to analyze the disk activity such as work files per minute or page read/writes per second.

**How does SQLdm calculate the metric values on this report?**

Most values on the Disk Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
The Replication Statistics report lets you view trends for the replicated transactions on a monitored SQL Server instance over a specified period of time.

**When should I run this report?**

You should run the Replication Statistics report to analyze the efficiency of replication between databases.

**How does SQLdm calculate the metric values on this report?**

Most values on the Replication Statistics report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
Memory Statistics

The Memory Statistics report lets you view trends for memory usage on a monitored SQL Server instance over a specified period of time. To successfully monitor OS metrics, ensure that WMI/OLE Automation is enabled on the monitored instance.

When should I run this report?

You should run the Memory Statistics report to analyze the memory needs of a monitored instance. This can help you identify problems with current usage or to plan for future increases in memory.

How does SQLdm calculate the metric values on this report?

Most values on the Memory Statistics report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

○ How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
○ How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
○ How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
○ What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
Planning reports allow you to plan for future needs based on current trends. Click the links below for more information.

**Disk Space Usage Forecast**
Use the Disk Space Usage Forecast report to forecast disk space needs based on average historical growth rates.

**Database Growth Forecast**
Use the Database Growth Forecast report to forecast future database growth based on historical trends.

**Table Growth Forecast**
Use the Table Growth Forecast report to forecast future table growth based on historical trends.
The Disk Space Usage Forecast report lets you predict how much disk space may be needed in the future based on current and historical growth rate trends. For more accurate forecasting, select the largest possible period of historical data and the shortest possible projection into the future.

**TIP** The Disk Space Usage Forecast report does not display the maximum capacity of the disk. It can be used to estimate how much space may be needed, but it does not indicate when existing space will run out.

**When should I run this report?**
You should run the Disk Space Usage Forecast report to help anticipate your needs and plan for future acquisition of disk space.

**Which forecast type should I use?**
You can choose Linear or Exponential (Aggressive) forecasting types when you run this report.

Linear forecasting follows the trend of the available data and extends the forecast out in a straight line by the number of forecast units.

Exponential forecasting tends to exaggerate the trend that exists in the data. This allows you to more easily see if your data (disk space usage) is trending toward or away from a critical threshold. If the rate of growth is not constant, the exponential forecast type will give a more accurate forecast of future growth.

**How does SQLdm calculate the metric values on this report?**
Most values on the Disk Space Usage Forecast report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237
The Database Growth Forecast report lets you predict how large selected databases may grow based on current and historical growth rate trends. For more accurate forecasting, select the largest possible period of historical data and the shortest possible projection into the future.

**When should I run this report?**

You should run the Database Growth Forecast report to help anticipate your needs and plan for future database expansion.

**Which forecast type should I use?**

You can choose Linear or Exponential (Aggressive) forecasting types when you run this report.

Linear forecasting follows the trend of the available data and extends the forecast out in a straight line by the number of forecast units.

Exponential forecasting tends to exaggerate the trend that exists in the data. This allows you to more easily see if your data (database growth) is trending toward or away from a critical threshold. If the rate of growth is not constant, the exponential forecast type will give a more accurate forecast of future growth.

**How does SQLdm calculate the metric values on this report?**

Most values on the Database Growth Forecast report correlate with the related SQL Server metric.

However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
The Table Growth Forecast report lets you predict how large selected tables may grow based on current and historical growth rate trends. For more accurate forecasting, select the largest possible period of historical data and the shortest possible projection into the future.

**When should I run this report?**

You should run the Table Growth Forecast report to help anticipate your needs and plan for future creation of additional tables.

**Which forecast type should I use?**

You can choose Linear or Exponential (Aggressive) forecasting types when you run this report.

- **Linear forecasting** follows the trend of the available data and extends the forecast out in a straight line by the number of forecast units.

- **Exponential forecasting** tends to exaggerate the trend that exists in the data. This allows you to more easily see if your data (table growth) is trending toward or away from a critical threshold. If the rate of growth is not constant, the exponential forecast type will give a more accurate forecast of future growth.

**How does SQLdm calculate the metric values on this report?**

Most values on the Table Growth Forecast report correlate with the related SQL Server metric. However, some metric values represent averages over time. Likewise, a few metric values are based on performance algorithms that use multiple statistics to calculate a more accurate metric.

You can access our [Customer Service Portal](www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

- How is the Worst Performing and Most Frequent Queries information gathered? - Solution #00000412
- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #00000301
- What does the ‘Memory Usage’ metric in the console and reports represent? - Solution #00002237
In addition to the included reports, SQLdm also allows you to generate reports using Microsoft Reporting Services. Microsoft Reporting Services allows you to build powerful, custom reports to deploy for a comprehensive auditing solution.

In order to use this feature, you must have access to a server with Microsoft Reporting Services. For more information about Microsoft Reporting Services, see SQL Server Books Online.
SQLdm includes built-in reports which have been specially designed to generate commonly requested metrics reports using the SQL Server data collected in the repository.

**How do I select reports?**

Check the box for each report you want to deploy to a Microsoft Reporting Services Server, then click Next.

**Available Reports**

You can generate reports from the following report categories:

### Monitoring Reports

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Summary</td>
<td>View the health of your SQL Servers</td>
</tr>
<tr>
<td>Server Summary</td>
<td>View the health details of a single SQL Server</td>
</tr>
<tr>
<td>Active Alerts</td>
<td>View the active alerts for all monitored SQL Servers</td>
</tr>
<tr>
<td>Mirroring Summary</td>
<td>View the health of your mirrored databases</td>
</tr>
</tbody>
</table>

### Server Analysis Reports

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Servers</td>
<td>Identify your worst performing SQL Servers</td>
</tr>
<tr>
<td>Server Statistics</td>
<td>Analyze and compare performance trends across two SQL Servers</td>
</tr>
<tr>
<td>Server Inventory</td>
<td>Find SQL Servers that share the same properties</td>
</tr>
<tr>
<td>Query Overview</td>
<td>Identify your worst performing queries</td>
</tr>
<tr>
<td>Top Queries</td>
<td>Find queries that are performing poorly or executing frequently</td>
</tr>
<tr>
<td>Alert History</td>
<td>Analyze the alert history for your SQL Servers</td>
</tr>
</tbody>
</table>

### Database Analysis Reports

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Databases</td>
<td>Identify your worst performing databases</td>
</tr>
<tr>
<td>Report Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Database Statistics</td>
<td>Analyze and compare performance trends across two databases</td>
</tr>
<tr>
<td>Top Database Applications</td>
<td>Find database applications that consume system resources</td>
</tr>
<tr>
<td>Mirroring History</td>
<td>Analyze the event history for a mirrored database</td>
</tr>
<tr>
<td>Top Tables by Growth</td>
<td>Identify the fastest growing tables</td>
</tr>
<tr>
<td>Top Tables by Fragmentation</td>
<td>Identify the most fragmented tables</td>
</tr>
</tbody>
</table>

**Resource Analysis Reports**

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Statistics</td>
<td>Track key session and network performance metrics over time</td>
</tr>
<tr>
<td>CPU Statistics</td>
<td>Track key CPU performance metrics</td>
</tr>
<tr>
<td>Memory Statistics</td>
<td>Track key memory performance metrics</td>
</tr>
<tr>
<td>Disk Statistics</td>
<td>Track key disk performance metrics</td>
</tr>
<tr>
<td>Replication Statistics</td>
<td>Track key replication performance metrics</td>
</tr>
</tbody>
</table>
Configure report email notifications

Enter the email address where you would like the report notification to be sent, then click Next. For more information on configuring report servers for email delivery, see How to: Configure a Report Server for E-mail Delivery (Reporting Services Configuration).
Review that all the Reporting Services settings are correct and click Finish to apply your changes. If you want to make any changes to the settings displayed, click the Back button to go back and make your changes.
The Custom Reports view allows you to create reports on any counter collected by SQLdm and on any custom counters that you have created. You can create custom reports to view graphs of the counters followed by a grid with all the counters graphed over the entered time frame.

**How can I filter my reports?**

Report filters are available by default. You can select the Server, Time Period, Date Range, and Sample type from the drop-down lists at the top of the report. If the filters are not displayed, click the Show Filters button.

**How can I deploy reports to Microsoft Reporting Server?**

You can use the SQLdm Reports Deployment wizard to deploy specified reports to your Microsoft Reporting Server.
Create a custom report

The Custom Reports wizard allows you to create or edit a custom report. Custom Reports can include any metric collected by SQL diagnostic manager. This wizard allows you to choose the counters you want to include in your report, order the way the metrics appear, and specify the aggregation method used on each of your metrics. To open the Custom Reports wizard, click the New button in the Custom Report Options tree.
Add or edit a custom report

On the Add/Select a report page you can select whether you want to create a new report or want to edit an existing report.

**How do I create a new report?**

To create a new report, enter a name for the report and click Add.

**How do I edit an existing report?**

To edit an existing report, select the custom report from the list and click Next.
The Add Counters page allows you to select the counters you want to include in your report.

**TIP** Each Custom Report can include a maximum of five counters.

**How do I select the counters I want to include in my custom report?**

1. Check the box next to the type of counters you want to include with your report to view an alphabetical list of counters.
2. Select the counter from the list and click Add to include it in your report. Do this for each counter you want to include.
3. Check the Show tabular data in report box if you want to show tabular data.
4. Click Next.

**What types of counters can I include in my custom report?**

You can select from all the counters that SQL diagnostic manager is currently collecting.
The Order Counters for this report window allows you to specify where the counters you included in your custom report appear and how the metrics are aggregated.

**How do I change the order of the counters in my report?**
You can move the order of your counters in your report by selecting the Up and Down arrows next to each counter.

**How do I change the way my metrics are aggregated?**
You can change the way metrics are aggregated for each of the metrics you have included with your custom report by selecting from the following options in the Aggregation drop-down list:

- **Maximum**
  Displays the maximum value for the metric since the last collection.

- **Average**
  Displays the average per second value since the last collection.

- **Per Minute**
  Displays the average per minute value since the last collection.
The Administration blade of SQL diagnostic manager allows you to control access to all the information that SQLdm collects, allowing you to designate the servers and databases that each of your SQLdm users can view or edit. In addition to controlling access, you can also add custom counters for SQLdm to monitor from the Administration blade.
Use application security to manage SQLdm

In many SQL Server production environments, there are several different types of users, such as DBAs, Operators, Help Desk Users, Administrators, and Auditors to name a few. Within each of these groups, there may be several sub-groups of responsibility, such as a DBA that manages a certain set of high security databases that other DBAs do not. SQLdm allows you to designate the access both individual users and groups of users have within SQL diagnostic manager.

**TIP**

When Application Security is enabled, the only users who have access to SQLdm are those who belong to the sysadmin role on the SQLdm Repository. All other users must be added to SQLdm.

**What happens when I enable Application Security?**

Once you enable Application Security, you must add User Permissions for each user you want to give SQLdm access. Based on the assigned permissions, each user can only access a subset of monitored SQL Server instances. These security settings are enforced in the SQLdm Console, the Idera Newsfeed, and SQLdm Mobile.

Note that users who belong to the sysadmin role on the SQLdm Repository will always have Administrator powers.

**What are the actions available on the Application Security window?**

The Application Security window lists each login name and the type of access the login has to SQLdm. The following actions are available on the Application Security window:

- **Disable/Enable Application Security**
  
  You can toggle Application Security on and off using the associated icon. Disabling Application Security gives all SQLdm users full access to SQLdm.

- **Add SQLdm Permissions**
  
  You can open the Add SQLdm Permissions wizard by clicking the Add button on the Application Security window. This wizard allows you to add SQL Server Logins, Windows Users and Group accounts and assign them SQLdm view, modify, and administration privileges.

- **Edit Login Permissions**
  
  Once you add SQLdm Permissions, the Edit Login Permissions window allows you to edit login permissions.

- **Delete Login Permissions**
  
  Click the Delete button to remove the previously added login from SQLdm.

- **Enable/Disable Permissions**
  
  You can toggle non-system, SQLdm login permissions by checking and un-checking the associated box in the Enabled column.

- **Export SQLdm Permission Settings to Excel**
  
  You can export all your SQLdm Permission settings to Microsoft Excel by right-clicking any user in on the Application Security window and selecting Export to Excel.
The Add Permission wizard allows you give SQLdm access to a specified login in your environment. By default, all users belonging to the sysadmin role on the SQLdm Repository have SQLdm Administrator privileges and all other users are denied access to any of the monitoring and diagnostic views in SQLdm.

**TIP** Logins belonging to the sysadmin role on the SQLdm Repository have SQLdm Administrator privileges and cannot be edited or deleted.

Each login account can have multiple permissions associated with it. For example, if you want User A to be able to view all your SQL Server instances but modify only one of them, you need to create two permissions. The first login permission should include View permissions to allow the login to view all your SQL Server instances, while a second login permission needs to be created with Modify permissions enabled for the specified SQL Server instance.

**TIP** If a SQL login is entered and it does not already exist, a new login account is created on the SQLdm Repository SQL Server instance.

**To create a SQLdm login account:**

1. Click Administration > Application Security.
2. Click the Add button on the Application Security window.
3. Click Next on the Add Permission wizard Welcome window.
4. Type either the Windows account or SQL login used to connect to the SQLdm Repository.
5. Select the Authentication type and click Next.
6. Select the type of SQL Server instance permissions to grant to the specified account and click Next.
7. Select the SQL Server instances to apply the permissions to and click Add to move them to the Selected Servers column and click Next.
8. Review your changes on the Summary window and click Finish to add the SQLdm login.
The Specify Permissions window allows you to enter the login name and authentication type. Enter either the Windows account name or the SQL login used to connect to the SQLdm Repository. If the SQL login entered does not exist, one is created on the SQLdm Repository Server.

**TIP** If the Windows account entered is not a valid Windows account, an error appears when you exit the Add Permission wizard and the SQLdm login is not created.
Specify permissions

Select the type of permissions to grant to the specified account. When you add login permissions to SQLdm, you have the following permission choices:

**View Permissions**

Logins with View permissions can view the data collected for the assigned SQL Server instances except where noted in the following table.

TIP  Idera is deprecating the To Do list feature beginning with SQL diagnostic manager 7.0. While users who had To Do activated in an earlier 6.x version can continue to use To Do items, SQLdm 7.1 does not support this feature for new users. Current To Do users can reference Idera Solution for information on how to enable the To Do view.

<table>
<thead>
<tr>
<th>SQLdm Area</th>
<th>Cannot do the following</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Menu</td>
<td>Manage Servers</td>
</tr>
<tr>
<td>Tools Menu</td>
<td>Default Alert Configuration</td>
</tr>
<tr>
<td>Tools Menu</td>
<td>Edit Notification Rules and Providers</td>
</tr>
<tr>
<td>Tools Menu</td>
<td>Edit Grooming Options</td>
</tr>
<tr>
<td>Help Menu</td>
<td>Manage Licenses</td>
</tr>
<tr>
<td>Servers Pane</td>
<td>Delete Servers</td>
</tr>
<tr>
<td>Servers Pane</td>
<td>Enter or Exit Maintenance Mode</td>
</tr>
<tr>
<td>Servers Pane</td>
<td>Configure Alerts &lt;can view alert settings&gt;</td>
</tr>
<tr>
<td>Servers Pane</td>
<td>Edit SQL Server Properties &lt;can view properties&gt;</td>
</tr>
<tr>
<td>Today Page &gt; To Do</td>
<td>Set Status</td>
</tr>
<tr>
<td>Today Page &gt; To Do</td>
<td>Delete To Dos</td>
</tr>
<tr>
<td>Today Page &gt; To Do</td>
<td>Edit Properties &lt;can view properties&gt;</td>
</tr>
<tr>
<td>Today Page &gt; Active Alerts</td>
<td>Configure Alerts &lt;can view alerts&gt;</td>
</tr>
<tr>
<td>Today Page &gt; Common Tasks</td>
<td>Manage Servers</td>
</tr>
<tr>
<td>Mini-Monitors Grid</td>
<td>Delete Mini-Monitors</td>
</tr>
<tr>
<td>Mini-Monitors Grid</td>
<td>Edit Properties &lt;can view properties&gt;</td>
</tr>
<tr>
<td>Server Overview &gt; Dashboard</td>
<td>Enter or Exit Maintenance Mode</td>
</tr>
<tr>
<td>Server Overview &gt; Dashboard</td>
<td>Configure Alerts &lt;can view alert settings&gt;</td>
</tr>
<tr>
<td>Server Overview &gt; Details</td>
<td>Configure Alerts &lt;can view alert settings&gt;</td>
</tr>
<tr>
<td>Server Overview &gt; Configuration</td>
<td>Edit Value</td>
</tr>
<tr>
<td>Session &gt; Details</td>
<td>Trace or Kill Session</td>
</tr>
<tr>
<td>SQLdm Area</td>
<td>Cannot do the following</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>Session &gt; Locks</td>
<td>Trace or Kill Session</td>
</tr>
<tr>
<td>Session &gt; Blocking</td>
<td>Trace or Kill Session</td>
</tr>
<tr>
<td>Queries</td>
<td>Configure Query Monitor Properties &lt;can view properties&gt;</td>
</tr>
<tr>
<td>Resources &gt; Procedure Cache</td>
<td>Clear Cache</td>
</tr>
<tr>
<td>Databases &gt; Tables and Indexes</td>
<td>Update Statistics or Rebuild Indexes</td>
</tr>
<tr>
<td>Services &gt; Summary</td>
<td>Start or Stop Services</td>
</tr>
<tr>
<td>Services &gt; SQL Agent Jobs</td>
<td>Start or Stop Job</td>
</tr>
<tr>
<td>Logs</td>
<td>Cycle Server Log</td>
</tr>
<tr>
<td></td>
<td>Configure</td>
</tr>
<tr>
<td>Alerts Pane</td>
<td>Default Alert Configuration</td>
</tr>
<tr>
<td></td>
<td>Notification Rules and Providers</td>
</tr>
<tr>
<td>Alerts Pane &gt; Grid</td>
<td>Configure Alerts &lt;can view alerts&gt;</td>
</tr>
<tr>
<td></td>
<td>Clear Alert</td>
</tr>
<tr>
<td></td>
<td>Clear Alert of this Type for this Instance</td>
</tr>
<tr>
<td>To Do Pane</td>
<td>Notification Rules and Providers</td>
</tr>
<tr>
<td>To Do Pane &gt; Grid</td>
<td>Delete</td>
</tr>
<tr>
<td>Administration Pane</td>
<td>No access to Administration, is not displayed</td>
</tr>
</tbody>
</table>

**Modify Permissions**

Logins with Modify permissions can edit basic settings for the data collected for assigned SQL Server instances except where noted in the following table.

<table>
<thead>
<tr>
<th>SQLdm Area</th>
<th>Cannot do the following</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Menu</td>
<td>Manage Servers</td>
</tr>
<tr>
<td>Tools Menu</td>
<td>Edit Alert Configuration</td>
</tr>
<tr>
<td></td>
<td>Edit Notification Rules and Providers</td>
</tr>
<tr>
<td></td>
<td>Edit Grooming options</td>
</tr>
<tr>
<td>Help Menu</td>
<td>Manage Licenses</td>
</tr>
<tr>
<td>Alerts Pane</td>
<td>Edit Default Alert Configuration</td>
</tr>
<tr>
<td></td>
<td>Edit Notification Rules and Providers</td>
</tr>
<tr>
<td>To Do Pane</td>
<td>Edit Notification Rules and Providers</td>
</tr>
<tr>
<td>To Do Pane Grid</td>
<td>Delete</td>
</tr>
<tr>
<td>Administration Pane</td>
<td>No access to Administration, is not displayed</td>
</tr>
</tbody>
</table>
**Administration Permissions**

Logins with Administration permissions can access all monitored SQL Server instances and modify all settings including grooming options, notification rules and providers, manage SQLdm licenses, manage application security including the ability to add custom counters, and has the ability to add and remove SQL Server instances in SQLdm.

**TIP** Logins belonging to the sysadmin role on the SQLdm Repository have SQLdm Administrator privileges and cannot be edited or deleted.
The Select SQL Server window allows you to select the SQL Server instances to apply the permissions you set on the Specify User Permissions window. Select Server tags from the Tags drop-down list or choose New tag from the list. Select the SQL Server Instances you want the login to have permissions on and click Add button to move them to the Selected Servers column. Click Next to continue.
Review that all the Add Permission settings are correct and click Finish to apply your changes. If you want to make any changes to the settings displayed, click the Back button to go back and make your changes.
The Edit Permissions window allows you to change the permissions for the specified login.

**TIP** Logins belonging to the sysadmin role on the SQLdm Repository have SQLdm Administrator privileges and you cannot edit or delete the login.

**Enable/Disable Login Permissions**

You can toggle non-system, SQLdm login permissions by checking and clearing the associated check box in the Enabled column.

**Edit Permissions**

Select the appropriate permission level for the login. You can select View, Modify, or Administrator permissions.

**Enter Comments**

The Comments field allows you to enter additional information about the login. This information appears in the Comment column on the Application Security window.
The Assigned Servers window allows you to change the SQL Server instances on which the login has permissions. To tag a server instance, select Server tags from the Tags drop-down list or add a new tag by choosing New tag from the list. Select the SQL Server instances on which you want the login to have permissions and click Add button to move them to the Selected Servers column. Click Next to continue.
**Use PowerShell to automate SQLdm functions**

SQLdm PowerShell integration allows you to make mass changes to your SQLdm configuration across multiple SQL Server instances in a few lines of text, saving you time.

**What is PowerShell?**

Microsoft® Windows® PowerShell is a task-based command-line shell and scripting language designed especially for system administration. Built on the .NET Framework, Microsoft® Windows® PowerShell helps IT professionals and power users control and automate the administration of the Windows operating system and applications that run on Windows.

Windows PowerShell furnishes an operating environment for commands that include cmdlets, functions, filters, scripts, aliases, and executables (applications). The main command type used in this environment is the cmdlet, with certain cmdlets made available only through Windows PowerShell providers that allow access to stored data.

Windows PowerShell operates within a hosting application (the default is powershell.exe) that exposes a command line to the user, and uses a host interface to communicate with the commands invoked by the command line. The hosting application can be a console application, a Windows application, or a Web application. In most cases, the hosting application uses its Main function to interact with the Windows PowerShell runtime through the internal host interface; however, a hosting application can optionally support its own custom host by implementing the PSHost class along with one or more related user interface classes. Together, these classes allow direct communication between the application and Windows PowerShell commands.

For more information about Windows PowerShell, see SQL Server Books Online.

**How do I install the SQLdm Provider for PowerShell?**

The SQLdm Provider is installed automatically with SQLdm.

To enable the Provider, type the following from within PowerShell:

```
add-pssnapin sqldmsnapin
```

To verify your snapin has been registered:

```
get-psprovider
```

To receive general help:

```
get-help
```

To display a list of cmdlets for the SQLdm snapin:

```
get-help sqldm
```

**What applications are available to use with PowerShell?**

There are several applications that allow you to use PowerShell and enhance its functionality.

PowerShell Plus is a powerful interactive scripting environment designed to help administrators and developers learn and master Windows PowerShell. Unlike most of the basic script editors on the market today, PowerShell Plus features a powerful interactive console, an advanced script editor and a comprehensive learning center. PowerShell Plus is built on the backbone of PowerShellIDE, the first PowerShell application to offer a very rich graphical user interface.

**Interactive PowerShell Console**

The PowerShellPlus Console allows you to work interactively with PowerShell from a feature rich Windows UI. This integration makes working with PowerShell faster and easier to use than ever before.
**Advanced Script Editor**

The advanced debugger and script editor lets you build and test complex PowerShell scripts, try one line PowerShell commands from an embedded console window, and sign your script with a security certificate, all from a single workspace.

**Comprehensive Learning Center**

The Comprehensive Learning Center helps you experience PowerShell by example. Short tutorials guide you through basic concepts at your own pace. The Comprehensive Learning Center also includes dynamically created help topics from currently installed PowerShell CmdLets, Snap-Ins and WMI objects.
The following cmdlets are available for automating the administration of your SQLdm deployment.

<table>
<thead>
<tr>
<th>Cmdlet Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get-SQLServers</td>
<td>Retrieve a list of SQL Servers on the network</td>
</tr>
<tr>
<td>Escape-SQLdmName</td>
<td></td>
</tr>
<tr>
<td>Grant-SQLdmPermissions</td>
<td>Grant SQLdm permissions on a server</td>
</tr>
<tr>
<td>Revoke-SQLdmPermission</td>
<td>Revoke SQLdm permissions on a server</td>
</tr>
<tr>
<td>New-SQLdmMonitoredInstance</td>
<td>Add a SQL Server Instance to be monitored by SQLdm</td>
</tr>
<tr>
<td>Set-SQLdmMonitoredInstance</td>
<td>Configure options on your SQL Server Instance</td>
</tr>
<tr>
<td>Remove-SQLdmMonitoredInstance</td>
<td>Remove a SQL Server Instance from SQLdm monitoring</td>
</tr>
<tr>
<td>Set-SQLdmAppSecurity</td>
<td>Enable or disable Application Security for SQLdm</td>
</tr>
<tr>
<td>New-SQLdmUser</td>
<td>Create a new SQLdm user</td>
</tr>
<tr>
<td>Set-SQLdmUser</td>
<td>Modify an existing SQLdm user</td>
</tr>
<tr>
<td>Remove-SQLdmUser</td>
<td>Remove an existing SQLdm user</td>
</tr>
<tr>
<td>New-SQLdmDrive</td>
<td>Create a drive for connecting to the SQLdm Repository</td>
</tr>
</tbody>
</table>
Manage your SQLdm license

SQLdm requires a license key. This key allows you to access all SQLdm features, including the Idera Newsfeed and SQLdm Mobile.

Though you can immediately begin using SQLdm with the included evaluation key, consider updating the evaluation license as soon as possible. A registered license key allows you to unlock the evaluation time and registered instance limits so you can begin using SQLdm to the fullest extent.

SQLdm provides an intuitive interface for license key management. You can view the status of your license keys and add licenses to monitor additional instances. Each license allows you to monitor a predetermined number of SQL Server instances.

When you reach your license limit, SQLdm does not let you add new instances. When your trial period expires, SQLdm ceases to run.

To open the License Keys window, select Help > Manage Licenses from the Toolbar menu.
SQLdm includes a limited-time trial account with every installation. As the end of the trial period nears, SQLdm will warn you each time you start the console that your trial period is ending. At the end of this trial period, if the license has not been upgraded, SQLdm will stop functioning.

**How many servers can I register when using a Trial license?**

SQLdm limits the number of monitored SQL Server instances to 15 when using a trial license.

**How do I upgrade my Trial license?**

1. Contact Idera Sales to upgrade your SQLdm trial license.
2. Select Help > Manage Licenses.
3. Enter the new license key in the New Key section and click Enter. When prompted to type or copy to replace the current key, click Yes to continue.
Add a new license

If you would like to monitor additional SQL Server instances beyond the limits of your existing license, contact Idera Sales to purchase additional licenses.

How are my licenses added?

Once you have received a new license, open the License Keys window (Help > Manage Licenses from the Toolbar menu) and select Add to enter the new license. Your new license will list the number of additional SQL Server instances that can be monitored by SQLdm. Each new license you add to SQLdm adds to the total number of SQL Server instances that can be monitored.

How do I get a new license?

To get a new license, contact your Idera Sales representative.
The Manage Servers window (File>Manage Servers) allows you to see all the SQL Server instances monitored by SQLdm. You can:

- Add new SQL Server instances to SQLdm
- Edit the SQLdm configuration for a particular SQL Server instance
- Remove a SQL Server instance from SQLdm
- Apply an existing alert template to a server or group of servers
- Test the connection status of your SQL Server instances
The SQLdm Management Service Configuration wizard allows you to change the way the SQLdm Management Service interacts with the SQLdm Repository.

**When would I want to use the Management Service Configuration wizard?**

You can edit the SQLdm Management Service settings whenever either the location of your SQLdm Repository has changed or if you want to change the account or authentication method the SQLdm Management Service uses to connect to the SQLdm Repository.

**How do I use the Management Service Configuration wizard?**

The SQLdm Management Service Configuration wizard is accessed by running the `SQLdmManagementServiceConsole.exe` file in your root SQLdm directory.

**To use the SQLdm Management Service Configuration wizard:**

1. Run `SQLdmManagementServiceConsole.exe` in your root SQLdm directory.
2. Click Next on the Welcome window.
3. Enter your Repository information.
4. Enter the Authentication method the SQLdm Management Service should use to write configuration and collected statistics to the SQLdm Repository.
5. Click Test to verify that the connection can be established.
6. Click Next.
7. Verify the changes are correct.
8. Click Finish to apply the changes.
SQLdm collects a number of SQL Server and operating system performance metrics to help you monitor, alert, and report on your system health. SQLdm includes the following metrics for your use:

- Batches Received (Per Second)
- Blocked Sessions (Count)
- Buffer Cache Hit Ratio (Percent)
- Client Computers
- Compiles (Per Second)
- Data Used (Percent)
- Database Status
- Disk Queue Length
- Disk Time (Percent)
- I/O Waits
- Lock Waits
- Log Flushes
- Log Space Used (Percent)
- Memory Page Faults (Per Second)
- Memory Paging
- Memory Waits
- Other Waits
- Packets Received (Per Second)
- Packets Sent and Received
- Packets Sent (Per Second)
- Page Reads (Per Second)
- Page Writes (Per Second)
- Percent of SQL Server Data Space Used
- Percent of SQL Server Log Space Used
- Physical Page Reads and Writes
- Processor Queue Length
- Re-compiles Per Second
- Signal Waits
- SQL Server Batches Received
- SQL Server CPU Usage
- SQL Server Log Space Used (Percent)
- SQL Server Memory Usage
- SQL Server Response Time
- Total Memory Usage
- Transaction Log Waits
- Transactions (Per Second)
- Wait type category - Backup
- Wait type category - I/O Page Latch
- Wait type category - Lock
- Wait type category - Memory
- Wait type category - Non-I/O Page Latch
- Wait type category - Non-Page Latch
- Wait type category - Other
- Wait type category - Transaction Log
This metric counts the total number of batches requests that the SQL Server computer is receiving per second. This rate is a measurement of throughput on the processor.

This metric uses the counter Batch requests/sec from the SQL Server SQL Statistics counter object. On SQL Server 2000 instances, this counter is available through sysperfinfo. On SQL Server 2005 or later instances, this counter is available through sys.dm_os_performance_counters.

To track this metric, use the CPU Call Rates chart in the CPU dashboard of the Server Overview tab.

While this counter has no good or bad value, each of these T-SQL batches has to be:

1. Transmitted to the server using the network.
2. Parsed by SQL Server to check syntax.
3. Have an execution plan compiled.
4. Executed according to the execution plan.
5. Have the results transmitted back to the client computer or Web site.

Therefore, the number of T-SQL batches needs to be both minimized in number and the impact of each reduced as much as possible by:

- Reducing the size and complexity of the T-SQL by creating a stored procedure containing the commands you want executed and then simply calling it in the T-SQL batch. This approach will reduce the size and complexity of the T-SQL batch, which means less network traffic as well as less CPU time consumed producing a complex execution plan.

- Having moved most of the T-SQL code into stored procedures, consider grouping as many of those calling T-SQL commands into a single network batch as possible. Also consider redesigning the application to call only those stored procedures you need to complete an application function all from within one batch (rather than sending each EXECUTE statement to execute separately in its own batch). This strategy will reduce network queues dramatically as better use is made of the default network packet size (which is 4 KB).
This metric counts the number of sessions on the SQL Server instance being blocked by other sessions holding requested locks.

By default, a blocked session will wait forever. This default behavior can be changed by adding `SET LOCK_TIMEOUT 60000` to the top of the first T-SQL batch after a connection is made.
This metric calculates the percentage of data pages SQL Server found in memory without having to resort to reading them from the disk subsystem. This value is measured by reading the 'buffer cache hit ratio' from `sysperfinfo`. The higher this value is, the more efficiently the system is running.

**What can I do to fix this problem?**

If values lower than 85% regularly occur (once SQL Server has been up for at least 30 minutes), consider the following solutions:

- Add more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures and they are only called very occasionally, then after calling them, issue a `DBCC dllname (FREE)` command. Once an extended stored procedure has been called, it remains in memory until SQL Server is shut down even if it is never called again! This is tremendously wasteful in terms of available memory. `DBCC dllname (FREE)` will release that memory for use by both the procedure cache as well as the Data Pages which will have a significant positive effect on both the Procedure Cache Hit Ratio as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server execute `SP_HELPEXTENDEDPROC`.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- Stopping unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.
- If the computer is running multiple instances of SQL Server, consider placing each instance on a separate physical computer.
This metric calculates the number of distinct connected network adapters currently using the SQL Server instance. This value is measured by getting the count of processes from distinct computers in `sysprocesses` table.
This metric calculates the number of SQL compilations per second, including statement-level re-compilations.

This metric uses the counter `SQL Compilations/sec` from the SQL Server SQL Statistics counter object. On SQL Server 2000 instances, this counter is available through `sysperfinfo`. On SQL Server 2005 or later instances, this counter is available through `sys.dm_os_performance_counters`.

A high number of compilations (greater than 100 per second) can indicate a high server workload or that there is more re-compilation overhead than necessary. Too many compilations will impact a server’s performance.
Data Used (Percent)

This metric calculates the size of all data files used as a percent of the added allocated size of all data files.

In a typical medium to large scale OLTP environment (with a data file size of over 50 GB), this value should slowly increase over time in a linear fashion as history tables (if there are any) are populated.

What can I do to reduce this problem?

If your site experiences a sudden sharp increase in this value of over 10% in an hour, it should be investigated and fully understood as the underlying causes can often lead to poor performance. Possible causes include:

- A large number of rows have been inserted into a table. If so, then make sure that the table has not become fragmented due to this operation (run `DBCC SHOWCONTIG` to check).
- An index with very poor cardinality (uniqueness) has been added to a large table. Indexes with poor cardinality can often exceed the size of the underlying table itself and in almost all cases will not be used by any SQL Server query making them a useless waste of space.
- A large number of rows were incorrectly changed by performing an insert of a new row followed by a delete of the old row. Even though the net effect of the change is essentially the same number of rows; the database would have grown significantly in size and a `DBCC SHRINKDATABASE` will have to be issued. It is likely that the table is also fragmented because the inserts were done before the deletes so run `DBCC SHOWCONTIG`. Always delete the old rows out of a table first before inserting the new rows; otherwise, tables will become fragmented more quickly, databases will increase in size, and the performance of the transactions will be degraded due to the large number of page splits the table had to undergo.
Database Status

This metric indicates the operational status of the database, such as Normal, Offline, or Suspect. You can select the database status about which you want to be alerted.
Disk Queue Length

This metric counts the average number of system requests that are waiting for disk access on the computer hosting the SQL Server instance. The Disk view on the Resources tab allows you to track your Disk Queue Length, along with other key disk statistics, over a period of time on the computer hosting your SQL Server instance.

Why does this metric have no value?

SQLdm has not been able to collect the corresponding counter due to one of the following reasons:

- It usually takes 2 refreshes for the CPU Usage metric to begin populating. This usually occurs within a few seconds. Another possible cause could be that either OLE Automation is not working properly or Lightweight Pooling is enabled on the SQL Server instance.
- OLE Automation must be enabled for SQLdm to collect hardware performance metrics. To enable OLE Automation, click the warning message below the ribbon menu on the SQLdm Console.
- SQLdm cannot collect OS metrics if Lightweight Pooling has been enabled on the SQL Server instance.
- The counter does not exist on this SQL Server instance.
Disk Time (Percent)

This metric calculates the percentage of elapsed time during which all the disks were busy servicing read and write requests on the SQL Server computer. This counter is measured by subtracting the Disk Idle Time percent from 100. The WMI object Win32_PerfRawData_PerfDisk_PhysicalDisk property PERCENTIDLETIME is measured to get Disk Idle Time percent.

What can I do to reduce Disk Time?

If this metric value regularly exceeds 75%, consider the following solutions:

- Check OS Paging to make sure that paging from/to the swap file is not causing these IOs.
- Distribute the data and logs over more disks.
- Replace disks with faster disks.
- Add more memory if the main cause is Disk Reads.
- Switch from a RAID 5 to RAID 10 solution if Disk Writes make up more than 20% of total disk access.
- Move non-SQL Server applications to another computer.
- If the computer is running multiple instances of SQL Server, consider placing each instance on a separate computer.
I/O Waits

This metric calculates the length of time that SQL Server processes had to wait for I/O to become available. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.
This metric calculates the length of time that SQL Server processes had to wait for a resource lock to resolve. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.
Log Flushes (Per Second)

This metric reads **LOG FLUSHES/SEC** from `sysperfinfo`. Each time data is changed, the changes are written to the log cache, which resides in memory. Under certain conditions, this log cache is flushed to disk. Some causes are:

- An explicit or implicit transaction is committed or rolled back.
- A `CHECKPOINT` statement is executed.
- SQL Server issues an automatic checkpoint.
- The Log Cache becomes full and needs to be emptied.

**What can I do to reduce log flushes?**

On an OLTP system that is running relatively small transactions (on average, less than 40 rows are added, changed, or deleted per transaction), the number of Log Flushes should not exceed SQL Batches by more than 40% during periods of high activity. If they do, then one of the following abnormal conditions exists:

- T-SQL batches are being sent through for execution which contain a number of individual data manipulation statements (inserts, deletes, or updates) which are not enclosed within a transaction.
- Large batch jobs which update significant amounts of data are being run during the day.
- A number of transactions are executing a `ROLLBACK TRANSACTION` statement or are being rolled back due to either Deadlocks or Lock Timeouts.
- The Recovery Interval parameter is set too low. Raise it to a value between 10-15 minutes.
- An explicit `CHECKPOINT` statement is being executed within some transactions.
- If the entire contents of a table need to be deleted; rather than using a `DELETE`, use the `TRUNCATE TABLE` statement as it produces far fewer log flushes.

If you have implemented the above measures and on a busy system the Log Flushes still greatly exceed 140% of SQL Batches then in order to minimize their impact consider:

- Switching your RAID array from a RAID 5 to a RAID 10 solution as each write IO results in two writes using RAID 10 vs. 4 for RAID 5 (100% more efficient with writes).
- So long as the RAID controller has some form of battery backup, then switch its cache mode from Write-through to Write-back, as this increases the system's ability to handle write IOs by an order of magnitude.
- Adding more cache memory to the RAID controller.
This metric calculates the page fault rate for all processes on the SQL Server computer. The WMI object Win32_PerfRawData_PerfOS_Memory property PagesPersec is measured to get this value. For optimal performance, the page fault rate should either be either 0 or very low.

What can I do to reduce memory page faults?

If the page fault rate rises above 100 for any period of time longer than a momentary spike, the following steps need to be taken to reduce it:

- Decrease the memory being used by stopping unnecessary programs such as the Print Spooler from running on the dedicated SQL Server computer. You should also eliminate activities such as the server acting as a PDC or BDC and (as a last resort) reducing the amount of memory SQL Server may consume.
- Add more physical memory on the computer.
- If the computer is running multiple instances of SQL Server, then consider moving each instance to a separate physical computer.
Memory Paging

This metric counts the number of pages being swapped in and out of memory per second. The Memory view on the Resources tab allows you to track paging and other key memory statistics over a period of time.

Why does this metric have no value?

SQLdm has not been able to collect the corresponding counter due to one of the following reasons:

- It usually takes 2 refreshes for the CPU Usage metric to begin populating. This usually occurs within a few seconds. Another possible cause could be that either OLE Automation is not working properly or Lightweight Pooling is enabled on the SQL Server instance.
- OLE Automation must be enabled for SQLdm to collect hardware performance metrics. To enable OLE Automation, click the warning message below the ribbon menu on the SQLdm Console.
- SQLdm cannot collect OS metrics if Lightweight Pooling has been enabled on the SQL Server instance.
- The counter does not exist on this SQL Server instance.
This metric calculates the length of time that SQL Server processes had to wait for SQL Server memory to become available. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.
This metric includes any additional waits occurring on this SQL Server outside of the following wait categories:

- I/O
- Locks
- Memory
- Signal

Waits are a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload. To get more details about monitored wait statistics, see the Server Waits view, and review the SQL Server 2005 Waits and Queues SQL Server Best Practices Article (http://technet.microsoft.com).
This metric is calculated as a rate between two refreshes of the value of the system statistical function `@@PACK_RECEIVED`. If this value is consistently high, consider changing the packet size.
This metric counts the number of packets being sent and received, per second, on your SQL Server instance.
Packets Sent (Per Second)

This metric is calculated as a rate between two refreshes of the value of the system statistical function @@PACK_SENT. If this value is consistently high, consider changing the packet size.
This metric counts the number of physical reads the disk sub-system is performing to bring SQL Server database pages into the data (buffer) cache. For the first few minutes after SQL Server is started / re-started almost any request for data will have to be retrieved from disk but, as the data cache is filled, these requests should be reduced. This value is measured by reading 'PAGE READS/SEC' from sysperfinfo.

Even the fastest disks can only support a maximum sustained rate of around 85 random IOs per second, so in order to determine if your IO subsystem is being "flooded", perform the following calculation:

On RAID 10: \( \frac{(\text{Page reads} + \text{read ahead pages} + ((\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 2))}{\text{number of disk spindles in RAID array}} \) must be < 65

What can I do to reduce page reads?

If the page reads rate exceeds 65 for more than five seconds every 20 minutes, then consider these solutions:

- Adding more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures that are not called very often, then, after calling them, issue a DBCC DLLNAME (FREE). Once an extended stored procedure has been called, it remains in memory until SQL Server is shut down, even if the command is never called again. This is tremendously wasteful in terms of available memory and a DBCC DLLNAME (FREE) will release that memory for use by both the procedure cache as well as the Data Pages which will have a significant positive effect on both the Procedure Cache Hit Ratio as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server, execute SP_HELP-EXTENDEDPROC.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- If large batch style reporting queries are being run on the OLTP server, then large numbers of data pages which most of the OLTP transactions read are being flushed from cache causing much additional IO overhead. In this case, set up a separate SQL Server Reporting Services computer (Report Server) to service all but the smallest reports. Data can be sent to this Report Server through backups and Log Shipping from the OLTP server.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.
- If the computer is running multiple instances of SQL Server, consider placing each instance on a separate physical computer.
This metric counts the number of physical database writes the disk sub-system is performing. This value is measured by reading \texttt{PAGE READS/SEC} from \texttt{sysperfinfo}.

Even the fastest disks can only support a maximum sustained rate of around 85 random IOs per second, so in order to determine if your IO subsystem is being "flooded", perform the following calculation:

- On RAID 5: $\frac{\text{Page reads} + \text{read ahead pages} + ((\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 4)}{\text{number of disk spindles in RAID array}}$ must be $< 65$
- On RAID 10: $\frac{\text{Page reads} + \text{read ahead pages} + ((\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 2)}{\text{number of disk spindles in RAID array}}$ must be $< 65$

### What can I do to reduce page writes?

If the page writes rate exceeds 65 for more than five seconds every 20 minutes, then consider the following:

- Add more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures that are not called very often, then after calling them, issue a \texttt{DBCC DLLNAME (FREE)}. Once an extended stored procedure has been called, it remains in memory until SQL Server is shut down even if it is never called again! This is tremendously wasteful in terms of available memory and a \texttt{DBCC DLLNAME (FREE)} will release that memory for use by both the procedure cache as well as the Data Pages which will have a significant positive effect on both the Procedure Cache Hit Ratio as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server, execute \texttt{SP_HELPEXTENDEDPROC}.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- If large batch style reporting queries are being run on the OLTP server, then large numbers of data pages which most of the OLTP transactions read are being flushed from cache causing much additional IO overhead. In this case set up a separate SQL Server Reports computer to be used to service all but the smallest reports. This Reports Server can be fed via backups and Log Shipping from the OLTP server.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.
This metric calculates the percentage of the allocated disk space for databases that is being used by the SQL Server instance. The total size allocated to the databases is the sum of unused space, data space, and index space. Data space includes leaf level clustered index pages, since these pages are the data pages. Index space excludes the leaf pages of clustered indexes. The Files view on the Databases tab allows you to view detailed data file information including the data size and the percentage of allocated file size being used on each of your databases.
## Percent of SQL Server Log Space Used

This metric calculates the percentage of the allocated space for logs that is currently being used. The Files view on the Databases tab allows you to view detailed log information including the log size and the percentage of allocated file size being used on each of your databases.
This metric counts the number of physical page reads and writes to disk from the SQL Server instance.
This metric counts the number of ready threads in the processor queue on the computer that hosts the SQL Server instance. The WMI object Win32_PerfRawData_PerfOS_System property ProcessorQueueLength is measured to get processor queue length count.
This metric calculates the rate of re-compiles on your monitored instance, using the counter SQL Re-Compilations/sec from the SQL Server SQL Statistics counter object.

On SQL Server 2000 instances, this counter is available through `sysperfinfo` and represents the rate of batch re-compilations per second.

On SQL Server 2005 or later instances, this counter is available through `sys.dm_os_performance_counters` and represents the rate of statement-level re-compilations per second.

Typically, this metric value is low. A high re-compilation rate may be a sign of excessive re-compilation overhead. Re-compiles can be caused by various events, including schema changes, statistics updates, and explicit calls to `WITH RECOMPILE`. Excessive re-compilation will impact a server’s performance.
Signal waits

This metric calculates the length of time that SQL Server processes had to wait for CPU time after the associated resources became available for processing. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.
This metric counts the number of T-SQL batches that SQL Server is receiving for execution every second. This value is measured by reading ‘batch requests/sec’ from sysperfinfo. While this counter has no good or bad value, each of these T-SQL batches has to be:

1. Transmitted to the server using the network.
2. Parsed by SQL Server to check syntax.
3. An execution plan compiled.
4. The execution plan executed.
5. The results transmitted back to the client computer or Web.

The number of T-SQL Batches need to be both minimized in number and the impact of each reduced as much as possible by:

- Reducing the size and complexity of the T-SQL by creating a stored procedure containing the commands you want executed and then simply calling it in the T-SQL batch. This will reduce the size and complexity of the T-SQL batch, which means less network traffic as well as less CPU time consumed producing a complex execution plan.
- Having moved most of the T-SQL code into stored procedures, it is then a good idea to group as many of those calling T-SQL commands into a single network batch as possible. Redesign the application to call whatever stored procedures you need to complete an application function all from within one batch (rather than sending each EXECUTE statement to execute separately in its own batch). This will reduce network queues dramatically as better use is made of the default network packet size (which is 4 KB).
This metric calculates the percentage of CPU time used by the SQL Server instance. This value is calculated by the percentage of time in a query spent by the I/O, idle, and CPU times on the computer that hosts the SQL Server instance.

To track this metric, use the CPU Usage chart in the CPU dashboard of the Server Overview tab.

**What can I do to lower SQL Server CPU Usage?**

If this value regularly exceeds 75%, consider taking the following actions:

- Reduce the number of SQL re-compilations, as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and most of these reasons can be removed by careful coding.
- Ensure that all T-SQL statements (whether in a Stored Procedure, Trigger, or Ad Hoc statement) that reference objects fully qualify the object referenced.

  For example: `SELECT * FROM Northwind.dbo Employees` is a fully qualified object reference whereas `SELECT * FROM Employees` is a poorly qualified object. The execution plans of fully qualified objects can be reused as is, whereas plans where the object are not fully qualified either cannot be reused at all or if they are then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. In both cases, a significant amount of CPU time is consumed.
This metric calculates the added size all log files used as a percent of summed allocated size of all data files.

A transaction log will continue to grow until its inactive portion is truncated. Transaction log truncation can occur due to two causes:

- At the successful conclusion of a BACKUP LOG statement.
- If the database is using the Simple Recovery Model, then whenever a CHECKPOINT T-SQL statement is executed or whenever SQL Server performs an internal automatic CHECKPOINT (which it will do if the log reaches 70% full or when SQL Server determines the length of time to recover from the log, in the event of a failure, exceeds the Recovery Interval option).

Only the inactive portion of the log is ever removed. The inactive portion of the log is the portion of the log up to but not including the start date/time of the oldest open transaction. So no matter which method is used to truncate the log, if there is a perpetually open transaction then the log will continue to grow in size until it has consumed all available disk space.

**What can I do to reduce the percentage of the log used?**

If your transaction log size continues to grow despite issuing BACKUP LOG or CHECKPOINT statements, you have a rogue transaction which can be located by sorting the Last Batch column on the SQL Processes grid into ascending mode such that the processes having the oldest batches are shown first. Then, simply scroll down the list until you find the first user process with a non-zero number of transactions in its Trans column, and kill this process. Once this is done, then perform a CHECKPOINT or BACKUP LOG and the Log File Used Size will be reduced.
This metric calculates the percentage total memory being used on the computer hosting the selected instance by SQL Server. The Memory view on the Resources tab allows you to track the SQL Server memory usage, along with other key memory statistics, on the computer hosting your SQL Server instance.
This metric calculates the time (in milliseconds) that SQL diagnostic manager currently needs to send a simple SQL command to
the SQL Server instance, have it processed, and receive the returned result set.

Typically, this value should average below 2000 ms for a well tuned network and server. Average values in excess of 2000 ms
indicate either an excessively busy network segment or a stressed SQL Server.
This metric calculates the percentage of physical memory and virtual memory Windows is using. The Memory view on the Resources tab allows you to track the Total Memory usage along with other key memory statistics on the computer hosting your SQL Server instance.
This metric calculates the length of time that SQL Server processes had to wait for the transaction log to become available. This rate is a measurement of throughput on the SQL Server computer processor, as determined by the SQL Server workload.
This metric calculates the total number of transactions per second across all databases on the monitored instance.

This metric uses the counter `Transactions/sec` from the SQL Server `Databases` counter object and `_Total` instance. On SQL Server 2000 instances, this counter is available through `sysperfinfo`. On SQL Server 2005 or later instances, this counter is available through `sys.dm_os_performance_counters`.

This rate is a measurement of throughput on the processor, as determined by the SQL Server workload. Although this rate measures activity within a transaction, and a higher transaction rate indicates a higher risk of resource issues, it does not account for all activity on the SQL Server computer.
The Backup wait type category includes all the waits that are associated with bottlenecks caused during a backup process, including the following:

- BACKUP
- BACKUP_CLIENTLOCK
- BACKUP_OPERATOR
- BACKUPBUFFER
- BACKUPIO
- BACKUPTHREAD
- DISKIO_SUSPEND
The I/O Page Latch wait type category includes all the waits that are associated with non-I/O page latch bottlenecks, including the following:

- ASYNC_IO_COMPLETION
- IO_COMPLETION
- PAGEIOLATCH_DT
- PAGEIOLATCH_EX
- PAGEIOLATCH_KP
- PAGEIOLATCH_NL
- PAGEIOLATCH_SH
- PAGEIOLATCH_UP
The Lock wait type category includes all the waits that are caused when a task is waiting to acquire a lock, including the following:

- LCK_M_BU
- LCK_M_IS
- LCK_M_IU
- LCK_M_IX
- LCK_M_RIn_NL
- LCK_M_RIn_S
- LCK_M_RIn_U
- LCK_M_RIn_X
- LCK_M_RS_S
- LCK_M_RS_U
- LCK_M_RX_S
- LCK_M_RX_U
- LCK_M_RX_X
- LCK_M_S
- LCK_M_SCH_M
- LCK_M_SCH_S
- LCK_M_SIU
- LCK_M_SIIX
- LCK_M_U
- LCK_M_UIX
- LCK_M_X
- XACTLOCKINFO
The Memory wait type category includes all the waits that are associated with memory bottlenecks, including the following:

- CMEMTHREAD
- RESOURCE_SEMAPHORE
- RESOURCE_SEMAPHORE_MUTEX
- RESOURCE_SEMAPHORE_QUERY_COMPILE
- RESOURCE_SEMAPHORE_SMALL_QUERY
- SOS_RESERVEDMEMBLOCKLIST
The Non-I/O Page Latch wait type category includes all the waits that are associated with non-I/O page latch bottlenecks, including the following:

- PAGELATCH_DT
- PAGELATCH_EX
- PAGELATCH_KP
- PAGELATCH_NL
- PAGELATCH_SH
- PAGELATCH_UP
The Non-Page Latch wait type category includes all the waits that are caused when a task is waiting for a latch that is not related to paging, including the following:

- LATCH_DT
- LATCH_EX
- LATCH_KP
- LATCH_NL
- LATCH_SH
- LATCH_UP
- TRAN_MARKLATCH_DT
- TRAN_MARKLATCH_EX
- TRAN_MARKLATCH_KP
- TRAN_MARKLATCH_NL
- TRAN_MARKLATCH_SH
- TRAN_MARKLATCH_UP
This wait type category includes all the waits that do not belong to the Memory, I/O, Lock, Transaction Log, Non-I/O Page Latch, Non-Page Latch, or Backup categories.

The following waits are associated with the Other category:

- ABR
- ASSEMBLY_LOAD
- AUDIT_GROUPCACHE_LOCK
- AUDIT_LOGINCACHE_LOCK
- AUDIT_ON_DEMAND_TARGET_LOCK
- AUDIT_XE_SESSION_MGR
- BUILTIN_HASHKEY_MUTEX
- CHECK_PRINT_RECORD
- CLEAR_DB
- CLR_AUTO_EVENT
- CLR_CRST
- CLR_JOIN
- CLR_MANUAL_EVENT
- CLR_MEMORY_Spy
- CLR_MONITOR
- CLR_RWLOCK_READER
- CLR_RWLOCK_WRITER
- CLR_TASK_START
- CLRHOST_STATE_ACCESS
- COMMIT_TABLE
- CURSOR_ASYNC
- CXROWSET_SYNC
- DAC_INIT
- DBMIRROR_DBM_EVENT
- DBMIRROR_DBM_MUTEX
- DBMIRROR_SEND
- DBMIRROR_WORKER_QUEUE
- DBMIRRORING_CMD
- DBTABLE
- DEADLOCK_ENUM_MUTEX
- DEADLOCK_TASK_SEARCH
- DEBUG
- DISABLE_VERSIONING
- DISPATCHER_QUEUE_SEMAPHORE
- DLL_LOADING_MUTEX
- DROPTEMP
○ DTC
○ DTC_ABORT_REQUEST
○ DTC_RESOLVE
○ DTC_STATE
○ DTC_TMDOWN_REQUEST
○ DTC_WAITFOR_OUTCOME
○ DUMP_LOG_COORDINATOR
○ DUMP_LOG_COORDINATOR_QUEUE
○ DUMPTRIGGER
○ EC
○ EE_PMOLOCK
○ EE_SPEC_PROC_MAP_INIT
○ ENABLE_VERSIONING
○ ERROR_REPORTING_MANAGER
○ EXECSYNC
○ EXECUTION_PIPE_EVENT_INTERNAL
○ FAILPOINT
○ FCB_REPLICA_READ
○ FCB_REPLICA_START
○ FCB_REPLICA_WRITE
○ FS_FC_RWLOCK
○ FS_GARBAGE_COLLECTOR_SHUTDOWN
○ FS_HEADER_RWLOCK
○ FS_LOGTRUNC_RWLOCK
○ FSA_FORCE_OWN_XACT
○ FSAGENT
○ FSTR_CONFIG_MUTEX
○ FSTR_CONFIG_RWLOCK
○ FT_COMPROWSET_RWLOCK
○ FT_IFTS_RWLOCK
○ FT_IFTSHC_MUTEX
○ FT_IFTSISM_MUTEX
○ FT_MASTER_MERGE
○ FT_METADATA_MUTEX
○ FT_RESTART_CRAWL
○ FT_RESUME_CRAWL
○ FULLTEXT_GATHERER
○ GUARDIAN
○ HTTP_ENDPOINT_COLLDELETE
○ HTTP_ENUMERATION
○ HTTP_START
○ IMP_IMPORT_MUTEX
- IMPROV_IOWAIT
- INDEX_USAGE_STATS_MUTEX
- INTERNAL_TESTING
- IO_AUDIT_MUTEX
- IO_RETRY
- IOAFF_RANGE_QUEUE
- KTM_ENLISTMENT
- KTM_RECOVERY_MANAGER
- KTM_RECOVERY_RESOLUTION
- LOGGENERATION
- LOGMGR
- LOGMGR_FLUSH
- LOGMGR_QUEUE
- LOGMGR_reserve_APPEND
- LOWFAIL_MEMMGR_QUEUE
- METADATA_LAZYCACHE_RWLOCK
- MIRROR_SEND_MESSAGE
- MSQL_DQ
- MSQL_SYNC_PIPE
- MSQL_XACT_MGR_MUTEX
- MSQL_XACT_MUTEX
- MSQL_XP
- MSSEARCH
- NET_WAITFOR_PACKET
- NODE_CACHE_MUTEX
- OLEDB
- PARALLEL_BACKUP_QUEUE
- PERFORMANCE_COUNTERS_RWLOCK
- PLACEHOLDER1
- PLACEHOLDER2
- PRINT_ROLLBACK_PROGRESS
- QNMGR_MANAGER_ACQUIRE
- QPJOB_KILL
- QPJOB_WAITFOR_ABORT
- QRY_MEM_GRANT_INFO_MUTEX
- QUERY.ERRHDL_SERVICE_DONE
- QUERY_EXECUTION_INDEX_SORT_EVENT_OPEN
- QUERY_NOTIFICATION_MGR_MUTEX
- QUERY_NOTIFICATION_SUBSCRIPTION_MUTEX
- QUERY_NOTIFICATION_TABLE_MGR_MUTEX
- QUERY_NOTIFICATION_UNITTEST_MUTEX
- QUERY_OPTIMIZER_PRINT_MUTEX
- QUERY_TRACEOUT
- QUERY_WAIT_ERRHDLD_SERVICE
- RECOVER_CHANGEDB
- REPL_CACHE_ACCESS
- REPL_HISTORYCACHE_ACCESS
- REPL_SCHEMA_ACCESS
- REPL_TRANHASHTABLE_ACCESS
- REPLICA_WRITES
- REQUEST_DISPENSER_PAUSE
- RESMGR_THROTTLED
- RFS_MSGPUMP
- RFS_NOTIFYTHREAD
- RFS_STARTUP
- RG_RECONFIG
- SEC_DROP_TEMP_KEY
- SECURITY_MUTEX
- SEQUENTIAL_GUID
- SERVER_IDLE_CHECK
- SHUTDOWN
- SNI_CRITICAL_SECTION
- SNI_HTTP_ACCEPT
- SNI_HTTP_WAITFOR_0_DISCON
- SNI_LISTENER_ACCESS
- SNI_TASK_COMPLETION
- SOAP_READ
- SOAP_WRITE
- SOS_CALLBACK_REMOVAL
- SOS_DISPATCHER_MUTEX
- SOS_LOCALALLOCATORLIST
- SOS_MEMORY_USAGE_ADJUSTMENT
- SOS_OBJECT_STORE_DESTROY_MUTEX
- SOS_PROCESS_AFFINITY_MUTEX
- SOS_SMALL_PAGE_ALLOC
- SOS_STACKSTORE_INIT_MUTEX
- SOS_SYNC_TASK_ENQUEUE_EVENT
- SOS_VIRTUALMEMORY_LOW
- SOSHOST_EVENT
- SOSHOST_INTERNAL
- SOSHOST_MUTEX
- SOSHOST_RWLOCK
- SOSHOST_SEMAPHORE
- SOSHOST_SLEEP
- SOSHOST_TRACELOCK
- SOSHOST_WAITFORDONE
- SQLCLR_APPDOMAIN
- SQLCLR.Assembly
- SQLCLR_DEADLOCK_DETECTION
- SQLCLR_QUANTUM_PUNISHMENT
- SQLSORT_NORMMUTEX
- SQLSORT_SORTMUTEX
- SQLTRACE_LOCK
- SQLTRACE_SHUTDOWN
- SQLTRACE_WAIT_ENTRIES
- SRVPROC_SHUTDOWN
- TIMEPRIV_TIMEPERIOD
- TRACE_EVTNOTIF
- TRACEWRITE
- TRANSACTION_MUTEX
- UTIL_PAGE_ALLOC
- VIA_ACCEPT
- VIEW_DEFINITION_MUTEX
- WAIT_FOR_RESULTS
- WAITFOR
- WAITFOR_TASKSHUTDOWN
- WAITSTAT_MUTEX
- WCC
- WORKTBL_DROP
- WRITE_COMPLETION
- XACT_OWN_TRANSACTION
- XACT_RECLAIM_SESSION
- XACTLOCKINFO
- XACTWORKSPACE_MUTEX
The Transaction Log wait type category includes all the waits that are associated with transaction log bottlenecks, including the following:

- LOGBUFFER
- LOGMGR
- WRITELOG
SQLdm offers numerous alerts to successfully monitor and warn you about your SQL Server instances. Proper configuration is key to making sure you receive alerts only on those metrics that actually fall outside the typical performance of your environment. SQLdm includes the following metric alerts for your use:

- Blocked Sessions
- Blocked Sessions Wait Time (Seconds)
- Client Computers
- CLR Enabled
- Cluster Active Node
- Cluster Failover
- Data File Autogrow
- Database Full
- Database Read and Write Error Occurred
- Database Status
- Deadlock
- Distribution Latency (Seconds)
- DTC Status
- Full Text Search Status
- Log File Autogrow
- Log Full
- Longest Running Version Store Transaction (Minutes)
- Maintenance Mode Enabled
- Mirror Commit Overhead
- Mirrored Server Role Change
- Mirroring Oldest Unsent Transactions
- Mirroring Session Preferred Role
- Mirroring Status
- Mirroring Unrestored Log
- Mirroring Unsent Log
- Mirroring Witness Connection
- Non-Distributed Transactions (Count)
- Oldest Open Transaction (Minutes)
- OLE Automation Disabled
- OS Disk Full (Percent)
- OS Disk Queue Length
- OS Memory Usage Alert
- OS Metrics Collection Status
- OS Paging
- OS Privileged Time (Percent)
- OS Processor Queue Length
- OS Processor Time
- OS User Time (Percent)
- Page Life Expectancy
- Procedure Cache Hit Ratio
- Query Monitor Events (Counts)
- Session CPU Time (Seconds)
- Session Tempdb Space Usage (MB)
- SQL Server Agent Error Log
- SQL Server Agent Job Completion
- SQL Server Agent Job Failure
- SQL Server Agent Long Running Job (Minutes)
- SQL Server Agent Long Running Job (Percent)
- SQL Server Agent Status
- SQL Server Agent XPs Disabled
- SQL Server CPU Usage
- SQL Server Database Space Used (Percent)
- SQL Server Error Log
- SQL Server Log Space Used (Percent)
- SQL Server Memory Usage
- SQL Server Response Time
- SQL Server Status
- SQLdm Service Stats
- Table Fragmentation (Percent)
- Tempdb Contention (ms)
- Unsubscribed Transaction Time (Seconds)
- Unsubscribed Transactions (Count)
- User Connections (Percent)
- Version Store Generation Ratio
- Version Store Size (MB)
- WMI Service Unavailable
This alert tracks the number of sessions on the SQL Server instance being blocked by other sessions holding requested locks.

By default, a blocked sessions waiting for a lock will wait forever (which is not good from an end-user point of view). This default behavior can be changed by adding \texttt{SET LOCKTIMEOUT 60000} to the top of the first T-SQL batch after a connection is made.

**What can I do to fix this problem?**

If the number of blocked sessions alerts exceeds 1 on an average day, your site may be experiencing one of the following issues:

- Some T-SQL batches may be submitted that contain a \texttt{BEGIN TRANSACTION} with no corresponding \texttt{COMMIT TRANSACTION}. You must correct the T-SQL.
- Some T-SQL batches may be submitted that contain a \texttt{BEGIN TRANSACTION} but where the \texttt{COMMIT TRANSACTION} statement is in a following T-SQL batch that is only executed once the end-user has "confirmed" the transaction. You must correct the T-SQL.
- The site has some very long-running transactions that are not running at off-peak times.

When your transactions are simply running longer than they should (and hence are clashing), consider drastically reducing the time that a lock is held. This can be done by:

- Doing as much work as possible (for example, SELECT statements that may be needed) before the transaction performs its first update, delete, or insert.
- Group all UPDATES, DELETEs, and INSERTS as closely as possible together within a transaction with as few selects as possible separating them.
- Commit the transaction as soon as possible after the last DML statement.
- Once the transaction has begun do not have any stops for user input. Gather all user inputs before the transaction starts.
- Avoid the use of server side cursors during a transaction as they slow execution considerably.
- If a stored procedure and/or trigger are invoked inside a transaction minimize or eliminate the number of SQL re-compilations made to that object. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- Increase the speed of transaction throughput such that it becomes less likely that one transaction will wait for the preceding one. This can be achieved by:
  - Adding more disks to your RAID solution.
  - Replacing your disks with faster disks.
  - Switching your RAID array from a RAID 5 to a RAID 10 solution as each write IO results in two writes using RAID 10 vs. four for RAID 5 (100% more efficient with writes).
  - So long as the RAID controller has some form of battery backup, then switch its cache mode from Write-through to Write-back as this increases the system's ability to handle write IOs by an order of magnitude.
  - Adding more cache memory to the RAID controller.
  - Adding more RAM to the server.
  - Adding another CPU to a SMP computer.
  - Upgrading the CPU, memory and motherboard with faster models.
  - Minimize the number of Context Switches by turning on Use NT Fibers in SQL Server.
  - Switch the Boost SQL Server Priority on.
This alert tracks the time in seconds that a SQL Server session has been blocking other sessions.

By default, blocked sessions waiting for a lock will wait forever (which is not good from an end-user point of view). This default behavior can be changed by adding `SET LOCKTIMEOUT 60000` to the top of the first T-SQL batch after a connection is made.

**What can I do to reduce blocked sessions wait time?**

If the number of blocked sessions alerts exceeds 1 on an average day, then your site may be experiencing one of the following issues:

- Some T-SQL batches may be submitted that contain a `BEGIN TRANSACTION` with no corresponding `COMMIT TRANSACTION`. You must correct the T-SQL.
- Some T-SQL batches may be submitted that contain a `BEGIN TRANSACTION` but where the `COMMIT TRANSACTION` statement is in a following T-SQL batch that is only executed once the end-user has "confirmed" the transaction. You must correct the T-SQL.
- The site has some very long-running transactions that are not running at off-peak times.

Where your transactions are simply running longer than they should (and hence are clashing), consider drastically reducing the time that a lock is held. This can be done by:

- Perform as much work as possible (for example, SELECT statements that may be needed) before the transaction performs its first update, delete, or insert.
- Group all UPDATES, DELETES, and INSERTS as closely as possible together within a transaction with as few SELECTS as possible separating them.
- Commit the transaction as soon as possible after the last DML statement.
- Once the transaction has begun, do not have any stops for user input. Gather all user inputs before the transaction starts.
- Avoid the use of server side cursors during a transaction as they slow execution considerably.
- If a stored procedure and/or trigger is invoked inside a transaction, minimize or eliminate the number of SQL re-compilations made to that object. See the SQL RE-compilations counter for ways to dramatically reduce recompiles.
- Increase the speed of transaction throughput such that it becomes less likely that one transaction will wait for the preceding one. This can be achieved by:
  - Adding more disks to your RAID solution.
  - Replacing your disks with faster disks.
  - Switching your RAID array from a RAID 5 to a RAID 10 solution as each write IO results in two writes using RAID 10 vs. four for RAID 5 (100% more efficient with writes).
  - So long as the RAID controller has some form of battery backup, then switch its cache mode from Write-through to Write-back as this increases the system’s ability to handle write IOs by an order of magnitude.
  - Adding more cache memory to the RAID controller.
  - Adding more RAM to the server.
  - Adding another CPU to a SMP computer.
  - Upgrading the CPU, memory and motherboard with faster models.
  - Minimize the number of Context Switches by turning on Use NT Fibers in SQL Server.
  - Switch the Boost SQL Server Priority on.
This alert tracks the number of unique client computers connected to the SQL Server instance. This value is measured by getting the count of processes from distinct computers in `sysprocesses` table.
This alert indicates that the Common Language Runtime (CLR) configuration option has been enabled for the SQL Server instance. Managed code can be run in the .NET Framework. Use the Server Configuration view to reconfigure this option on SQL Server 2005 instances.
This alert indicates that the active node of the cluster is a non-preferred node.

**What would cause the active cluster node to change?**

The active cluster node changes when a failover event occurs. If the original preferred node is offline for an extended period of time, the preferred node setting can be changed by right-clicking the instance in the Servers tree and selecting Properties > Cluster Settings. By default, the preferred node is first node encountered when monitoring was started.
Cluster Failover Alert

This alert indicates that there has been a change in the active node. This alert will remain active for the amount of time specified in the alert's advanced settings.

To edit these settings, right-click the server in the Servers tree and select Configure Alerts. Click the Advanced button on the Cluster Failover Alert and edit the time.

**How can I prevent cluster failover alerts when I have a scheduled failover?**

You can temporarily put the server into maintenance mode by right-clicking the server in the Servers tree and selecting Maintenance mode. When the scheduled failover is over, right-click the instance in the Servers tree and select Disable Maintenance Mode.
This alert indicates that a data file autogrow occurred on the specified database. Your Query Monitor must be running to trigger this alert.

To learn more about how autogrow works and related best-practices, see Microsoft Support Article 315512, "Considerations for the 'autogrow' and 'autoshrink' settings in SQL Server".
This alert indicates the percentage of the allowable disk space for the database currently used by the sum of data, text, and indexes. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings and available disk space.

If this value exceeds 80%, consider replacing disks with bigger disks.
This alert indicates that an I/O error occurred while attempting to read from or write to a database file. If this error persists consider the following solutions:

- Replace the disks.
- File may be corrupted, see if you can restore from a backup.
This alert indicates that the Database Status metric has reached an alert status. This status can be caused when the database:

- Is inaccessible
- Has an issue with loading, crashed while loading, has not recovered, in recovery, suspect, or in emergency mode
- Contains databases sizes in the error threshold
- Contains logs with sizes in the error threshold
- Contains tables with reorganization needs in the error threshold
- Is transitioning, loading, being check pointed, set as DBO use only, or in single-user mode

What can I do to fix this problem?

Navigate to the Details view on the Database tab of the SQLdm Console. Look for databases that have a critical or warning status and review what the issue is (the problem column will be highlighted either yellow or red). You may need to rebuild an index, increase allocated file size, or turn on the autogrowth feature for your database or logs.
This alert indicates that a deadlock has occurred on the monitored server. For the Deadlock Alert to display, Query Monitor must be running with Deadlock Events enabled.

Why do deadlocks occur?

- Deadlocks occur when a process or several processes attempt to access a resource that the other holds a lock on. Neither task can complete because both processes are attempting to lock the same resource.
- SQL Server chooses one or more deadlock to terminate so that processing can continue.

How do I diagnose deadlock issues?

- When a Deadlock Alert occurs, check the Alerts view in SQLdm and select a deadlock alert. The SPID, Hostname, Login Name, Client Application, and Database are listed.
- For more information, select Show Deadlock Details. This will help with diagnosing the problem application and includes the option to export deadlock data to view in SQL Server Management Studio or SQL Server Profiler.
This alert tracks the amount of time, in seconds, that a replication transaction has been at the Publisher waiting to be received by the Distributor, and has exceeded the warning threshold.

**What causes high distribution latency?**

The most common cause of high distribution latency is when a transaction has a large amount of commands. Once the transaction has completed, distribution latency will decrease.

The Replication view on the Services tab displays a graph with the non-distributed count and the non-distributed time. If these are trending upwards, there may be a problem that needs to be resolved.

**What can I do to reduce distribution latency?**

It is possible to reduce distribution latency by:

- Reducing traffic on the publisher
- Limiting non-replication related work the publisher is doing
- Increasing the bandwidth available to the publisher on the network
This alert indicates the status of the Distributed Transaction Coordinator service.
This alert indicates the status of the Full-Text Search service.
This alert indicates that a log file autogrow occurred on the specified database. Your Query Monitor must be running to trigger this alert.

To learn more about how autogrow works and related best-practices, see Microsoft Support Article 315512, "Considerations for the 'autogrow' and 'autoshrink' settings in SQL Server".
This alert indicates the percentage of the allowable disk space for the database currently used by the transaction log. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings and available disk space.

If this value exceeds 80%, consider replacing disks with bigger disks.
This alert indicates the time in minutes that the longest version store transaction has been open and preventing version store cleanup. This alert occurs only on instances running SQL Server 2005 or above.
This alert indicates that the SQLdm Maintenance Mode is enabled. Maintenance Mode allows you to temporarily stop alert generation and the collection of performance metrics for the time period that this SQL Server instance is offline.
Mirror Commit Overhead Alert

This alert indicates that the commit overhead (in MS) of the mirrored database has reached or exceeded an alert threshold.

Specifies the number of milliseconds of average delay per transaction that are tolerated before a warning is generated on the principal server. This delay is the amount of overhead incurred while the principal server instance waits for the mirror server instance to write the transaction's log record into the redo queue. This value is relevant only in high-safety mode.

For more information about mirror commit overhead, see SQL Server Books Online.
This alert indicates when the role of a mirrored database changes.

When you configure this alert, you can set the time at which the alert will reset. Reset this alert when the role of the mirrored database has no impact.
This alert tracks the age of unsent transactions queued on the processor has reached an alert threshold.

This alert specifies the number of minutes worth of transactions that can accumulate in the send queue before a warning is generated on the principal server instance. This warning helps measure the potential for data loss in terms of time and is especially relevant for high-performance mode. However, the warning is also relevant for high-safety mode when mirroring is paused or suspended because the partners become disconnected.

For more information about oldest unsent transactions, see SQL Server Books Online.
This alert indicates that a database mirror partner took on a non-preferred role. This means the roles in the mirroring relationship do not conform to the roles that have been specified as Normal. This alert does not reset over time. This alert remains until the roles are restored or you change the Normal state.

You can restore the roles or modify the Normal definition in the Database Mirroring view.
This alert indicates that the mirrored database is in a Critical or Warning state.

You can change the status for the mirrored databases in the Mirroring table of the Databases tab.
This alert indicates that the amount of unrestored logs (in KB) queued on the mirrored database has reached or exceeded an alert threshold.

This warning helps measure failover time. Failover time consists mainly of the time that the mirror server requires to roll forward any log remaining in its redo queue, plus a short additional time.

For more information about unrestored logs, see SQL Server Books Online.
This alert indicates that the amount of unsent logs (in KB) queued on the mirrored database has reached or exceeded an alert threshold.

This alert specifies how many kilobytes (KB) of unsent logs generate a warning on the principal server instance. This warning helps measure the potential for data loss in terms of KB and is especially relevant for high-performance mode. However, the warning is also relevant for high-safety mode when mirroring is paused or suspended because the partners become disconnected.

For more information about unsent logs, see SQL Server Books Online.
This alert indicates that the connection to the mirror has been lost.
Non-Distributed Transactions (Count) Alert

This alert tracks the number of replication transactions published by the Publisher but not received by the Distributor.
This alert tracks the time (in minutes) that the longest running transaction in the database has used since issuing the BEGIN TRANSACTION statement.
This alert indicates that the OLE Automation Procedures configuration option has been disabled (2005) or the procedures have been dropped (2000) for the SQL Server instance. SQL diagnostic manager will not collect data such as OS metrics and service details. Use the Server Configuration view to reconfigure this option on SQL Server 2005 instances.
OS Disk Full (Percent) Alert

This alert indicates the percentage of space used on a logical disk on the SQL Server computer. This will alert separately for each disk on a server.
This alert indicates the average number of both read and write requests that were queued for all the disks on the SQL Server computer.

**What can I do to reduce the queue length?**

If this queue often remains above 6 per physical disk for more than 10 seconds at a time then your disk subsystem is overloaded. In this case consider the following:

- Check OS Paging to make sure that paging from/to the swap file is not causing these IOs.
- Replacing disks with faster disks.
- Add more disks to you RAID array.
- Switch your RAID array from RAID 5 to RAID 10 solution as each write IO results in two writes using RAID 10 vs. 4 for RAID 5.
- Move other applications to another computer.
- So long as the RAID controller has some form of battery backup then switch its cache mode from Write-through to Write-back as this increases the system ability to handle write IOs by an order of magnitude.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.
This alert tracks the percentage of physical and virtual memory has reached an alert status. Consistently high Total Memory Usage could indicate either:

- A program in use could have a memory leak.
- The amount of total memory needs to be increased.

The Memory view on the Resources tab allows you to track the Total Memory usage along with other key memory statistics on the computer hosting your SQL Server instance.

**What can I do to fix this problem?**

A high percentage of Total Memory Usage indicates a memory bottleneck. You should open the Memory view on the Resources tab to view how memory is being utilized on your SQL Server instance. You may need to allocate more memory to SQL Server to increase your SQL Server performance, or limit the usage of other applications on the computer hosting your SQL Server instance.
This alert indicates the status of SQL diagnostic manager OS Metrics Collection.
This alert indicates that a memory threshold has been met or exceeded, due to one of the following causes:

- The OS Memory Paging is high
- The OS Memory Usage is high
- The SQL Server Memory Usage is high

Consistently high memory usage could cause system performance problems because of heavy disk usage and CPU load.

What can I do to fix this problem?

A high number of memory paging indicates a memory bottleneck. You should open the Memory view on the Resources tab to see how memory is being utilized on your SQL Server instance. You may need to allocate more memory to SQL Server to increase your SQL Server performance.
This alert tracks the percentage of CPU time spent by all processes executing in privileged mode on the SQL Server computer.
This alert indicates that the number of threads waiting on the processor has reached an alert threshold. A consistently high processor queue length can indicate processor congestion. If your system has multiple processors, your alert thresholds may need to be increased.

**What can I do to fix this problem?**

If the Processor Queue Length is often high, consider making one of the following changes:

- Upgrade to faster CPU(s).
- Reduce the number of SQL re-compilations as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled, and most of these reasons can be removed by careful coding. See the SQL re-compilations counter for ways to dramatically reduce recompiles.
- Stop unnecessary programs from running (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC).
OS Processor Time Alert

This alert tracks the percentage of CPU time used by all processes on the computer that hosts your SQL Server instance. WMI object WIN32_PERFRAWDATA_PERFOS_PROCESSOR property PERCENTPROCESSORTIME is measured to get percent processor time.

What can I do to reduce this value?

If this value regularly exceeds 75%, action needs to be taken, such as:

- Reduce the number of SQL re-compilations since they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and most of these reasons can be removed by careful coding. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- Ensure that all T-SQL statements (whether in a stored procedure, trigger, or ad hoc statement) that reference objects fully qualify the object referenced. As an example: `SELECT * FROM Northwind.dbo.Employees` is a fully qualified object reference whereas `SELECT * FROM Employees` is a poorly qualified object. The execution plans of fully qualified objects can be reused as is, whereas plans where the object are not fully qualified either cannot be reused at all or if they are then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. In both cases a significant amount of CPU time is consumed.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.

Why does this metric have no value?

SQLdm has not been able to collect the corresponding counter due to one of the following reasons:

- SQLdm needs to refresh. It usually requires two refreshes for the CPU Usage metric to begin populating, which occurs within a few seconds.
- OLE Automation is not working properly. OLE Automation must be enabled for SQLdm to collect hardware performance metrics. To enable OLE Automation, click the warning message below the ribbon menu on the SQLdm Console.
- Lightweight Pooling is enabled on the SQL Server instance.
- No alerts are enabled. SQLdm displays an alert status when at least one of the three alerts is enabled.
- This counter does not exist on this SQL Server instance.
This alert tracks the percentage of CPU time spent by all processes executing in user mode on the SQL Server computer.
This alert indicates that the number of seconds a page stays in the buffer pool without references has fallen below the warning level. A decline in page life expectancy can indicate an increase in the physical I/O requirements for a user database. The rate decrease could likely indicate that the memory taken away from the buffer pool is forcing database pages to exit the buffer pool prematurely.

How can I use the Page Life Expectancy metric to diagnose problems with my SQL Server instance?

Microsoft recommends a minimum target for page life expectancy of 300 seconds. This means that any given page in memory will be kept in the buffer for 5 minutes before the buffer flushes the page to disk. A page life expectancy value of less than 300 seconds is indicative of either a memory problem or inefficient query plans.
This alert tracks the ratio of procedure cache hits to procedure cache lookups. This indicates when an execution plan is being reused (from memory) as opposed to being compiled (from disk).

What can I do to fix problems with my Procedure Cache Hit Ratio?

If values below 80% are regularly encountered (once SQL Server has been up and actively running for at least 30 minutes), then consider one of the following remedies:

- Make sure that all T-SQL statements (whether in a stored procedure, trigger or ad hoc statement) that reference objects fully qualify the object referenced. As an example: `SELECT * FROM NORTHWIND.DBO.EMPLOYEES` is a fully qualified object reference whereas `SELECT * FROM Employees` is a poorly qualified object. The execution plans of fully qualified objects can be reused as is, whereas plans where the object are not fully qualified either cannot be reused at all or if they are then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache.

- Add more physical memory (RAM) to the computer.

- If your site makes use of extended stored procedures that are not called very often, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure has been called, it remains in memory until SQL Server is shut down even if it is never called again! This is tremendously wasteful in terms of available memory and a `DBCC DLLNAME (FREE)` will release that memory for use by both the procedure cache as well as the Data Pages which will have a significant positive effect on both the Procedure Cache Hit Ratio, as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server, execute `SP_HELPEXTENDEDPROC`.

- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).

- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.

- If the computer is running multiple instances of SQL Server, consider placing each instance on a separate physical computer.
Query Monitor Events (Counts) Alert

This alert tracks the number of queries captured by the Query Monitor on the SQL Server instance during the last scheduled refresh. A warning is generated if a query matches the criteria specified for a poorly performing query.
This alert tracks the CPU Time (in seconds) used by the SQL Server session since the last scheduled refresh.
This alert indicates the amount of tempdb space being used by a running session. This alert occurs only on instances running SQL Server 2005 or above.
This alert tracks when specific text appears in the SQL Server Agent error log. You can customize which text will trigger the alert. You can also customize the severity of the alert based on the frequency of occurrence.

The default settings are:

- 2 - Warning
- 3 - Critical
This alert indicates the status of the Agent Job Completion metric. The value can either be Succeeded, Retry, Canceled, Failed, or Unknown.
This alert indicates that the scheduled SQL Server Agent job ended with a Failed status.
This alert indicates the maximum number of minutes a scheduled SQL Server Agent job should take to complete.
This alert indicates the percentage of time a scheduled SQL Server Agent job is taking over the average successful completion time for this job in the current job history.
This alert indicates the status of the SQL Server Agent Service.
This alert indicates that the Agent XPs configuration option has been disabled (2005) or the procedures have been dropped (2000) for the SQL Server instance. SQL diagnostic manager will not collect SQL Agent job details. Note that Agent XPs are disabled when SQL Server Agent is stopped, but they may be re-enabled through the Server Configuration view to allow agent job histories to be shown even when the agent is not running.
This alert indicates that the amount of CPU usage being utilized by SQL Server is unusually high. The percentage of SQL Server processor usage is listed under the control. A high SQL Server percentage could indicate a large number of active client sessions.

What can I do to fix this problem?

Consistently high SQL Server processor usage could indicate the need to:

- Reduce the number of SQL re-compilations as they are CPU intensive. There are many reasons that an object such as a stored procedure is recompiled and most of these reasons can be removed by careful coding.
- Ensure that all T-SQL statements (whether in a stored procedure, trigger or ad hoc statement) that reference objects fully qualify the object referenced. For example:

  ```sql
  SELECT * FROM Northwind.dbo Employees
  ```

  is a fully qualified object reference whereas

  ```sql
  SELECT * FROM Employees
  ```

  is a poorly qualified object. The execution plans of fully qualified objects can be reused as is, whereas plans where the object are not fully qualified either cannot be reused at all or if they are then they are subject to a highly restrictive `COMPILE` lock while SQL Server determines if all of the objects referenced in the T-SQL code have the same owners as the execution plan currently in cache. In both cases, a significant amount of CPU time is consumed.
SQL Server Database Space Used (Percent) Alert

This alert indicates that the percentage of allocated database space being used is high.

What can I do to fix this problem?

○ The amount of space being allocated to databases may need to be increased, disks may need to be added to the computer hosting the instance, or databases may need to be moved to another host computer.
○ If you have autogrowth turned off, this alert indicates that you either need to allocate more space for databases or that you may need to add more disk space.
○ If you have the autogrowth option turned on and you are seeing this alert quite often, you may need to adjust your autogrowth settings to a higher percentage to improve performance that can be degraded due to constant growth adjustments.
○ In certain cases, SQLdm may incorrectly report the file size. This occurs when certain columns in the `sysindexes` table in SQL Server 2000 become inaccurate and provide SQLdm with incorrect information regarding the number of rows and table size.
○ To correct this problem, use the following T-SQL syntax:

```sql
DBCC UPDATEUSAGE ('database_name') GO
```

This command can be server intensive so it is recommended that you execute it during non-peak hours.
SQL Server Error Log Alert

This alert tracks when specific text appears in the SQL Server log. You can customize which text will trigger the alert. You can also customize the severity of the alert based on the frequency of occurrence.

The default settings are:

- 11 - Warning
- 18 - Critical
SQL Server Log Space Used (Percent) Alert

The percentage of allocated disk space used by logs on the SQL Server instance has reached an alert threshold. Once the log space is completely filled, all updates to the event log will fail.

What can I do to fix this problem?

- You can allocate more disk space to the event log, turn on the auto-growth feature for your SQL Server log, or add additional disk space to the computer hosting your SQL Server instance.
- If you have autogrowth turned off for your logs, this alert may indicate the need to either increase the allocated space for logs and/or add disk space to your SQL Server environment.
- If you have the autogrowth option turned on and you are seeing this alert quite often, you may need to adjust your autogrowth settings to a higher percentage to improve performance that can be degraded due to constant growth adjustments.
- In certain cases, SQLdm may incorrectly report the file size. This occurs when certain columns in the 'sysindexes' table in SQL Server 2000 become inaccurate and provide SQLdm with incorrect information regarding the number of rows and table size.
- To correct this problem, use the following T-SQL syntax:

```sql
DBCC UPDATEUSAGE ('database_name') GO
```

This command can be server intensive so it is recommended that you do this outside of production work hours.
SQL Server Memory Usage Alert

This alert tracks the total server memory (Total Server Memory per `sysperfinfo`) as a percent of total physical memory from WMI.

**What can I do to reduce the percentage of memory used by SQL Server?**

If this value is regularly over 80%, SQL Server needs more memory or needs to use the memory it has more efficiently. Consider implementing one or more of the following solutions:

- If your site makes use of extended stored procedures that are not called very often, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure has been called, it remains in memory until SQL Server is shut down even if it is never called again! This is tremendously wasteful in terms of available memory and a `DBCC dllname (FREE)` will release that memory for use by both the procedure cache, as well as the Data Pages, which will have a significant positive effect on both the Procedure Cache Hit Ratio, as well as the Buffer Cache Hit Ratio. In order to view the extended stored procedures currently loaded in memory by SQL Server, execute `SP_HELP_EXTENDEDPROC`.

- As each SQL Server lock requires 96 bytes of memory, the granting of lock space is done at the expense of Data Pages and Procedure Cache Pages. It is therefore in the best interests of system performance and throughput that the number of locks is kept to a minimum and this is achieved by:
  - Wherever possible, on select statements use the `(NOLOCK)` optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` as this neither issues any shared locks on the data it reads nor honors any exclusive locks set by other transactions.
  - When updating all rows in a table with more than 50 rows, use the `TABLOCKX` table hint as this will prevent SQL Server from initially taking exclusive row locks and then once it has granted many of these, then escalating to an exclusive table lock anyway.
  - When deleting all rows in any table, do not use the `DELETE` statement but rather issue the `TRUNCATE TABLE` statement as far fewer locks and other system resources are consumed in the process.
  - Drastically reduce the time that a lock is held. This can be done by:
    - Doing as much work as possible (for example, SELECT statements that may be needed) before the transaction performs its first update/delete/insert.
    - Group all updates/deletes/inserts as closely as possible together within a transaction with as few selects as possible separating them.
    - Commit the transaction as soon as possible after the last DML statement.
    - Once the transaction has begun, do not have any stops for user input. Gather all user inputs before the transaction starts.
    - Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).

- Add more physical memory (RAM) to the computer.
- If the computer is running multiple instances of SQL Server then consider placing each instance on a separate physical computer.
- If the computer is running other memory-intensive applications, such as IIS or Exchange, then consider moving each instance to a separate physical computer.
- A SQL Server computer should be performing only SQL Server work so stop unnecessary programs from running, such as allowing the computer to act as either a PDC or BDC.
This alert tracks the time in milliseconds SQL diagnostic manager currently needs to send a simple SQL command to the SQL Server instance, have it processed, and receive the returned result set.

High response times generally occur when sessions are blocked or when either there is heavy network traffic or network problems or when either the computer the SQLdm Repository or the computer the SQLdm Console is hosted on is running slow due to various problems.

Typically, this value should average below 2000 ms for a well tuned network and server. Average values in excess of 2000 ms indicate either an excessively busy network segment or a stressed SQL Server.

**What can I do to fix this problem?**

A common cause to high response times is when sessions are being blocked. To fix this problem, go to the Sessions tab and select the Blocking view to see if any sessions are blocking. You can then select the session and click either Trace Session to see what queries the session is creating and to see what is causing the block, or Kill Session to stop it.
This alert indicates the status of the SQL Server service.
This alert indicates the status of the SQLdm Manager Service. The possible statuses can be Running, Stopped, or Unknown.
This alert tracks the percentage of pages in the table that are fragmented. It is the scan density from \texttt{dbcc showcontig} on a clustered index of a table. If the table does not have a clustered index, table fragmentation is not collected. If this value exceeds 80%, consider the following remedies:

- Drop and re-create the clustered index.
- Reorder the leaf-level pages of the index in a logical order.
- Rebuild the index online.
This alert indicates the latch wait time for tempdb allocation maps in milliseconds. The three tracked allocation page types are page free space (PFS), global allocation map (GAM), and shared global allocation map (SGAM). Latch contention of this sort is usually an indication that more tempdb data files need to be created. In some situation, using Trace Flag 1118 may also alleviate tempdb contention. This alert occurs only on instances running SQL Server 2005 or above.
Unsubscribed Transaction Time (Seconds) Alert

This alert tracks the time (in seconds) that a replication has been holding at the Distributor and waiting to be received by a Subscriber.
This alert tracks the number of replication transactions received by the Distributor but not received by a Subscriber.
This alert tracks the percentage of the user connections allowed by the SQL Server instance that are currently used.

If the percentage is high it indicates a load on the SQL Server, the higher this value the more SQL Server is having to reserve for connections.
**Version Store Generation Ratio Alert**

This alert indicates the percentage by which the tempdb version store generation rate exceeds the cleanup rate. This alert is disabled by default and occurs only on instances running SQL Server 2005 or above.

**TIP** You can raise this alert only when at least one minute elapses between alert refreshes as version store cleanup occurs once per minute.
This alert indicates the size of the tempdb version store in megabytes. This alert is disabled by default and occurs only on instances running SQL Server 2005 or above.
This alert tracks the WMI service is unavailable on the SQL Server computer. SQL diagnostic manager will not collect OS metrics.
Advanced solutions

SQLdm includes functionality aimed at the more advanced user who wants to manage these features without contacting Idera Support for assistance. Please do not attempt to perform these actions if you are unsure of what you are doing or why you would want to take such steps. These more advanced topics include how to:

- migrate or recover your installation
- deploy the Idera Newsfeed in a clustered environment
- deploy SQLdm in a clustered environment
- default alert IDs
- troubleshoot WMI connectivity issues
Establishing a migration or disaster recovery strategy for your SQLdm Repository and Services allows you to preserve historical data and existing configuration settings. You can also continue monitoring your SQL Server environment to meet your auditing requirements.

**TIP** SQLdm Application security is not maintained when your SQLdm installation is migrated to a new server.

This process helps you execute a migration or disaster recovery strategy that addresses one of the following situations:

- The computer hosting SQLdm requires maintenance, such as new hardware or a software upgrade (Microsoft Windows or SQL Server Service Pack)
- The computer hosting SQLdm becomes permanently unavailable
- You decommission and replace the computer hosting SQLdm

Refer to the following when migrating your SQLdm installation:

- Choosing to migrate or recover SQLdm
- Creating your migration or recovery plan
- Recovering your SQLdm installation
- Migrating your complete SQLdm installation
- Migrating the SQLdm Repository only
- Migrating the SQLdm Services only
The System Diagnostics window allows you to test the connections between the SQLdm Services and other SQLdm components. This test is designed to assist you when the location of SQLdm components change due to a migration.

You can perform the following tests using system diagnostics:

**Management Service Tests**
- GUI to management service connection
- GUI and SQLdm services using the same repository
- Management service repository connection

**Collection Service Tests**
- Management service to collection service connection
- Connection service configured to use correct management service
- Collection service can connect to management service
- Collection service heartbeat status
Choose to migrate or recover SQLdm

When choosing whether a migration or disaster recovery strategy is best for your environment, consider whether you will want to permanently move the SQLdm Repository or Services to another computer.

You can also choose to have the Repository and Console components located on computers that belong to different domains. This deployment option can be advantageous during disaster recovery, especially if the only available computer is in another domain. To ensure full communication between the Repository and the Consoles, verify that a two-way trust exists between the target domains.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replace the SQLdm Repository or Services computer</td>
<td>Migration</td>
</tr>
<tr>
<td>Repair the original Repository or Services computer</td>
<td>Recovery</td>
</tr>
</tbody>
</table>

In both cases, you will want to back up the Repository database and then later restore it.
Create your migration or recovery plan

Once you have decided whether a SQLdm Recovery or Migration is best for your environment, the next important step is to create a plan so that when the time comes, you are prepared and have SQL diagnostic manager up and running as quickly as possible.

Creating a Recovery Plan

A disaster recovery plan details the steps that must be taken for unexpected outages, to ensure you can continue monitoring SQL Server activity and performance metrics. This document addresses disaster recovery best-practices for establishing a new Repository.

When you implement SQLdm in your production SQL Server environment, consider preparing a disaster recovery plan to minimize audit data loss should the SQLdm Repository become unavailable. Use the procedures and guidelines covered in this document to implement or modify your disaster recovery plan.

Creating a Migration Plan

A migration plan details moving the SQLdm Repository and Services to another SQL Server instance, thereby replacing the original components. You can use a migration plan to respond to an immediate maintenance need. Use the procedures and guidelines in this document to implement or modify your migration plan.

Understanding the Repository Database

The SQLdm Repository consists of a SQL Server database named, by default, SQLdmRepository. This database contains the following information:

- Performance metrics and statistics collected from your monitored SQL Server instances
- Configuration settings such as the connection information for the SQLdm Services and alert notification rules
- Alert messages

By default, the Repository database uses the simple recovery model. When this setting is enabled, SQL Server does not maintain the transaction logs for the database. Likewise, any existing transaction logs are not included in backup data. If your corporate policies require transaction log backups, consider changing the recovery model to full so that transaction logs are maintained and archived.

Understanding SQLdm Services

SQLdm has two centralized services, the Management Service and the Collection Service. These two services reside on the same computer.

The Management Service performs the following primary functions:

- Provides real-time data to the SQLdm Console
- Receives historical data from the Collection Service for storage in the Repository
- Raises alerts and sends alert notifications

The Collection Service performs on-demand and scheduled collection from the monitored SQL Server instances.

Recovery and Migration Best Practices
Verify the Configuration of the Target SQL Server

When identifying the new SQL Server instance that will host the Repository and Services, ensure this instance meets or exceeds the product requirements as well as these specific requirements.

- The target instance can be dedicated to hosting the SQL diagnostic manager only
- The target instance is running the same version of SQL Server software that is currently running on the existing SQLdm computer

Back up the Repository Database

Use a tool such as Idera SQLsafe to perform a full backup of the Repository database. If you have changed the default recovery model to full, ensure your backup includes any transaction logs.

Identify how often to backup the Repository database

The frequency at which you backup the Repository database depends on the following factors:

- How often your alert settings change
- How often your SQL Server environment changes as you add new servers and databases or remove older servers and database
- How much performance data you collect in a given time period
- How much risk you are willing to incur

The backup frequency should reflect your maintenance needs and allow you to meet future monitoring requirements.

Schedule routine backups of the Repository database

After you identify the appropriate backup frequency for your monitoring needs, use a tool such as Idera SQLsafe to schedule routine backups of the Repository database. If you have changed the default recovery model to full, ensure your backup includes any transaction logs.

Review disaster recovery guidelines

Ensure your recovery strategy includes plans to reinstate the original computer once it is repaired. Consider the following guidelines:

- To minimize data loss, plan to back up the Repository database on the temporary Repository computer immediately before reinstating the original computer.
- Use the procedures in this Technical Solution to reinstate the Repository on the original computer and configure the Management Service.
- To verify all components were reinstated correctly, test your implementation.
- Uninstall the components you previously implemented on the temporary computer.

Recovery and Migration Best Practices

The backup frequency should reflect your maintenance needs and allow you to meet future monitoring requirements.

Schedule Routine Backups of the Repository Database

After you identify the appropriate backup frequency for your monitoring needs, use a tool such as Idera SQLsafe to schedule routine backups of the Repository database. If you have changed the default recovery model to full, ensure your backup includes any transaction logs.

Review Disaster Recovery Guidelines

Ensure your recovery strategy includes plans to reinstate the original computer once it is repaired. Consider the following guidelines:
○ To minimize data loss, plan to back up the Repository database on the temporary Repository computer immediately before reinstating the original computer.
○ Use the procedures in this Technical Solution to reinstate the Repository on the original computer and configure the Management Service.
○ To verify all components were reinstated correctly, test your implementation.
○ Uninstall the components you previously implemented on the temporary computer.
Recovering your SQLdm installation

In the event that your existing SQLdm Repository is no longer available, you can recover your SQLdm installation back to the last point a backup occurred.

Recovering the SQLdm Repository

To recover lost or damaged data, restore the SQLdm Repository database using the following guidelines:

- Perform a full restore
- Schedule the restore during off-hours, or times when you expect the least collection activity

To restore the Repository database:

1. Use SQL Server Enterprise Manager or Management Studio to close any open connections to the Repository database.
2. Use SQL Server Enterprise Manager or Management Studio to take the Repository database offline. If you cannot take the Repository database offline, stop the Collection Service.
3. Use a tool such as Idera SQLsafe to restore the Repository database using the appropriate backup archive file.
4. On the computer hosting the SQLdm services, restart the SQLdm Collection and Management services.
5. On all existing SQLdm consoles, reconnect to the SQLdm Repository by selecting File > Connect to SQLdm Repository.

Recovering the SQLdm Services

To install the SQLdm services:

1. Log on with an administrator account to the computer on which you want to install the SQLdm services.
2. Run Setup.exe in the root of the installation kit.
3. Click INSTALL on the Welcome window of the setup program.
4. Click Install SQL diagnostic manager on the Install window.
5. Read the Welcome window, and then click Next.
6. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
7. Accept the default folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
8. Select the Custom Install setup type, and then click Next.
9. Select SQLdm Services (also select the Desktop Client if you want to run it on the same computer), and then click Next.
10. Specify which SQL Server instance is hosting the Repository database, verify that the displayed database name is correct, and then click Next.
11. Specify the account credentials that the SQLdm services should use to connect to the SQLdm Repository and the monitored SQL Server instances, and then click Next. Ensure this account belongs to the sysadmin role on the SQLdm Repository database as well as each instance you plan to monitor.
12. Click Install to begin your installation.
13. Click Finish to exit the setup program.
The procedures in this section detail the migration steps for moving both the SQLdm Repository and the Services to a new computer. Follow each section in the order presented to ensure a successful migration.

**TIP**
SQLdm Application security is not maintained when your SQLdm installation is migrated to a new server. For more information, see Use application security to manage SQLdm.

### Get a New SQLdm License Key

Because you are migrating the Repository database to a different SQL Server instance, SQLdm will require a new license key. You can easily request a new license key by sending an email to: licensing@idera.com.

To complete the request, you will need the name of the SQL Server instance where you plan to install the SQLdm Repository database.

### Restore the SQLdm Repository on the Target Computer

For best results, use the following guidelines migrate your SQLdm Repository to the target computer:

- Perform a full restore
- Schedule the restore during off-hours, or times when you expect the least collection activity

To restore the Repository database:

1. Use SQL Server Enterprise Manager or Management Studio to close any open connections to the Repository database.
2. Use SQL Server Enterprise Manager or Management Studio to take the Repository database offline. If you cannot take the Repository database offline, stop the Collection Service.
3. Use a tool such as Idera SQLsafe to restore the Repository database on the target server using the appropriate backup archive file.

### Disable or Uninstall the SQLdm Services on the Current Computer

SQLdm Services must be uninstalled to avoid conflict with the restored SQLdm Repository. For best results, use the following procedure:

**If you do not want to keep the SQLdm console on the current computer, perform the following steps:**

1. Select Add > Remove Programs on your current computer.
2. Uninstall SQL diagnostic manager completely.

**If you want to keep the SQLdm console on the current computer, perform the following steps:**

1. Select Start > Administrative Tools > Services on the current computer.
2. Scroll down and right-click SQLdm Collection Service in the list, and select Properties.
3. In the Startup Type drop-down list, select Disabled.
4. Click OK.
5. Scroll down and right click SQLdm Management Services in the list, and select Properties.
6. In the Startup Type drop-down list, select Disabled.
7. Click OK.

### Perform a Full SQL diagnostic manager Install on the New Computer

A full install of SQL diagnostic manager will install the new SQLdm Services, link them to the restored SQLdm Repository, and
install the SQLdm Console. For best results, use the following procedure for installing SQLdm:

1. Log on with an administrator account to the computer on which you want to install the SQLdm services.
2. Run Setup.exe in the root of the installation kit.
3. Click INSTALL on the Welcome window of the setup program.
4. Click Install SQL diagnostic manager on the Install window.
5. Read the Welcome window, and then click Next.
6. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
7. Accept the default folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
8. Select your setup type and then click Next.
9. Browse and select the SQL Server instance your SQLdm Repository was restored to and click Next.
10. Enter your service account credentials and click Next.
11. Click Install to begin installing SQLdm with the options you have selected.

Verify that the Migration was Successful

The SQLdm Console includes a system diagnostics test that allows you to test the connections of all the SQLdm components.

1. Open the SQLdm console.
2. Enter your new SQLdm license key.
3. Add the SQL Server instances you want to monitor.
4. Open the System Diagnostics utility by selecting Help > System Diagnostics.
5. Click Test.
6. Verify that all the tests are successful.
The procedures in this section detail the migration steps for moving the SQLdm Repository to a new computer while leaving the SQLdm Services. Follow each section in the order presented to ensure a successful migration.

### Get a New SQLdm License Key
Because you are migrating the Repository database to a different SQL Server instance, SQLdm will require a new license key. You can easily request a new license key by sending an email to: licensing@idera.com.

To complete the request, you will need the name of the SQL Server instance where you plan to install the SQLdm Repository database.

### Restore the SQLdm Repository on the Target Computer
To recover lost or damaged data, restore the Repository database. For best results, use the following guidelines:

- **Perform a full restore**
- **Schedule the restore during off-hours, or times when you expect the least collection activity**

**To restore the Repository database:**
1. Use SQL Server Enterprise Manager or Management Studio to close any open connections to the Repository database.
2. Use SQL Server Enterprise Manager or Management Studio to take the Repository database offline. If you cannot take the Repository database offline, stop the Collection Service.
3. Use a tool such as Idera SQLsafe to restore the Repository database using the appropriate backup archive file.

### Configure the Management Service
If you migrated the Repository to a different computer, you will need to configure the Management Service so that it can communicate with the new Repository.

**To configure the SQLdm Management Service:**
1. Start the Management Service Configuration wizard by selecting Start > All Programs > Idera > SQL diagnostic manager > Tools > Management Service Configuration Wizard on the computer that hosts the SQLdm Services.
2. Read the Welcome window, and then click Next.
3. Verify that the following settings are correct, and then click Next.
   - The name of the SQL Server instance hosting the Repository database
   - The name of the Repository database
   - The type of authentication the Management Service should use to connect to the Repository database
4. Click Test to validate your settings.
5. Click Finish to exit the wizard.

### Verify that the Migration was Successful
The SQLdm Console includes a system diagnostics test that allows you to test the connections of all the SQLdm components.

1. Open the SQLdm console.
2. Enter your new SQLdm license key.
3. Add the SQL Server instances you want to monitor.
4. Open the System Diagnostics utility by selecting Help > System Diagnostics.
5. Click Test.
6. Verify that all the tests are successful.
The procedures in this section detail the migration steps for moving the SQLdm Services to a new computer while leaving the SQLdm Repository. Follow each section in the order presented to ensure a successful migration.

Uninstall SQLdm on the Original Computer

Uninstalling SQLdm on the original computer via Add or Remove Programs, removes all the SQLdm components from the original computer except for the SQLdm Repository. For best results, use the following procedure for uninstalling SQLdm on the original computer:

1. Select Start > Control Panel > Add or Remove Programs.
2. Scroll down to SQL diagnostic manager and select Remove.
3. Click Yes to the prompt asking if you want to completely remove SQL diagnostic manager.

Perform a Full SQL diagnostic manager Install on the New Computer

A full install of SQL diagnostic manager will install the new SQLdm Services, link them to the restored SQLdm Repository, and install the SQLdm Console. For best results, use the following procedure for installing SQLdm:

1. Log on with an administrator account to the computer on which you want to install the SQLdm services.
2. Run Setup.exe in the root of the installation kit.
3. Click INSTALL on the Welcome window of the setup program.
4. Click Install SQL diagnostic manager on the Install window.
5. Read the Welcome window, and then click Next.
6. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
7. Accept the default folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
8. Select your setup type and then click Next.
9. Browse and select the SQL Server instance on the original computer that contains your SQLdm Repository, and click Next.
10. Enter your service account credentials and click Next.
11. Click Install to begin installing SQLdm with the options you have selected.

Verify that the Migration was Successful

The SQLdm Console includes a system diagnostics test that allows you to test the connections of all the SQLdm components.

1. Open the SQLdm console.
2. Enter your new SQLdm license key.
3. Add the SQL Server instances you want to monitor.
4. Open the System Diagnostics utility by selecting Help > System Diagnostics.
5. Click Test.
6. Verify that all the tests are successful.
Deploy Idera Newsfeed in a clustered environment

You can deploy the SQLdm Mobile & Newsfeed Service to a clustered environment to ensure continuous remote alerting. See the following topics for installation and configuration instructions for Windows Server 2003 and Windows Server 2008 environments.

- Installing in a Windows Server 2003 Clustered Environment
- Installing in a Windows Server 2008 Clustered Environment
Use the following instructions to deploy the Idera Newsfeed Platform in a Windows Server 2003 clustered environment.

**Things to consider before you install the Idera Newsfeed Platform in a clustered environment**

- You will need to assign the Idera Newsfeed Platform to a new or existing cluster resource group. You may want to create a separate cluster resource group to host the SQLdm Mobile & Newsfeed Service.

  - Creating a new cluster resource group for this deployment allows the service to fail without causing other resources in the cluster to failover. However, this configuration also requires a new dedicated IP, a new Net-BIOS name, and a new dedicated drive.

  - Adding the SQLdm Mobile & Newsfeed Service to your existing SQLdm Services resource cluster group ensures all 4 services failover together, and does not require additional resources.

- The SQLdm Mobile & Newsfeed Service will run as a generic service on the assigned cluster resource group.

- You may need to provide a Network Name and Network IP during the cluster configuration.

- When you install the Idera News Platform, you will need to provide credentials for the service account and the name of a SQL Server instance that can host the SQLdm Mobile Repository database. Ensure these components meet the requirements.

- Install or upgrade your clustered SQLdm Services to version 7.1.

**Install the Idera Newsfeed Platform on each cluster node**

To set up Idera Newsfeed Platform to work correctly when the primary cluster node fails, you need to install the platform on each cluster node.

**To install Idera Newsfeed Platform:**

1. Log on using an administrator account.
2. Run `Setup.exe` in the root of the SQLdm installation kit.
3. Click Install, and then click Install Idera Newsfeed.
4. On the Welcome window, click Next.
5. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
6. Select the Mobile & Newsfeed components, and then click Next.
7. Specify the appropriate credentials for the service account, and then click Next.
8. Select which SQL Server instance should host the Repository database, and then click Next.
9. Specify the account that the setup program should use to create the Repository database, test the connection, and then click Next.
10. Indicate which credentials the SQLdm Mobile & Newsfeed Service account should use to connect to the Repository database, and then click Next.
11. Specify which computer hosts your SQLdm Repository, and then click Next.
12. Click Install.
13. In Windows Services, stop the following services, and set their Startup type to Manual.
   - SQLdm Mobile & Newsfeed Service
   - SQLdm Management Service
   - SQLdm Collector Service
   - SQLdm Predictive Analytics
Create the cluster resource group (optional)

For more granular failover control, you can create a dedicated cluster resource group for the Idera Newsfeed Platform.

**To create a cluster resource group for the Idera Newsfeed Platform:**

1. In the Cluster Administrator tool, create the cluster resource group:
   a. Select File > New > Group, and then specify a name and description.
   b. Add the nodes that belong to this cluster, and then click Finish.

2. Add a disk to the cluster resource group:
   a. Right-click the group you just created, select New > Resource, and then specify a name and description.
   b. For the Resource Type, select Physical Disk.
   c. Select the Idera Newsfeed Platform group.
   d. Click Next until you get to the last window and click Finish.

3. Add an IP address for the cluster resource group:
   a. Right-click the group, select New > Resource, and then specify a name and description.
   b. Select the Resource type IP Address.
   c. Type the IP address that you acquired for your new cluster resource group.
   d. Click Finish.

4. Add the Network Name resource to the cluster resource group:
   a. Right-click the group, select New > Resource, and then specify a name and description.
   b. Select the resource type Network Name.
   c. Add the IP address as a dependency for this release.
   d. Specify a name, and then click Finish.

Add the SQLdm Mobile & Newsfeed Service to the appropriate cluster resource group

On each cluster node, modify the cluster resource group settings to allow the Idera Newsfeed to recognize the cluster nodes.

**To add the SQLdm Mobile & Newsfeed Service to your resource cluster group:**

1. In the Cluster Administrator tool, navigate to the cluster resource group to which you want to add the Idera Newsfeed Service as a resource. You can use the new cluster resource group you previously created, or the cluster resource group you configured for your SQLdm Services.

2. Select the disk you created for this cluster resource group, and then create a directory on that disk. The SQLdm Mobile & Newsfeed Service will use this folder to store data.

3. Create an IP Address resource for the SQLdm Mobile & Newsfeed Service. If a Network Name is already specified for this cluster resource group, skip this step.
   ○ Create a Network Name resource (it will map to the IP Address)
   ○ Add the IP Address as a dependency

4. Create a Generic Service resource for the SQLdm Mobile & Newsfeed Service.
   ○ Add the Disk for the shared data as a dependency
   ○ Add the Network Name as a dependency
   ○ Specify SQLdmMobileandNewsService$Default as the name of the service to start
5. After the service has been successfully added, update its properties.
   a. On the Parameters tab, check the Use network name for computer name option.
   b. On the Registry Replication tab, add the SOFTWARE\Idera\Idera Newsfeed & Mobile\DataPath key.
   c. Click OK.
6. In Windows Services, bring the following services online:
   - SQLdm Mobile & Newsfeed Service
   - SQLdm Management Service
   - SQLdm Collector Service
   - SQLdm Predictive Analytics

Test your cluster configuration using the SQLdm Console
Ensure the SQLdm services can communicate with each other and the Repository databases.
1. On the active node of the cluster, use the Management Service Configuration wizard to re-register the services and test your configuration. You can start this tool from the Programs menu.
2. Verify that you are able to create a Newsfeed account and monitor your virtual SQL Server instances using the News Feed interface in the SQLdm Console.
Deploy the Idera Newsfeed to a Windows 2008 cluster

Use the following instructions to deploy the Idera Newsfeed Platform in a Windows Server 2008 clustered environment.

Things to consider before you install the Idera Newsfeed Platform in a clustered environment

- You will need to assign the Idera Newsfeed Platform to a new or existing clustered service. You may want to create a separate clustered service to host the SQLdm Mobile & Newsfeed Service.
  - Creating a new clustered service for this deployment allows the service to fail without causing other resources in the cluster to failover. However, this configuration also requires a new dedicated IP, a new NetBIOS name, and a new dedicated drive.
  - Adding the SQLdm Mobile & Newsfeed Service to your existing SQLdm Services clustered service ensures all 4 services failover together, and does not require additional resources.
- The SQLdm Mobile & Newsfeed Service will run as a generic service on the assigned clustered service.
- You may need to provide a Network Name and Network IP during the cluster configuration.
- When you install the Idera News Platform, you will need to provide credentials for the service account and the name of a SQL Server instance that can host the SQLdm Mobile Repository database. Ensure these components meet the requirements.
- Install or upgrade your clustered SQLdm Services to version 7.1.

Install the Idera Newsfeed Platform on each cluster node

To set up Idera Newsfeed Platform to work correctly when the primary cluster node fails, you need to install the platform on each cluster node.

To install Idera Newsfeed Platform:

1. Log on using an administrator account.
2. Run Setup.exe in the root of the SQLdm installation kit.
3. Click Install, and then click Install Idera Newsfeed.
4. On the Welcome window, click Next.
5. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
6. Select the Mobile & Newsfeed components, and then click Next.
7. Specify the appropriate credentials for the service account, and then click Next.
8. Select which SQL Server instance should host the Repository database, and then click Next.
9. Specify the account that the setup program should use to create the Repository database, test the connection, and then click Next.
10. Indicate which credentials the SQLdm Mobile & Newsfeed Service account should use to connect to the Repository database, and then click Next.
11. Specify which computer hosts your SQLdm Repository, and then click Next.
12. Click Install.
13. In Windows Services, stop the following services, and set their Startup type to Manual.
   - SQLdm Mobile & Newsfeed Service
   - SQLdm Management Service
   - SQLdm Collector Service
   - SQLdm Predictive Analytics
Create the clustered service (optional)

For more granular failover control, you can create a dedicated clustered service for the Idera Newsfeed Platform.

To configure the clustered service:

1. Start the Registry Editor and navigate to the `HKLM\SOFTWARE\Idera\Idera Newsfeed & Mobile` key. For this key, add a String Value named `DataPath` and set it to directory path for the new folder.
2. In the Failover Cluster Management tool, right-click the Services and Applications node, and then click Configure a Service or Application.
   a. Provide a name for the new clustered service.
   b. Select Generic Service.
   c. Select the SQLdm Mobile & Newsfeed Service.
   d. Enter the network name and IP address for the services.
   e. Select the clustered storage that will host the data folder path.
   f. Skip the Registry Replication.
   g. Verify your selections, and then click Next to complete the wizard.
3. Under Other Resources in the Services and Applications view, right-click on your new clustered service, and then take the SQLdm Mobile & Newsfeed Service offline.

Add the SQLdm Mobile & Newsfeed Service to the appropriate clustered service

On each cluster node, modify the clustered service settings to allow the Idera Newsfeed Platform to recognize the cluster nodes.

1. Select a disk that is configured for the target clustered service, and create a folder on that disk to store data for the SQLdm Mobile & Newsfeed Service. You can choose your new clustered service or the existing clustered service already used for the SQLdm Services.
2. Expand Services and Applications, right-click the newly created service, and then click SQLdm Mobile & Newsfeed Service > Properties.
   a. On the General tab, select Use Network Name for the computer name.
   b. On the Dependencies tab, specify the Disk and Name used by the service.
   c. On the Registry Replication tab, add `HKLM\SOFTWARE\Idera\Idera Newsfeed & Mobile\DataPath`.
3. Right-click the new clustered service node, then select Add a Resource > Generic Service.
4. Select the SQLdm Mobile & Newsfeed Service from the list of services.
5. Click Next to finish the wizard.
6. In Windows Services, bring the following services online:
   a. SQLdm Mobile & Newsfeed Service
   b. SQLdm Management Service
   c. SQLdm Collector Service
   d. SQLdm Predictive Analytics

Test your cluster configuration using the SQLdm Console

Ensure the SQLdm services can communicate with each other and the Repository databases.

1. On the active node of the cluster, use the Management Service Configuration wizard to re-register the services and test your configuration. You can start this tool from the Programs menu.
2. Verify that you are able to create a Newsfeed account and monitor your virtual SQL Server instances using the News Feed interface in the SQLdm Console.
SQL diagnostic manager allows you to monitor your clustered SQL Server environment. See the following topics for installation and configuration instructions for Windows Server 2003 and Windows Server 2008 environments.

- Installing in a Windows Server 2003 Clustered Environment
- Installing in a Windows Server 2008 Clustered Environment
- Configure alerts for important cluster events
The following instructions guide you through the installation of SQLdm in a Windows Server 2003 based clustered environment.

Things to consider before you install SQLdm in a clustered environment

- You will need to create a separate cluster resource group to host SQLdm services. This allows the SQLdm services to fail without causing other resources in the group to failover.
- To create this cluster resource group you will need a dedicated IP, the NetBIOS name, and a dedicated disk for that resource group.
- SQLdm Collection, Management, and Predictive Analytics services will run as generic services on the newly created cluster resource group.
- You may need to provide the network Name and network IP as part of cluster configuration.

Install SQLdm Services on Cluster Nodes

To set up SQLdm services to work correctly when the primary cluster node they are hosted on fails, you need to install the services on each cluster node.

To install SQLdm services on cluster nodes:

1. Log on with an administrator account to the computer on which you want to install SQLdm.
2. Run Setup.exe in the root of the SQLdm installation kit on the first cluster node.
3. Click Install and then click Install SQL diagnostic manager.
4. Read the Welcome window, and then click Next.
5. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
6. Accept the default installation folder, or click Change to specify a different folder, and click Next.
7. Select the Typical setup type and click Next.
8. Select the SQL Server instance and enter a database name.
9. Enter the service account information and click Next.
10. Click Install.
11. In Windows Services, stop the SQLdm Management service and set the Startup type to Manual.
12. In Windows Services, stop the SQLdm Collection service and set the Startup type to Manual.
13. In Windows Services, stop the SQLdm Predictive Analytics service and set the Startup type to Manual.
Repeat all steps on each cluster node. Point to the SQLdm Repository installed on the first node.

Configure the Cluster Resource Group

Cluster Resource Group settings must be modified to allow SQLdm to recognize the cluster nodes.

To configure the Cluster Resource Group:

1. In the Cluster Administrator tool, create the Cluster Resource Group:
   - Select File > New > Group and give the group and description.
   - Add the nodes and click Finish.
2. In the Cluster Administrator tool, add a disk to the Cluster Resource Group:
1. Right-click the group you just created, select New > Resource and give it a name and description.
   For the Resource Type, select Physical Disk.
   Select the SQLdm group.
   Click Next until you get to the last window and click Finish.

2. In the Cluster Administrator tool, add an IP address for the Cluster Resource Group:
   Right-click the group and select New > Resource and give it a name and description.
   Select the Resource type IP Address.
   Enter the IP address that you acquired for your new Cluster Resource Group.
   Click Finish.

3. In the Cluster Administrator tool, add a Network Name resource to the Cluster Resource Group:
   Right-click the group and New > Resource and give it a name.
   Select the Resource type Network Name.
   Add the dependency for the IP address resource.
   Specify the name and click Finish.

4. Select a Disk that is in a Cluster Resource Group and create a directory on that disk that will be used to hold local data for the services.

5. On the primary node, create the default registry key and add the full data path value under HKEY_LOCAL_MACHINE\SOFTWARE\Idera\SQLdm\Default.

6. Create the DataPath value.

7. Select a Cluster Resource Group in which you want to define the SQLdm resources.

8. Create an IP Address resource for the SQLdm Services. If there is already an existing Network Name in the Cluster Resource Group, then you can skip this step.
   - Create an Network Name resource that will map to the IP Address
   - Create the Network Name resource and add the IP Address as a dependency

9. Create a Generic Service resource for the SQLdm Management Service.
   - Add the Disk for the shared data as a dependency
   - Add the Network Name as a dependency
   - Specify SQLdmManagementService$Default as the name of the service to start

10. After the service is added, right-click it and select Properties.
    - Select the Parameters tab and check the Use network name for computer name box
    - Select the Registry Replication tab and add SOFTWARE\Idera\SQLdm\Default
    - Click OK

11. Create a Generic Service resource for the SQLdm Collection Service.
    - Add the Disk for the shared data as a dependency
    - Add the Network Name as a dependency
    - Specify SQLdmCollectionService$Default as the name of the service to start

12. After the service is added, right-click it and select Properties.
    - Select the Parameters tab and check the Use network name for computer name box
    - Select the Registry Replication tab and add SOFTWARE\Idera\SQLdm\Default
    - Click OK

13. Create a Generic Service resource for the SQLdm Predictive Analytics Service.
- Add the Disk for the shared data as a dependency
- Add the Network Name as a dependency
- Specify SQLdmPredictiveAnalyticsService$Default as the name of the service to start

15. After the service is added, right-click it and select Properties.
   - Select the Parameters tab and check the Use network name for computer name box
   - Select the Registry Replication tab and add SOFTWARE\Idera\SQLdm\Default
   - Click OK

16. In Windows Services, bring the SQLdm Management service resource online.
17. In Windows Services, bring the SQLdm Collection service resource online.
18. In Windows Services, bring the SQLdm Predictive Analytics service resource online.

![Registry Editor](image)

**Complete the Cluster Configuration in SQL diagnostic manager**

1. Use the Management Service Configuration wizard to force the services to re-register.
2. Start the SQLdm Desktop Client and select SQLdm Repository. If the SQLdm Repository is clustered make sure to use the virtual server name for the SQL Server hosting the SQLdm Repository.
3. Verify that your configuration displays correctly in the SQLdm Desktop Client.
The following instructions guide you through the installation of SQLdm in a Windows Server 2008 based clustered environment.

**Things to consider before you install SQLdm in a clustered environment**

- You will need to create a separate cluster resource group to host SQLdm services. This allows the SQLdm services to fail without causing other resources in the group to failover.
- To create this cluster resource group you will need a dedicated IP, the NetBIOS name, and a dedicated disk for that resource group.
- SQLdm Collection, Management, and Predictive Analytics services will run as generic services on the newly created cluster resource group.
- You may need to provide the network Name and network IP as part of cluster configuration.

**Install SQLdm Services on Cluster Nodes**

To set up SQLdm services to work correctly when the primary cluster node they are hosted on fails, you need to install the services on each cluster node.

**To install SQLdm services on cluster nodes:**

1. Log on with an administrator account to the computer on which you want to install SQLdm.
2. Run `Setup.exe` in the root of the SQLdm installation kit on the first cluster node.
3. Click **Install** and then click **Install SQL diagnostic manager**.
4. Read the Welcome window, and then click **Next**.
5. Review and accept the license agreement by clicking **I accept the terms in the license agreement**, and then click **Next**.
6. Accept the default installation folder, or click **Change** to specify a different folder, and click **Next**.
7. Select the **Typical setup** type and click **Next**.
8. Select the SQL Server instance and enter a database name.
9. Enter the service account information and click **Next**.
10. Click **Install**.
11. In Windows Services, stop the SQLdm Management service and set the Startup type to **Manual**.
12. In Windows Services, stop the SQLdm Collection service and set the Startup type to **Manual**.
13. In Windows Services, stop the SQLdm Predictive Analytics service and set the Startup type to **Manual**.

Repeat the previous steps on each cluster node. Point to the SQLdm Repository installed on the first node.

**TIP**

The installations cannot be done concurrently, as the installers will collide when checking the repository. The installations must be done sequentially.

**Configure the Cluster Resource Group**

Cluster Resource Group settings must be modified to allow SQLdm to recognize the cluster nodes.

**To configure the Cluster Resource Group:**

1. Select a disk that is configured in the Cluster Resource Group and create a directory on that disk that will be used to hold local data for the services.
2. On the primary node, edit the registry and add a new key named Default to the key \Software\Idera\SQLdm. In this key add a string value called DataPath. Set its value to the full path to the data directory created in step 1 (as shown in the image at the end of the steps).

3. In the Failover Cluster Management tool, right-click on Services and Applications and select Configure a Service or Application.
   - Provide a name for the service
   - Select Generic Service
   - Select the SQLdm Management Service
   - Enter the network name and IP address for the services
   - Select the clustered storage that hosts the path defined in step 1
   - Skip the Registry Replication
   - Verify the settings and click Next to complete the wizard

4. In Services and Applications, click the newly-created service name in the Other Resources section and then on the SQLdm Management service and take it offline.

5. In Services and Applications, right-click on the newly created service and then on the SQLdm Management Service and select Properties.
   - On the General tab, select Use Network Name for computer name
   - On the Dependencies tab, add the Disk and Name of the service
   - On the Registry Replication tab, if blank, add SOFTWARE\Idera\SQLdm\Default\DataPath

6. Right-click on the new services node and select Add a Resource and Generic Service.

7. Select the Collection service from the list of services.

8. Select Next and finish out the wizard.

9. In Services and Applications, right-click on the newly created service and then on the SQLdm Collection Service and select Properties.
   - On the Dependencies tab, add the Disk and Name of the Service
   - On the General tab, select Use Network Name for computer name
   - On the Registry Replication tab, add SOFTWARE\Idera\SQLdm\Default\DataPath

10. Right-click the new services node and select Add a Resource and Generic Service.

11. Select the Predictive Analytics service from the list of resources.

12. Select Next and then Finish to exit the wizard.

13. In Services and Applications, click the newly-created service and then on the SQLdm Predictive Analytics service and select Properties.
   - On the General tab, select Use Network Name for computer name
   - On the Dependencies tab, add the Disk and Name of the service
   - On the Registry Replication tab, if blank, add SOFTWARE\Idera\SQLdm\Default\DataPath

14. In Windows Services, bring the SQLdm Management service resource online.

15. In Windows Services, bring the SQLdm Collection service resource online.

16. In Windows Services, bring the SQLdm Predictive Analytics service resource online.

17. On the active node on which you installed the SQLdm services, use the Management Service Configuration wizard (Start > All Programs > Idera > SQL diagnostic manager > Tools > Management Service Configuration Wizard) to force the services to re-register.

18. Test the configuration from the Management Service Configuration wizard before finishing the wizard.
Complete the Cluster Configuration in SQL diagnostic manager

Start the SQLdm Desktop Client and select SQLdm Repository. If the SQLdm Repository is clustered make sure to use the virtual server name for the SQL Server hosting the SQLdm Repository.
SQL diagnostic manager allows you to monitor your clustered SQL Server environment including alerts when a failover occurs and when the preferred active cluster node changes.

An important way to track clustering events using SQLdm is through the usage of cluster alerts.

**Cluster Failover Alert**

By default, this alert is enabled and set to critical. When a failover occurs, SQLdm generates an alert and then performs one of the following actions:

- Clears after 12 hours if no additional alerts are generated on the cluster.
- Generates an additional alert if another cluster failover occurs.
- Clears instantly when manually cleared.

**Cluster Active Node Alert**

By default, this alert is enabled and pre-populated with the first node added to SQLdm. You can change this setting on the Cluster Settings tab of the Monitored SQL Server Properties dialog box.
<table>
<thead>
<tr>
<th>Repository ID</th>
<th>Windows Event ID</th>
<th>Category</th>
<th>Metric Name</th>
<th>Metric Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>101</td>
<td>Resources</td>
<td>SQL Server CPU Usage (Percent)</td>
<td>Percentage of CPU Time used by the SQL Server instance out of the total on the SQL Server computer.</td>
</tr>
<tr>
<td>1</td>
<td>202</td>
<td>Sessions</td>
<td>User Connections (Percent)</td>
<td>Percentage of the user connections allowed by the currently used SQL Server instances.</td>
</tr>
<tr>
<td>4</td>
<td>708</td>
<td>Services</td>
<td>Non-distributed Transactions (Count)</td>
<td>Number of replication transactions published by the Publisher but not received by the Distributor.</td>
</tr>
<tr>
<td>5</td>
<td>809</td>
<td>Services</td>
<td>Unsubscribed Transactions (Count)</td>
<td>Number of replication transactions received by the Distributor but not received by a Subscriber.</td>
</tr>
<tr>
<td>6</td>
<td>910</td>
<td>Queries</td>
<td>Oldest Open Transaction (Minutes)</td>
<td>Time in minutes that the longest running transaction in the database has used since issuing the Begin Transaction statement.</td>
</tr>
<tr>
<td>7</td>
<td>1011</td>
<td>Databases</td>
<td>Table Fragmentation (Percent)</td>
<td>Percentage of pages in the table that are fragmented.</td>
</tr>
<tr>
<td>8</td>
<td>1112</td>
<td>Databases</td>
<td>Database Full (Percent)</td>
<td>Percentage of the allowable disk space for the database currently used by the sum of data, text, and indexes. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings, and available disk space.</td>
</tr>
<tr>
<td>9</td>
<td>1213</td>
<td>Databases</td>
<td>Log Full (Percent)</td>
<td>Percentage of the allowable disk space for the database currently used by the transaction log. Allowable disk space for the database is calculated by taking into account the current allocated space, auto-growth settings and available disk space.</td>
</tr>
<tr>
<td>10</td>
<td>1314 - Running 1315 - Stopped 1316 - Other</td>
<td>Services</td>
<td>SQL Server Agent Status</td>
<td>Status of the SQL Server Agent service.</td>
</tr>
<tr>
<td>11</td>
<td>1417</td>
<td>Databases</td>
<td>Database Read/Write Error Occurred</td>
<td>An I/O error occurred while attempting to read from or write to a database file.</td>
</tr>
<tr>
<td>12</td>
<td>1518 - Running 1519 - Stopped</td>
<td>Services</td>
<td>SQL Server Status</td>
<td>Status of the SQL Server service.</td>
</tr>
<tr>
<td>Repository ID</td>
<td>Windows Event ID</td>
<td>Category</td>
<td>Metric Name</td>
<td>Metric Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
<td>----------</td>
<td>-------------</td>
<td>--------------------</td>
</tr>
<tr>
<td></td>
<td>1520 - Paused</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1521 - Not Admin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1522 Not Accessible</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1523 - Other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1624</td>
<td>Resources</td>
<td>SQL Server Memory Usage (Percent)</td>
<td>Percentage of memory used by the SQL Server instance out of the total memory on the SQL Server computer.</td>
</tr>
<tr>
<td>14</td>
<td>1725</td>
<td>Databases</td>
<td>Database Status</td>
<td>Operational status of the database, such as Normal, Offline, or Suspect.</td>
</tr>
<tr>
<td>17</td>
<td>2028</td>
<td>Services</td>
<td>Unsubscribed Transactions (Seconds)</td>
<td>Time in seconds that a replication transaction is holding at the Distributor and waiting to be received by a Subscriber. This alert is for visual indicators in the Desktop Client only. No notifications are sent.</td>
</tr>
<tr>
<td>18</td>
<td>2129 - Running</td>
<td>Services</td>
<td>DTC Status</td>
<td>Status of the Distributed Transaction Coordinator service.</td>
</tr>
<tr>
<td></td>
<td>2130 - Stopped</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2131 - Other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>2232 - Running</td>
<td>Services</td>
<td>Full-Text Search Status</td>
<td>Status of the Full-Text Search service.</td>
</tr>
<tr>
<td></td>
<td>2233 - Other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>N/A</td>
<td>Databases</td>
<td>Index Row hits (Percent)</td>
<td>Percentage of density of an index as used by the SQL Query Optimizer to determine index effectiveness. A high density indicates many duplicates and an inefficient index while a low density indicates high selectivity and an effective index. This alert is for visual indicators in the Desktop Client only. No notifications are sent.</td>
</tr>
<tr>
<td>21</td>
<td>N/A</td>
<td>Services</td>
<td>Last Full-Text Catalog Update (Hours)</td>
<td>Time in hours since the last Full-Text Catalog update was performed and beyond which the statistics data is considered outdated. This alert is for visual indicators in the Desktop Client only. No notifications are sent.</td>
</tr>
<tr>
<td>22</td>
<td>2534</td>
<td>Sessions</td>
<td>SQL Server Response Time (Milliseconds)</td>
<td>Time in milliseconds SQL diagnostic manager currently needs to send a simple SQL command to the SQL Server instance, have it processed, and receive the returned result set.</td>
</tr>
<tr>
<td>23</td>
<td>4010 - Available</td>
<td>Operational</td>
<td>OS Metrics Collection Status</td>
<td>Status of SQL diagnostic manager OS Metrics Collection.</td>
</tr>
<tr>
<td>Repository ID</td>
<td>Windows Event ID</td>
<td>Category</td>
<td>Metric Name</td>
<td>Metric Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>----------</td>
<td>-------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>4013</td>
<td>4020</td>
<td>Resources</td>
<td>OS Used Memory (Percent)</td>
<td>Percentage of total memory used on the SQL Server computer.</td>
</tr>
<tr>
<td>4020</td>
<td>4030</td>
<td>Resources</td>
<td>OS Paging (Per Second)</td>
<td>Page fault rate for all processes on the SQL Server computer.</td>
</tr>
<tr>
<td>4030</td>
<td>4040</td>
<td>Resources</td>
<td>OS Processor Time (Percent)</td>
<td>Percentage of CPU time used by all processes on the SQL Server computer.</td>
</tr>
<tr>
<td>4040</td>
<td>4050</td>
<td>Resources</td>
<td>OS Privileged Time (Percent)</td>
<td>Percentage of CPU time spent by all processes executing in privileged mode on the SQL Server computer.</td>
</tr>
<tr>
<td>4050</td>
<td>4060</td>
<td>Resources</td>
<td>OS User Time (Percent)</td>
<td>Percentage of CPU time spent by all processes executing in user mode on the SQL Server computer.</td>
</tr>
<tr>
<td>4060</td>
<td>4070</td>
<td>Resources</td>
<td>OS Processor Queue Length (Count)</td>
<td>Number of ready threads in the processor queue on the SQL Server Computer.</td>
</tr>
<tr>
<td>4070</td>
<td>4080</td>
<td>Resources</td>
<td>OS Disk Time (Percent)</td>
<td>Percentage of elapsed time that all the disks were busy servicing read and write requests on the SQL Server computer.</td>
</tr>
<tr>
<td>4080</td>
<td>4090</td>
<td>Resources</td>
<td>OS Average Disk Queue Length (Count)</td>
<td>Average number of both read and write requests that were queued for all the disks on the SQL Server computer.</td>
</tr>
<tr>
<td>4090</td>
<td>2737</td>
<td>Sessions</td>
<td>Session CPU Time (Seconds)</td>
<td>CPU Time in seconds used by the SQL Server session since the last scheduled refresh.</td>
</tr>
<tr>
<td>2737</td>
<td>2838</td>
<td>Sessions</td>
<td>Session CPU Time (Seconds)</td>
<td>Time in seconds that the SQL Server session has been blocking other sessions.</td>
</tr>
<tr>
<td>2838</td>
<td>2940</td>
<td>Services</td>
<td>SQL Server Agent Long Running Job</td>
<td>Percentage of time a scheduled SQL Server Agent job is taking over the average successful completion time for this job in the current job history.</td>
</tr>
<tr>
<td>2940</td>
<td>3039</td>
<td>Services</td>
<td>SQL Server Agent Job Failure</td>
<td>The scheduled SQL Server Agent job ended with a Failed status.</td>
</tr>
<tr>
<td>3039</td>
<td>4001 - Enabled 4002 - Disabled</td>
<td>Operational</td>
<td>Maintenance Mode Enabled</td>
<td>Maintenance Mode has been enabled for the SQL Server instance. SQL diagnostic manager does not collect scheduled data or process alerts.</td>
</tr>
<tr>
<td>4001 - Enabled 4002 - Disabled</td>
<td>4200</td>
<td>Operational</td>
<td>CLR Enabled</td>
<td>The Common Language Runtime (CLR) configuration option is enabled for the SQL Server instance. Managed code can be run in the .NET Framework.</td>
</tr>
<tr>
<td>Repository ID</td>
<td>Windows Event ID</td>
<td>Category</td>
<td>Metric Name</td>
<td>Metric Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>---------------</td>
<td>--------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>50</td>
<td>4210</td>
<td>Operational</td>
<td>OLE Automation Disabling</td>
<td>The OLE Automation Procedures configuration option is disabled (2005) or the procedures were dropped (2000) for the SQL Server instance. SQL diagnostic manager does not collect data such as OS metrics and service details.</td>
</tr>
<tr>
<td>51</td>
<td>4220</td>
<td>Queries</td>
<td>Query Monitor Events (Count)</td>
<td>The number of queries captured by the Query Monitor on the SQL Server instance during the last scheduled refresh.</td>
</tr>
<tr>
<td>52</td>
<td>4230</td>
<td>Operational</td>
<td>SQL diagnostic manager Collection Service Status</td>
<td>Status of the SQL diagnostic manager Collection Service.</td>
</tr>
<tr>
<td>53</td>
<td>4240</td>
<td>Operational</td>
<td>SQL Server Agent XPs Disabled</td>
<td>The Agent XPs configuration option is disabled (2005) or the procedures were dropped (2000) for the SQL Server instance. SQL diagnostic manager does not collect SQL Agent job details.</td>
</tr>
<tr>
<td>54</td>
<td>4250</td>
<td>Operational</td>
<td>WMI Service Unavailable</td>
<td>The WMI service is unavailable on the SQL Server computer. SQL diagnostic manager does not collect OS metrics.</td>
</tr>
<tr>
<td>57</td>
<td>4310</td>
<td>Sessions</td>
<td>Client Computers (Count)</td>
<td>Number of unique client computers connected to the SQL Server instance.</td>
</tr>
<tr>
<td>58</td>
<td>4320</td>
<td>Sessions</td>
<td>Blocked Sessions (Count)</td>
<td>Number of sessions on the SQL Server instance blocked by other sessions holding requested locks.</td>
</tr>
<tr>
<td>59</td>
<td>4330</td>
<td>Operational</td>
<td>SQL Server Data Used (Percent)</td>
<td>Percentage of allocated data space used across all data files on the SQL Server instance.</td>
</tr>
<tr>
<td>60</td>
<td>4340</td>
<td>Operational</td>
<td>SQL Server Log Used (Percent)</td>
<td>Percentage of allocated log space used across all log files on the SQL Server instance.</td>
</tr>
<tr>
<td>62</td>
<td>4110</td>
<td>Resources</td>
<td>OS Disk Time Per Disk (Percent)</td>
<td>Percentage of elapsed time that an individual disk was busy servicing read and write requests on the SQL Server computer. This event alerts separately for each disk on a server.</td>
</tr>
<tr>
<td>63</td>
<td>4120</td>
<td>Resources</td>
<td>OS Average Disk Queue Length Per Disk (Count)</td>
<td>Average number of both read and write requests queued for an individual disk on the SQL Server computer. This event alerts separately for each disk on a server.</td>
</tr>
<tr>
<td>64</td>
<td>4100</td>
<td>Resources</td>
<td>OS Disk Full (Percent)</td>
<td>Percentage of space used on a logical disk on the SQL Server computer. This event alerts separately for each disk on a server.</td>
</tr>
<tr>
<td>65</td>
<td>2940</td>
<td>Services</td>
<td>SQL Server Agent Long Running Job (Minutes)</td>
<td>Maximum number of minutes a scheduled SQL Server Agent job should take to complete.</td>
</tr>
<tr>
<td>66</td>
<td>4400</td>
<td>Logs</td>
<td>SQL Server Error Log</td>
<td>The number of error log events requiring attention based on either message severity or specified text.</td>
</tr>
<tr>
<td>Repository ID</td>
<td>Windows Event ID</td>
<td>Category</td>
<td>Metric Name</td>
<td>Metric Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>67</td>
<td>4410</td>
<td>Logs</td>
<td>SQL Server Agent Log</td>
<td>The number of agent log events requiring attention based on either message severity or specified text. When configuring this alert, use the indicator or grid to specify which log message severity corresponds to warning or critical. Use the advanced options to specify additional filters.</td>
</tr>
<tr>
<td>68</td>
<td>5105</td>
<td>Databases</td>
<td>Mirroring Unsent Log (KB)</td>
<td>Size of the unsent log in the send queue on the principal in kilobytes.</td>
</tr>
<tr>
<td>69</td>
<td>5106</td>
<td>Databases</td>
<td>Mirroring Unre-stored Log (KB)</td>
<td>Size of the redo queue on the mirror in kilobytes.</td>
</tr>
<tr>
<td>70</td>
<td>5107</td>
<td>Databases</td>
<td>Mirroring Oldest Unsent Transaction (Minutes)</td>
<td>The age of the oldest unsent transaction in minutes on the principal. This is only meaningful on the principal server instance.</td>
</tr>
<tr>
<td>71</td>
<td>5108</td>
<td>Databases</td>
<td>Mirror Commit Overhead (Milliseconds)</td>
<td>Average delay in milliseconds on the principal server instance for each transaction because of database mirroring.</td>
</tr>
<tr>
<td>72</td>
<td>5109</td>
<td>Databases</td>
<td>Mirroring Status</td>
<td>Status of the mirrored database such as synchronized suspended or disconnected.</td>
</tr>
<tr>
<td>73</td>
<td>5110</td>
<td>Databases</td>
<td>Mirroring Preferred Configuration</td>
<td>This operational configuration of this mirroring session changed.</td>
</tr>
<tr>
<td>74</td>
<td>5111</td>
<td>Databases</td>
<td>Mirrored Server Role Change</td>
<td>A mirrored database role changed.</td>
</tr>
<tr>
<td>75</td>
<td>5112</td>
<td>Databases</td>
<td>Mirroring Witness Connection</td>
<td>The mirroring witness is not connected.</td>
</tr>
<tr>
<td>76</td>
<td>4130</td>
<td>Resources</td>
<td>Page Life Expectancy</td>
<td>The number of seconds a page stays in the buffer pool without references.</td>
</tr>
</tbody>
</table>
Troubleshoot WMI connectivity issues

The user account under which the Collection Service runs must have administrator permissions on the remote server to be able to use WMI.

The most frequently encountered problems with WMI relate to RPC traffic not getting through to the remote computer. The other most frequently encountered issue involves DCOM/WMI permissions.

The following links may provide additional information about how to troubleshoot WMI connectivity issues:

- Securing a remote WMI Connection
- Help with Scripts

Using WbemTest (Windows Management Instrumentation Tester)

You can use the WbemTest tool to connect to a server and issue WMI queries. Download this tool from Microsoft TechNet. This tool can help you test and troubleshoot WMI issues.

**To use WbemTest:**

1. Run `wbemtest.exe`.
2. Click Connect.
3. In the NameSpace test box, enter `\server\root\cimv2` where server is the name of the server you want to connect to.
4. Click Connect.
5. Click Query.
6. Enter `select* from win32_process`.
7. Click Apply.

If WbemTest was able to connect to the remote server and issue the query using WMI, you should see a query result with output. In this case, WMI to the required server is working and no further action is needed. For more information on the Windows Management Instrumentation Tester, refer to Windows Management Instrumentation Tester overview.

If you receive an error message, use the following processes to help identify and resolve the issue.

**Error: The RPC Server Is Unavailable**

This error usually indicates that the RPC traffic is not getting to the remote server, or there is no RPC listener on the remote server.

**To troubleshoot this RPC error:**

1. Ensure the Remote Procedure Call (RPC) service is running on the remote server.
2. Verify that there is a TCP listener on the remote server by running the `netstat -nao` command and verifying that there is the following entry: TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 1304
3. In the Tools subdirectory, run `rpcping /s <servername> /t ncacn_ip_tcp` where `<servername>` is the name of the remote server. This command verifies that RPC can communicate with the remote server. The output should be similar to:
   
   Completed 1 calls in 15 ms
   66 T/S or 15.000 ms/T

4. Ensure that the traffic is not blocked by local or internal network firewalls. Either disable the firewall or configure the Windows firewall to allow incoming RPC traffic.
5. Try to use the remote server IP address instead of the server name. If the IP address works, you may have a DNS issue.

6. If the remote server resides in a different domain, the two domains may not trust each other, or the user account does not have administrator permissions on the remote server/domain.

7. If both computers are in the same domain, and the user account has administrator permissions, try rejoining both computers to the domain.

**Error: Access Denied**

This error can indicate permission issues.

**To troubleshoot this access error:**

1. If the remote computer is running Windows XP, make sure it is not set to Force Guest. This setting forces any connection to be impersonated as Guest.
   a. Open the Local Security Policy console from Administrative Tools.
   d. Change the settings from Guest Only to Classic.

2. Ensure that DCOM is enabled on the remote server:
   a. Run **DcomCnfg** on the remote server.
   b. Click Component Services.
   c. Expand Computers.
   d. Right click My Computer and select Properties.
   e. Click the Default Properties tab.
   f. Ensure Enable Distributed COM on this computer is checked.

3. Ensure the correct DCOM remote launch and activation permissions are configured:
   a. Run **DcomCnfg** on the remote server.
   b. Click Component Services.
   c. Expand Computers.
   d. Right click My Computer and select Properties.
   e. Ensure Enable Distributed COM on this computer is checked.
   f. Click the Com Security tab.
   g. Under Launch and Activation Permissions, click Edit Limits.
   h. In the Launch Permissions dialog box, make sure your user account or group is listed in the Groups or user names list. If your user account or group is not listed, click Add and add it to the list.
   i. In the Launch Permission dialog box, select your user account or group in the Group or user names list. In the Allow column under Permissions for User, select Remote Launch and Remote Activation, and then click OK.
4. Ensure the correct DCOM remote access permissions are configured:
   a. Run DcomCnfg on the remote server.
   b. Click Component Services.
   c. Expand Computers.
   d. Right click My Computer and select Properties.
   e. Ensure Enable Distributed COM on this computer is checked.
   f. Click the Com Security tab.
   g. Under Access Permissions, click Edit Limits.
   h. In the Access Permission dialog box, select ANONYMOUS LOGON name in the Group or user names list. In the Allow column under Permissions for User, select Remote Access, and then click OK.

5. Ensure the correct WMI namespace permissions are configured.
   a. Run wmimgmt.msc.
   b. Right-click WMI Control, and then select Connect to another computer.
   c. Enter the remote server name, and then click OK.
   d. Right-click WMI Control, and then select Properties.
   e. In the Security tab, select the namespace, and then click Security.
   f. Locate the appropriate account, and then check Remote Enable in the Permissions list.