1. SQL Diagnostic Manager Home ................................. 8
  1.1 Access the SQLdm Release Notes .......................... 8
    1.1.1 New features and fixed issues ...................... 8
    1.1.2 Previous features and fixed issues ............... 9
    1.1.3 Known issues .................................... 22
    1.1.4 Recommended Idera Solutions ..................... 27
  1.2 Get started ............................................. 28
    1.2.1 Upgrade to this release ......................... 28
      1.2.1.1 Upgrade SQLdm in a clustered environment 29
      1.2.1.2 Perform a distributed upgrade .................. 29
      1.2.1.3 Upgrade SQLdm console only ................... 30
      1.2.1.4 Test your deployment .......................... 30
    1.2.2 Welcome to SQLdm ................................ 30
      1.2.2.1 What is SQLdm? ............................ 31
      1.2.2.2 How does SQLdm help me? ...................... 31
    1.2.3 Use the SQLdm interface ............................ 32
      1.2.3.1 Navigation pane .............................. 32
      1.2.3.2 View SQLdm Today ............................. 33
      1.2.3.3 Refresh and pause views ...................... 34
      1.2.3.4 Filter views .................................. 35
      1.2.3.5 Charts .......................................... 35
      1.2.3.6 Monitor multiple SQL Server instances ....... 37
      1.2.3.7 Server Configuration Properties ............... 38
    1.2.4 Use the SQLdm web console ......................... 43
      1.2.4.1 Navigation pane ................................ 43
      1.2.4.2 Customer support portal ....................... 43
      1.2.4.3 About Idera .................................... 44
      1.2.4.4 Idera products .................................. 44
      1.2.4.5 Contact Idera .................................. 44
      1.2.4.6 Legal notice .................................... 44
      1.2.4.7 Document conventions ......................... 46
  1.3 Installation and deployment ......................... 48
    1.3.1 Supported installation scenarios .................. 48
    1.3.2 Choose your deployment strategy ................... 49
    1.3.3 How SQLdm works with a virtual environment .... 50
    1.3.4 Deploy the Idera Dashboard and SQLdm ............ 53
      1.3.4.1 Product components and architecture .......... 54
      1.3.4.2 Product requirements .......................... 56
      1.3.4.3 How to install the Idera Dashboard and SQLdm 60
      1.3.4.4 Log in the Idera Dashboard and web console .... 68
    1.3.5 Deploy SQLdm Mobile or Idera Newsfeed .......... 68
      1.3.5.1 SQLdm Mobile components and architecture including Idera Newsfeed 69
      1.3.5.2 SQLdm Mobile and Idera Newsfeed requirements 70
      1.3.5.3 How to install SQLdm Mobile and Idera Newsfeed 72
      1.3.5.4 How to install Idera Newsfeed only .......... 74
    1.3.6 Deploy SQLdm Console only ......................... 75
      1.3.6.1 How to install SQLdm Console only .......... 75
    1.3.7 Deploy SQLdm in a high availability group ....... 76
      1.3.7.1 AlwaysOn availability group listener name and port designation 78
  1.4 Configure your deployment .............................. 81
    1.4.1 Connect to the SQLdm Repository .................. 81
    1.4.2 Add your SQL Server instances .................... 82
      1.4.2.1 Configure authentication ....................... 82
      1.4.2.2 Select instances to monitor .................... 83
      1.4.2.3 Configure data collection ...................... 83
      1.4.2.4 Configure OS metric collection ............... 84
      1.4.2.5 Select an alert template and tags ............. 84
    1.4.3 Configure your virtual machine connections .... 85
      1.4.3.1 Configure your VM host server connection .... 87
      1.4.3.2 Define common properties ...................... 88
      1.4.3.3 Configure the Alerts view .................... 89
      1.4.3.4 Configure notification settings ............... 89
    1.4.4 Configure console options .......................... 88
      1.4.4.1 Add a custom counter .......................... 92
    1.4.5 Set your SQLdm Repository grooming schedule .... 91
    1.4.6 Use custom counters to track metrics ............. 91
      1.4.6.1 Add a custom counter .......................... 92
      1.4.6.2 Edit a custom counter ......................... 97
      1.4.6.3 Delete a custom counter ....................... 98
      1.4.6.4 Test a custom counter ......................... 98
      1.4.6.5 Link custom counter to monitored instance .... 98
    1.4.7 Work with tags ..................................... 99
      1.4.7.1 Manage tags ..................................... 99
1.9.4 Client Computers .................................................. 277
1.9.5 SQL Compilations (Per Second) ................................. 277
1.9.6 Data Used (Percent) .................................................. 278
1.9.7 Database Status ...................................................... 278
1.9.8 Disk Queue Length .................................................. 278
1.9.9 Disk Time (Percent) .................................................. 279
1.9.10 I/O Waits ............................................................ 279
1.9.11 Lock Waits ........................................................... 279
1.9.12 Log Flushes (Per Second) ........................................ 279
1.9.13 Memory Page Faults (Per Second) ............................. 280
1.9.14 Memory Paging ...................................................... 280
1.9.15 Memory Waits ....................................................... 281
1.9.16 Other Waits .......................................................... 281
1.9.17 Packets Received (Per Second) ................................. 281
1.9.18 Packets Sent and Received ....................................... 281
1.9.19 Packets Sent (Per Second) ....................................... 281
1.9.20 Page Reads (Per Second) ......................................... 281
1.9.21 Page Writes (Per Second) ........................................ 282
1.9.22 Percent of SQL Server Data Space Used ...................... 283
1.9.23 Percent of SQL Server Log Space Used ....................... 283
1.9.24 Physical Page Reads and Writes ............................... 283
1.9.25 Processor Queue Length .......................................... 283
1.9.26 Re-compiles (Per Second) ....................................... 283
1.9.27 Signal waits .......................................................... 284
1.9.28 SQL Server Batches Received (Per Second) ................ 284
1.9.29 SQL Server CPU Usage (Percent) .............................. 284
1.9.30 Log Space Used (Percent) ........................................ 285
1.9.31 SQL Server Memory Usage ...................................... 285
1.9.32 SQL Server Response Time (ms) ............................... 285
1.9.33 Total Memory Usage ............................................... 285
1.9.34 Transaction Log Waits ............................................. 285
1.9.35 Transactions (Per Second) ....................................... 286
1.9.36 Wait type category - Backup .................................... 286
1.9.37 Wait type category - I/O Page Latch ......................... 286
1.9.38 Wait type category - Lock ........................................ 286
1.9.39 Wait type category - Memory .................................... 287
1.9.40 Wait type category - Non-I/O Page Latch .................... 287
1.9.41 Wait type category - Non-Page Latch ......................... 287
1.9.42 Wait type category - Other ....................................... 288
1.9.43 Wait type category - Transaction Log ......................... 288
1.10 Metric alerts ............................................................. 291
    1.10.1 Availability Group Estimated Data Loss (Seconds) alert 291
    1.10.2 Availability Group Estimated Recovery time (Seconds) alert 293
    1.10.3 Availability Group Log Send Queue Size (KB) alert 293
    1.10.4 Availability Group Redo Queue Size (KB) alert 293
    1.10.5 Availability Group Redo Rate (KB/sec) alert 293
    1.10.6 Availability Group Role Change alert 293
    1.10.7 Availability Group Synchronization Health alert 294
    1.10.8 Availability Group Synchronization Performance (Seconds) alert 294
    1.10.9 Average Disk Milliseconds Per Read alert 294
    1.10.10 Average Disk Milliseconds Per Transfer alert 294
    1.10.11 Average Disk Milliseconds Per Write alert 294
    1.10.12 Blocked Sessions (Count) alert 294
    1.10.13 Blocking Session Wait Time (Seconds) alert 295
    1.10.14 Client Computers (Count) alert 296
    1.10.15 CLR Enabled alert ............................................... 297
    1.10.16 Cluster Active Node alert .................................... 297
    1.10.17 Cluster Failover alert ........................................ 297
    1.10.18 Data File Autogrow alert .................................... 297
    1.10.19 Database Full (Percent) alert 298
    1.10.20 Database Full (Size) alert .................................... 298
    1.10.21 Database Read/Write Error Occurred alert 298
    1.10.22 Database Status alert ......................................... 298
    1.10.23 Deadlock alert .................................................. 298
    1.10.24 Disk Reads Per Second alert 299
    1.10.25 Disk Transfers Per Second alert 299
    1.10.26 Disk Writes Per Second alert 300
    1.10.27 Distribution Latency (Seconds) alert 300
    1.10.28 Error C Status alert .......................................... 300
    1.10.29 Host CPU Usage (Percent) alert 300
    1.10.30 Host Memory Swap Detected alert 301
    1.10.31 Host Memory Usage (Percent) alert 301
1.10.32 Host Power State alert .......................................................... 302
1.10.33 Full-Text Search Status alert .................................................. 302
1.10.34 Index Row Hits (Percent) alert ................................................. 302
1.10.35 Last Full-Text Catalog Update (Hours) alert ............................. 302
1.10.36 Log File Autogrow alert .......................................................... 303
1.10.37 Log Full (Percent) alert ......................................................... 303
1.10.38 Log Full (Size) alert .............................................................. 303
1.10.39 Longest Running Version Store Transaction (Minutes) alert ........ 303
1.10.40 Maintenance Mode Enabled alert .............................................. 304
1.10.41 Mirror Commit Overhead (Milliseconds) alert ......................... 304
1.10.42 Mirrored Server Role Change alert ........................................... 304
1.10.43 Mirroring Oldest Unsent Transactions (Minutes) alert ............... 304
1.10.44 Mirroring Preferred Configuration alert .................................... 305
1.10.45 Mirroring Status alert ............................................................ 305
1.10.46 Mirroring Unrestored Log (KB) alert ....................................... 305
1.10.47 Mirroring Unsente Log (KB) alert ........................................... 305
1.10.48 Mirroring Witness Connection alert .......................................... 306
1.10.49 Non-Distributed Transactions (Count) alert ............................. 306
1.10.50 Oldest Open Transaction (Minutes) alert .................................. 306
1.10.51 OLE Automation Disabled alert ................................................ 306
1.10.52 OS Average Disk Queue Length (Count) alert .......................... 307
1.10.53 OS Disk Free Space (Size) alert .............................................. 307
1.10.54 OS Disk Full (Percent) alert ................................................... 307
1.10.55 OS Disk Time (Percent) alert .................................................. 308
1.10.56 OS Disk Time Per Disk (Percent) alert .................................... 308
1.10.57 OS Memory Usage (Percent) alert .......................................... 308
1.10.58 OS Metrics Collection Status alert ......................................... 309
1.10.59 OS Paging (Per Second) alert ................................................ 309
1.10.60 OS Privileged Time (Percent) alert ........................................ 309
1.10.61 OS Processor Queue Length (Count) alert ............................... 309
1.10.62 OS Processor Time (Percent) alert ......................................... 310
1.10.63 OS User Time (Percent) alert ................................................ 310
1.10.64 Page Life Expectancy alert ................................................... 310
1.10.65 Procedure Cache Hit Ratio (Percent) alert ............................... 311
1.10.66 Query Monitor Events (Count) alert ....................................... 311
1.10.67 Session CPU Time (Seconds) alert ......................................... 311
1.10.68 Session Tempdb Space Usage (MB) alert ................................ 311
1.10.69 SQL Server Agent Job Completion alert .................................. 312
1.10.70 SQL Server Agent Job Failure alert ........................................ 312
1.10.71 SQL Server Agent Log alert .................................................. 312
1.10.72 SQL Server Agent Long Running Job (Minutes) alert ............... 312
1.10.73 SQL Server Agent Long Running Job (Percent) alert ............... 312
1.10.74 SQL Server Agent Status alert ............................................... 313
1.10.75 SQL Server Agent XPs Disabled alert .................................... 313
1.10.76 SQL Server CPU Usage (Percent) alert ................................... 313
1.10.77 SQL Server Data Used (Percent) alert .................................... 313
1.10.78 SQL Server Error Log alert ................................................... 314
1.10.79 SQL Server Log Used (Percent) alert .................................... 314
1.10.80 SQL Server Memory Usage (Percent) alert ............................. 315
1.10.81 SQL Server Response Time (Milliseconds) alert .................... 315
1.10.82 SQL Server Status alert ....................................................... 316
1.10.83 SQIDm Service Status alert .................................................. 316
1.10.84 Tempdb Contention (ms) alert ............................................... 316
1.10.85 Unsubscribed Transactions (Count) alert ................................ 316
1.10.86 Unsubscribed Transactions (Seconds) alert ............................. 316
1.10.87 User Connections (Percent) alert .......................................... 317
1.10.88 Version Store Generation Ratio alert .................................... 317
1.10.89 Version Store Size (MB) alert ............................................... 317
1.10.90 VM CPU Ready Wait Time (ms) alert .................................... 317
1.10.91 VM CPU Usage (Percent) alert ............................................. 318
1.10.92 VM Host Server Change alert ............................................... 318
1.10.93 VM Memory Swap Delay Detected alert .................................. 318
1.10.94 VM Memory Usage (Percent) alert ........................................ 318
1.10.95 VM Power State alert .......................................................... 319
1.10.96 VM Reclaimed/Ballooned Memory (KB) alert ......................... 319
1.10.97 VM Resource Configuration Change alert ............................... 320
1.10.98 VM Resource Limits Detected alert ....................................... 320
1.10.99 VM Resource Limits Detected alert ....................................... 320
1.10.100 WMI Service Unavailable alert ........................................... 320
1.10.101 Availability Group Preferred Replica alert .............................. 320
1.10.102 SQLdm Repository Timed Out alert .................................... 321
Access the SQLdm Release Notes

Idera 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 agent-less 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.

To get a glimpse into the newest features, fixed issues, and known issues in this release of SQL diagnostic manager, review the following sections of the Release Notes:

- See 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

New features and fixed issues

SQL diagnostic manager provides the following new features and fixed issues.

9.0 New features

Integration with the Idera Dashboard

SQLdm 9.0 now integrates with the Idera Dashboard, a common technology framework, designed to support the Idera product suite. Users are able to obtain an overview of the status of their SQL Servers and hosted databases all in a consolidated view and navigate to individual product dashboards for details. The Idera Dashboard provides a central set of services for managing users, product registry, instance registry, aggregated alerts across Idera applications, a central web server, and tags for grouping instances. For additional information, see Navigate the Idera Dashboard.

New advanced query views

SQLdm 9.0 takes SQL diagnostic manager’s capabilities around query monitoring and diagnostics to the next level with a new Main Query view along with two low-level views (Query Signature and Query Details views). Users are now able to explore query data in multiple ways and easily get a handle of query performance in their environment. In SQLdm 9.0 query monitoring is enabled by default for SQL Server instances running SQL Server 2008+, and collection is performed through Extended Events which has minimum impact on your SQL Server performance. Improved query monitoring allows for capturing of Query Execution Plans and provides graphical representations of the same in the Plan Diagram of the Query Details view. In addition, you can view the actual XML, SQL Text, and referenced columns of a query execution plan. For additional information, see View your SQL Server queries information.
Improved and integrated SQLdm web console

SQLdm 9.0 provides a fully functional web console that integrates seamlessly with the Idera Dashboard (common technology framework) and is easy to access from any connected browser. The SQLdm web console provides the overall status of a SQL Server environment to help diagnose any issues. Users can quickly view the status of a single instance or instance groups, view all active alerts in their SQL Server environment and drill down for details, or select different sub-views to diagnose performance issues faster, and view top values for selected metrics. In addition, users are able to access the new advanced query views (see above) only from the SQLdm web console and not from the SQLdm desktop client console. For additional information, see Navigate the web console dashboard.

Licensing and version upgrading –

Upgrade to SQLdm 9.0

Customers upgrading from previous SQLdm versions to SQLdm 9.0 need a new license key or it is not possible to continue with the upgrade process. Users can access the Idera Customer Portal at the moment of upgrade from the license management window in SQLdm (Help > Manage Licenses) or directly at https://idera.secure.force.com. Information necessary to generate a new license key is provided in the license management screen or the portal.

Help documentation –

SQLdm Help (User Guide and Release Notes) is now wiki-hosted and available at wiki.idera.com > SQL Diagnostic Manager.

9.0 Fixed issues

Query Monitor

- An issue preventing users from enabling the Query Monitor in monitored SQL Server 2000 instances no longer occurs.

Queries

- An issue in the Queries tab and corresponding Statement mode, Signature mode and query history views, causing queries exported into Excel to truncate, no longer occurs.

Alerts

- This release fixes an issue that prevented users from knowing when Repository grooming had timed out. SQLdm 9.0 includes a new alert SQLdm Repository Grooming Time out that notifies users when SQLdm no longer deletes stored metrics in the Repository at the set schedule. To configure this alert access the Alert Configuration window.

Reports

- An issue preventing users from running the Disk Details report of the Resources analysis reports no longer occurs.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Previous features and fixed issues

This build includes many fixed issues, including the following updates from previous releases.

8.6 New features

Full support for SQL Server Express instances


Reporting on Query Wait Statistics

You can now run the Query Wait Statistics report to quickly analyze different wait type categories on your SQL Server and identify where your biggest bottlenecks are occurring impacting your performance. For additional information about the new Query Waits Statistics report, see Query Waits Statistics report.

Query Waits information is now extended to the Top Servers and Top Databases reports

SQLdm has extended Query Waits information to the Top Servers and Top Databases report, providing users insight into the queries on your
SQL Server instance that exceed a specific wait threshold. Users can now view information on Top Servers by wait time and Top Databases by wait time. For additional information, see Top Servers and Top Databases.

New database alert for Availability Group Preferred Replica

SQLdm triggers an Availability Group Preferred Replica whenever the primary role changes to a different replica and is unavailable for a defined amount of time in minutes.

8.6 Fixed Issues

Upgrade

- Users upgrading from SQL diagnostic manager 8.5 to 8.6 may encounter a message (see image below) that requests closing the Java framework application to proceed with the upgrade process. This framework is in use by the SQLdm WebUI Service of our web console feature. Clicking Ignore allows the upgrade installation process to complete successfully. You can also stop the SQLdm WebUI Service manually and click Retry to proceed with the Upgrade Wizard.

General

- This release fixes an issue where scheduled SQLdm Repository grooming failed.
- This release fixes an issue that prevented data from being returned on a SQL 2012 instance when selecting the option Use Extended Events (SQL2012 + only) in the Wait Monitoring tab of the Monitored SQL Server Properties window.

Views

- An issue causing Data File Spaced Used and Log File Space Used values in the Overview -> Details view to vary from the real server values no longer occurs.
- This release fixes an issue where the number of sessions displayed in the Sessions -> Details view, after applying the Affecting Tempdb only filter, does not match the number of sessions in the Databases->Tempdb Summary view.
- An issue preventing disk drive statistics to return in the Disk view of the Resources tab no longer occurs.
- This release fixes an issue that caused the Query waits and Query History views to lag displaying data when the frequency of how often the console refreshes data was modified in the Tools -> Console Options window.
- SQLdm is designed to return the information that you want to view in the Signature Mode, Statement Mode, and Query History views through filters. However, the filter Exclude Currently Running Queries that excludes data returned with null values is not applicable to the Query History view and is now disabled by default.

Alerts

- An issue issue causing OS Disk Full (Percent) alerts to trigger for disk drives previously excluded from alerting in the Alert Filters tab of the Advanced Configuration window no longer occurs.
- This release fixes an issue impairing the Deadlock Sessions report with detailed deadlock information to display when right-clicking (Show Deadlock Details) or double-clicking over a deadlock alert in the Alerts view. Note that after upgrading to our latest version it is not possible to view deadlock details for alerts generated in SQLdm 8.5 since this issue prevented deadlock information from being stored.
- This release fixes an issue impairing the Blocking Sessions report with detailed blocking and blocked sessions’ information to display when right-clicking (Show Block Details) in the Alerts view, double-clicking over a Block report in the Blocking view, or clicking the Show Block Details link in the Details section of the Alert view. Note that after upgrading to our latest version it is not possible to view block details for alerts generated in SQLdm 8.5 since this issue prevented block information from being stored.
- An issue impeding new custom counters to be added in the Administration tab is solved.

Console

- This release fixes an issue present after upgrading to SQLdm 8.5 with custom counters that had customized categories. This issue caused a .NETFramework error and the SQLdm console to crash when clicking any monitored server.

Dashboard
An issue preventing the Lock Waits and Tempdb panels from displaying in a customized Dashboard view no longer occurs.

Reports

- An issue preventing the Top Queries report to deploy correctly into a Microsoft Reporting Services Server no longer occurs.
- This release fixes an issue where a custom report was deployed into the Microsoft Reporting Services Server instead of the selected report in SQL dm with a similar name.

8.5 New features

Full support for SQL Server 2014

SQLdm 8.5 fully supports monitoring of SQL Server 2014 RTM. Users who implement the SQL Server 2014 In-Memory OLTP Engine functionality should note that SQLdm does not support this feature at the moment.

Hyper-V support

SQLdm 8.5 has extended virtual support to Hyper-V, providing users insight into the performance of their virtual system more effectively. Users can track and monitor Hyper-V specific performance metrics for a complete view of the virtualized server and the physical host. For additional information, see [How SQLdm works with a virtual environment](#).

Tech Preview of the new SQLdm web console

In SQLdm 8.5 rapidly and easily identify problems. The SQLdm web console is designed to provide quick access to the overall status of the SQL Server environment to help diagnose any issues.

- Quickly view status of a single instance or instance groups.
- Access from any connected browser. No SQLdm Desktop Console required.
- Stand-alone web application with no need for IIS.
- Provide an easy to use web console to non-DBAs without giving access to entire management console so they can check on status of their SQL Server environment.

To provide feedback on the Tech Preview of the SQLdm web console, access the [Idera community forum](#).

The SQLdm REST Service requires a service account that is a local administrator. If the service account provided during installation for the SQLdm services is not a local administrator, you need to manually change the service account in Windows Service Control Manager.

Monitoring and alerting enhancements

Enhanced alert response usability

Ability to set up Alert Response that is triggered only after multiple user defined metric thresholds have been breached. For additional information, see [Configure automated responses to alerts](#).

Inclusion filters for Query monitors

Ability to do query monitoring on selected applications, databases, and SQL text. For additional information, see [Configure advanced query monitor options](#).

New filter in the Signature Mode and Statement Mode views

Ability to exclude data returned with null values through the Exclude Currently Running Queries filter in the Signature Mode and Statement Mode views.

Improved Maintenance Mode flexibility

Additional monthly option for scheduling maintenance mode on monitored SQL server instances. For details, see [Schedule Maintenance Mode](#).

Virtualization counters available for custom reports

Select both Hyper-V and VMware virtualization counters when creating a custom report. For additional information, see [Select counter type and counters](#).

Run one-click System Diagnostics

Ability to run system diagnostics and collect service and desktop client logs in the System Diagnostics window. See [System Diagnostics](#) for details.

Deadlock victim details now available in the Deadlock Sessions Report

The Deadlock Sessions Report of the Sessions>Blocking view includes deadlock victim information by default. Users no longer have to manually add this option through the Column Chooser dialog. For additional information see, [Analyze blocked sessions](#).
Configuration option added to control WMI timeouts

Configure the WMI timeout value in SQLdm when suitable. For instructions, see Configure OS metrics monitoring.

8.5 Fixed Issues

General

- An issue causing settings in the Monitored SQL Server Properties window to revert no longer occurs.
- This release fixes an issue in deadlock data collection that caused discrepancies in the UTCCollectionDateTime.
- An issue causing SQLdm to display an error on the monitored SQL Server 2014 instance when expanding the databases node, no longer occurs.
- Some users may experience Windows console crashes, applying the Microsoft update 2919355 solves this issue. For additional information on this solution, see the Microsoft knowledge base articles:
  - Out of memory when you load some image resources in a Windows application.
  - Platform update for Windows 7 SP1 and Windows Server 2008 R2 SP1.

Alerts

- An issue preventing cluster failover alerts to be triggered no longer occurs in SQLdm 8.5.
- An issue preventing SQLdm to update statistics for some monitored SQL servers that triggered a persistent alert ‘Unable To Monitor’ no longer occurs in version 8.5.

Reports

- This release fixes an issue impeding the VM Statistics report to display correctly.
- An issue preventing users from exporting the Blocking Sessions report in XML no longer occurs.
- An issue preventing the CPU Statistics report to display correctly when being deployed to the Microsoft Reporting Services no longer occurs.

8.0.1 New features

There are no new features in this release.

8.0.1 Fixed issues

General

- Users with large Repository databases no longer experience high CPU Usage by the sqlserv.exe process.
- An issue causing occasionally negative CPU Delta (ms) and Physical I/O Delta values in the Details view of the SQLdm Sessions tab, when parallel usage of CPU is enabled, no longer occurs.

Management Service Log

- An issue causing exceptions in the SQLdm Management Service logs when removing an availability group or replica from Availability Group Topology no longer occurs.
- The exceptions logged for ‘primary and foreign key duplication errors’ no longer occur.

AlwaysOn Availability Groups

- An issue causing deadlock errors after updates to the AlwaysOn Availability Group Topology tables in SQLdm no longer occurs.

Mobile

- Users who remain logged in the Idera Newsfeed account and upgrade to SQLdm 8.0.1 no longer experience console crashes.

8.0 New features

SQLdm now fully supports the AlwaysOn Availability Groups feature of SQL Server 2012

SQLdm 8.0 now allows DBAs to monitor their availability groups, availability replicas, and availability databases. Support for this feature comes with:

- An Availability Group Statistics report that allows you view the historical health of your availability groups, availability replicas, and availability databases.
- An Availability Group Topology report that allows you to view the current topology of your availability groups configuration.
- Monitoring of key metrics specific to the AlwaysOn Availability Groups feature.
- Queue Size and Transfer Rates charts.

For additional information on SQLdm and the AlwaysOn Availability Groups feature, see Monitor AlwaysOn Availability Groups.
**SQLdm now offers integration with Microsoft System Center Operations Manager (SCOM)**

SQLdm 8.0 now includes the SQLdm Management Pack (SQLdm MP) to enhance the SQL Server monitoring capabilities of Microsoft System Center Operations Manager (SCOM). Detailed alert and status information tracked by SQLdm provides SCOM with a wealth of knowledge not previously available including:

- Seamless integration between SQLdm and the SCOM health model and console.
- More robust overview of your server health and performance through the addition of SQLdm metrics to the current SCOM ability to use availability to determine SQL Agent health.
- Detailed record of state changes and events that might impact the availability and performance of your server.
- Instant access to the SQLdm Console from SCOM, providing access to vital insights not readily available using SCOM alone.

For additional information about using SQLdm with System Center Operations Manager, see [Integrate SQLdm with SCOM](#).

**Change Log Summary report**

The Change Log Summary report allows users to view a list of all actions and configuration changes performed in their SQLdm environment over a specified period of time. For additional information on this report, see [Change Log Summary](#).

**Improved collection of query and non-query activities**

SQLdm 8.0 now allows users to collect solely queries' performance data through the Query Monitor tab in the Monitored SQL Server Properties window. Configure collection of non-query activities such as deadlocks and autogrow events through an independent window, the Activity Monitor tab.

**Blocked Process report available in SQLdm**

SQLdm users are now able to view the captured Blocked Process report directly in their SQLdm console. In the Activity Monitor tab, users can configure the Blocked Process Threshold (in seconds) that results in the Blocking Process and Blocked Process details report. To access this report, users can go to Sessions>Blocking and double click under Block Reports. For more information on this feature, see [Set activity monitor options](#).

**Improved Maintenance Mode and Snooze Alerts features**

In the SQLdm 8.0 version, applying maintenance mode modifications to a large number of servers at once is possible through the Maintenance Mode command. Review [Schedule Maintenance Mode](#) for details. Also users that want to tackle several alerts, can mass snooze/resume alert generation through the Snooze Alerts/Resume Alerts commands. For additional information, see [Snooze Alerts](#).

**SQLdm console supports Operating Systems where FIPS compliance is required**

The SQLdm 8.0 console supports operating systems with enabled system cryptography that use FIPS (Federal Information Processing Standard) compliant algorithms. For more information about FIPS compliance, see the Microsoft document, [FIPS Compliance](#).

**SQL Server 2014 experimental support**

SQLdm 8.0 is SQL Server 2014 compatible. This version of SQLdm is not certified against newer builds of SQL Server and should not be used with these builds in a production environment. Idera provides experimental support while you use your installation in a testing environment to ensure the features you rely on most are working as or better than expected.

Users who implement the SQL Server 2014 In-Memory OLTP Engine functionality should note that SQLdm does not support this feature at the moment.

**8.0 Fixed Issues**

**General**

- An arithmetic overflow error that converts expression to data type integer in the Query Waits and Session Details views no longer occurs.
- An issue causing errors related to data retention or data visibility in the History Browser no longer occurs. Note that some views in SQLdm only collect data on demand in the repository after configuration.
- In SQLdm 8.0 users are now able to limit the amount of memory used by the desktop client.

**Servers**

This release fixes an issue that occurred while refreshing the Server Summary view.

**Databases**

An issue causing errors related to fragmentation collector timeouts or fragmentation statistics collection in the Tables & Indexes view no longer occurs.
Services

This release fixes an issue caused when the collector for the SQL Agent Jobs View could not interpret null values, resulting in job and job steps collection errors.

Alerts

- The transactions (Per Second) metric now displays correctly in the History Browser when user selects a historical snapshot.
- This release fixes an issue where after modifying the advanced configuration settings of the SQL Server Status alert, the Lowered to OK status still showed.

Resources

An issue causing an unknown value in the User/Schema column of the Procedure Cache view no longer occurs.

Reports

An issue preventing users from running the Alert History Report after deployment to Reporting Services no longer occurs.

7.5.4 New Features

General

Charts now properly manage real-time and scheduled collection data in the same time span

SQLdm charts provide data whether collected real time or as part of a scheduled collection event. If you pause or leave this view and then return during the same Console session, SQLdm displays the real-time data points as previously viewed, followed by a period of data points from any scheduled collections that occurred while you were away. For additional information about how SQLdm displays charts, see Charts.

Queries

Query Waits now located on Queries tab

The Query Waits feature is no longer located on the Resources tab but now is available on the Queries tab. For additional information about Query Waits, see View Query Waits.

Alerts

Manage how the SQLdm Console responds to alert notifications

SQLdm can now play a sound when an event triggers an alert to change states. You can also control how SQLdm displays the resulting alert notification in the Console. For additional information about Console notifications for alerts, see Configure notification settings.

Set SQLdm to notify you of only critical alerts

Use the Notifications section of the Console Options feature to set SQLdm to notify you for only critical alerts. You can select whether you want to view or even play a sound when a critical alert occurs. For additional information about Console notifications for alerts, see Configure notification settings.

Mobile

SQLdm Mobile & Idera Newsfeed now supports IIS 8

SQLdm Mobile & Idera Newsfeed now supports IIS 8 on Windows 2012 and Windows 8 environments. For additional information about SQLdm Mobile & Idera Newsfeed requirements, see SQLdm Mobile and Idera Newsfeed requirements.

7.5.4 Fixed Issues

General

- SQLdm now responds to the collection filtering set for disk drives to prevent the Collection service from gathering data on non-monitored disks.
- Users who upgraded to SQLdm 7.5.3, and who collect data using a direct WMI connection no longer experience an issue causing an instance to stop collecting snapshots.
- An issue causing errors when the Collection service attempted to collect Table Growth information no longer occurs.

Servers

- SQLdm no longer pauses data point creation in other real-time panels when the user accesses the Timeline view.
This release fixes an issue caused when the scheduled refresh did not successfully complete, resulting in missing History Browser snapshots.
- The SQL Server Plan Cache object no longer continues to grow with SQLdm queries even with the SQL Server Optimize for ad hoc workloads server configuration option selected.
- The SQLdm Powershell plug-in now properly puts monitored SQL Server instances into maintenance mode when selected.

**Alerts**

- The AutoGrow alert no longer displays the same previous and current data file size when viewing alert details in the Alerts view.
- SQLdm no longer dismisses certain alerts when SQLdm generates an Unable to Connect, Unable to Monitor, or Paused alerts. These alerts include SQL Server Agent job alerts, VM Host Server Change, VM Resource Configuration Change, Mirroring Server Role Change, and Cluster Failover alerts.
- SQLdm no longer displays an incorrect percentage in the Database Full (Percent) alert.
- SQLdm no longer delays generation of some Oldest Open Transaction alerts until the transaction is significantly older than the alert threshold. The alert now triggers immediately when the transaction crosses the threshold.
- An issue preventing some users from monitoring multiple instances and causing job completion error alerts no longer occurs.

**Reports**

- The Server Summary report now correctly displays statistics graphs and grids even when the results include no data for that interval.
- The Server Statistics report now consistently applies the UTC offset to both servers when using the report to compare two servers.

**Counters/Custom Counters**

- When WMI counters are unavailable due to a stall, SQLdm now continues to collect available counter data until it hits the connection limit and triggers an alert.
- VM custom counters now close the connection to the vCenter Server immediately after use, thereby avoiding a high number of connections to the server at one time.

7.5.3 New Features

There are no new features for SQLdm 7.5.3.

7.5.3 Fixed Issues

**General**

- Users who upgrade to SQLdm 7.5.2 no longer experience missed alerts.
- Users who upgrade from SQLdm 6.2 to a more recent version of SQLdm no longer experience issues upon subsequent upgrades. SQLdm displayed the message, "Query Monitor upgrade paused," in the log and caused excessive connections to the Repository.
- Users with large databases who upgrade to SQLdm 7.5.2 no longer experience Management Service blockage, causing slow inserts into the Repository.
- SQLdm now successfully installs and no longer stalls on the splash screen while displaying the message, "Connecting to Repository."
- The SQLdm Management Service no longer causes high CPU on the Repository.

7.5.2 New Features

There are no new features for SQLdm 7.5.2.

7.5.2 Fixed Issues

**General**

- When upgrading to SQLdm 7.5.x and a timeout occurs while upgrading the database statistics, SQLdm now properly re-submits the query and no longer indicates that it is running without re-starting the stopped upgrade. This issue caused some users to not see older database information.
- An issue causing the SQLdm console to take a long time to load or time out after upgrading to 7.5 and display the message, "Unable to connect: Timeout expired attempting to open server connection," no longer occurs.
- Users who upgrade to SQLdm 7.5 no longer experience an issue causing the transaction log file to grow dramatically.
- An issue with the stored procedure p_GetTagServersAsXML not closing or de-allocating a cursor no longer occurs as the stored procedure no longer uses a cursor.
- An issue with table statistics collection causing a communication problem between the Management Service and the Collection Service is resolved.
- SQLdm no longer displays an "Arithmetic overflow error" message when refreshing the Server Summary view This issue was caused by a timestamp containing a value greater than the SQL Server Max Integer value.

**Overview**

- An issue causing SQLdm to display some monitored SQL Server instances more than once in the Servers tree no longer occurs.
The CPU Dashboard panel no longer displays an occasional red “X” in the Call Rates chart when the panel is resized.

**Sessions**

- SQLdm no longer displays the messages, “Error executing Session List collector: Arithmetic overflow error converting expression to data type int,” or “Error processing Session List Collector: Object reference not set to an instance of an object,” when some users attempt to access pages in the Sessions view.

**Resources**

- SQLdm no longer displays the error message, “The given key was not present in the dictionary,” when a user attempts to view Server Waits.

**Alerts**

- SQLdm custom counters now properly trigger alert responses.
- Deadlock and Autogrow alerts no longer persist after the triggering condition is resolved.

**Reports**

- SQLdm no longer displays the message, “Item with the same key has already been added,” when some users attempt to run the Top tables by Growth or Top Tables by Fragmentation report.
- Users can now select the time of day in all SQLdm reports by using the Start Hours and End Hours fields.

### 7.5.1 New Features

There are no new features for SQLdm 7.5.1.

### 7.5.1 Fixed Issues

**General**

- An issue causing an exception error and possibly preventing some users from closing the SQLdm desktop client no longer occurs.

**Overview**

- The Memory, Network, and Virtualization panel chart selections are now properly saved when a user saves a Dashboard layout.
- Timeline users who double-click an alert to access the Alerts pane with the selected alert highlighted now receive the proper selected alert information in the Details pane of that view.

**Sessions**

- On a case-sensitive server, the Sessions Blocking graph now shows the correct deadlocks in the sessions.

**Resources**

- The Query Waits graphs right-click menu options View Text and View Query History are now operative.

**Logs**

- A user interface update fixed an issue causing some Microsoft Windows 7 users to see a cut-off Search window in the Log tab.

**Alerts**

- SQLdm and the Windows Event Log now both show accurate metrics for the OS Disk Free Space (Size) alert.

**Reports**

- Users who attempt to schedule email notifications for some reports no longer receive an error message at the end of the wizard.
- Generation of the Top Databases report is improved, preventing the report from suffering a time out because of the long load time.
- Disk Details report users no longer receive an exception message when attempting to run the report with the Drive Name field selection of All Drives.

### 7.5 New Features

There are no new features for SQLdm 7.5.
7.5 Fixed Issues

General

- The SQLdm installer no longer checks your Windows firewall service. If you have communication issues after installation, verify that ports 5166 and 5167 are open for access in Windows firewall and/or whichever firewall you use.
- SQLdm now properly aggregates data based on the number of days displayed in the Aggregate query data into daily records after X days field in the Grooming Options.
- Users monitoring disk activity on Windows 2008 and later operating systems no longer have to wait one second per disk for statistics gathering.
- Users no longer receive false “Unable to connect” alerts for monitored SQL Server instances known to be online. The server availability check now attempts one connection retry in order to avoid alerting in case of transient network issues.
- SQLdm now can send email notifications to Gmail accounts and other email accounts using SSL encryption.
- The About Idera SQL diagnostic manager dialog box no longer includes the System Info button.

Virtual Machines

- An issue in the SQL dm Management Service causing some VM statistics to not display for monitored virtual SQL Server instances no longer occurs.

Dashboard

- An issue causing the CPU Dashboard panel to display 0 when a greater value should appear no longer occurs. This issue was caused by SQLdm switching from one counter to another at certain refresh rates when calculating CPU values.

Server Overview

- SQLdm no longer attempts to delete a monitored SQL Server instance in some situations when the user pressed the Delete key while in the Server view.
- An issue causing some users to receive an exception error when attempting to view Server Details after quickly clicking through different panes of the SQLdm Management Console no longer occurs.

Sessions

- The Y-axis unit of time measurement now appears in the Sessions > Locks Wait Time chart as “ms” (milliseconds).

Resources

- Resources > Query Waits times are now in chronological order.
- Users no longer receive the error message, “The given key was not present in the dictionary,” when attempting to view the Resources > Server Waits view.

Databases

- SQLdm no longer displays restore activity on the original database instead of the destination database. Now the destination database includes any restore information.

Services

- The SQL Agent Jobs list now retains the open (+) and closed (-) settings after a refresh. This list also remains on the same row after a refresh and no longer jumps to the top of the list.
- An issue causing the job collector to fail for some users no longer occurs. This issue occurred when the user attempted to filter jobs or job categories with a string that included a single quote character (‘) in the title.

Alerts

- The Alerts pane now retains your column sort selection when you return to the pane after visiting another page, as when investigating an individual alert.
- SQLdm now sends an alert when a cluster failover occurs even when the server appears as offline in the preceding refresh of the SQLdm Collection Service. Previously, this situation would not trigger a cluster failover alert.
- The Session Tempdb Space Usage (MB) alert now triggers based on the Total User Space Used and no longer includes the Total Internal Space Used in the calculation. This update fixes an issue where users received an alert that shows a tempdb session using more space than the entire size of tempdb.
- The Cluster Failover alert no longer displays a timestamp of 1-1-1900 when an event triggers an alert.
- The Unsubscribed Transactions alert now displays the resulting time in hours, minutes, and seconds for easier comprehension.
- This release fixes an issue that caused some users to receive alerts stating that the data files are over 1000.00% full.
- Users no longer receive multiple email messages for the same alert.

Reports
Users no longer receive the error message, "An error occurred while retrieving data for this report. Additional information: Invalid Column name. (Microsoft SQL Server, Error: 207)," when attempting to run a custom report.

An issue causing the Top Queries report to appear empty when the user selected a Custom Range older than seven days no longer occurs.

**Custom Counters**

An issue causing some users to receive an error message when attempting to modify an existing custom counter no longer occurs.

### 7.2.2 New Features

There are no new features for SQLdm 7.2.2.

### 7.2.2 Fixed Issues

- SQLdm 7.2.2 now displays an accurate Physical I/O metric on the Blocking view of the Session tab.
- SQLdm 7.2.2 no longer displays XML document related error messages to users who upgrade to SQLdm 7.2 with a SQL 2005 repository. To learn more about the cause of this issue, see Microsoft Support Article 968914, "FIX: When you convert a TEXT column to a VARCHAR(MAX) column, an NTEXT column to a NVARCHAR(MAX) column, or an IMAGE column to a VARBINARY(MAX) column, the original NULL values in these columns may become empty strings in SQL Server 2005."
- An issue causing some users to receive an error message when attempting to view details for a monitored SQL Server instance no longer occurs. When this issue occurred, the results were replaced by a large red X.
- Both the Alerts and Custom Area views in SQLdm Mobile now display your custom counters regardless of whether you defined unique category names. By default, SQLdm assigns the category “Custom” to all custom counters.

### 7.2.1 New Features

**SQL Server 2012 RTM Support**

SQLdm 7.2.1 supports the use of SQL Server 2012 RTM version. Users who implement the SQL Server 2012 AlwaysOn functionality should note that while SQLdm seamlessly monitors servers with this feature enabled, it does not currently administer or diagnose issues with AlwaysOn.

### 7.2.1 Fixed Issues

- Users who have implemented SQLdm 7.2 with SQL Server 2012 RC0 in a non-production environment may experience issues with data display on the Tempdb Summary and Database Files views after upgrading to SQLdm 7.2.1. Upgrading to SQL Server 2012 RTM their test environment will remedy this issue.
- SQLdm 7.2.1 no longer displays error messages related to an “XML document” when users who upgraded to SQLdm 7.2 attempt to access the SQLdm Management Console.
- Users who implemented SQLdm 7.2 with SQL Server 2012 RC0 in a non-production environment now receive an accurate name and version number for SQL Server 2012 RTM when viewing the Compatibility Level for a monitored SQL Server 2012 RTM instance.

### 7 New Features

**Monitoring SQL Server virtual environments**

Get complete insight into the performance of your virtualized SQL Server databases. Monitor and track VMware-specific performance counters allowing you to get a complete performance picture of the SQL Server environment. Two VM-related reports provide performance trends and a summary of your virtualized environment. For additional information about configuring the new VM functionality, see Configure your virtual machine connections.

**SQL Server 2012 experimental support**

SQLdm 7.2 is SQL Server 2012 RC0 compatible. This version of SQLdm is not certified against newer builds of SQL Server and should not be used with these builds in a production environment. Idera provides experimental support while you use your installation in a testing environment to ensure the features you rely on most are working as or better than expected.

Users who implement the SQL Server 2012 AlwaysOn functionality should note that while SQLdm seamlessly monitors servers with this feature enabled, it does not currently administer or diagnose issues with AlwaysOn. For additional information about SQLdm and SQL Server 2012, see SQLdm requirements.

**SQLdm Mobile views**

SQLdm Mobile features a new server dashboard specifically for tablets and incorporated VMware monitoring data. For additional information about the new SQLdm Mobile features, see SQLdm Mobile Help.

**Reporting on Tempdb Statistics**
You can now run the Tempdb Statistics report to quickly identify and keep tabs on tempdb performance-related issues. For additional information about the new tempdb report, see Tempdb Statistics report.

**SQL Server Memory Usage Percent counter available for custom reports**

You can now select the SQL Server Memory Usage Percent counter when creating a custom report. For additional information about creating a custom report, see Custom reports.

**Updated alert notification functionality**

SQLdm now allows you to suppress system tray notifications when a monitored SQL Server instance returns to an OK status. Click Tools > Console Options. On the Notifications tab, check the Never show console alert notifications when the server status goes to OK checkbox, and then click OK. For additional information about the Management Console options, see Configure Console Options.

**CPU Statistics report now available in a custom range of hours**

The CPU Statistics report now includes an option for you to select a custom date range and view results in hours. To select this range, click Reports > CPU Statistics. Select the appropriate monitored SQL Server instance, and then select Custom Range from the Period drop-down list. Use the calendar tool to select the range, and then click OK. Select Hours from the Sample drop-down list to view your results in hours. For additional information about the CPU Statistics report, see CPU Statistics report.

**Input how long SQLdm waits before raising an alert for a long-running job**

Users can now specify a minimum job length for the SQL Server Agent Long Running Job (Percent) alert. This may be used to prevent alerting on short or trivial jobs. For additional information about configuring job exclusions, see Configure Job Exclusions.

### 7.2 Fixed Issues

- Users with a large number of databases no longer notice a performance issue caused by leaving the SQLdm Dashboard open for a long period of time.
- The Value Configuration window in SQLdm no longer displays blank column headings but includes alert state icons to easily identify the content. You can access the Value Configuration window by clicking Edit on the Alert Configuration window.
- The Login Time column of the Sessions Details page now uses the local time on the client server for the timestamp. You can access the Sessions Details page by clicking Sessions > Details while viewing a monitored SQL Server instance.
- SQLdm no longer displays an error message when a user attempts to view tempdb detail for a SQL Server 2000 monitored instance. Tempdb monitoring is supported for SQL Server 2005 and above monitored instances only.
- Table growth collection no longer fails for monitored SQL Server 2000 servers, allowing the Top Table Growth report to function correctly.
- SQLdm no longer displays an error message when a user attempts to use PowerShell cmdlets to configure Query Monitor.
- The CLR Enabled alert now properly functions after triggering an Informational alert.
- The Export to Excel function on the Tempdb Sessions view no longer fails for users who had cells in the Last Command column containing more that 32,767 characters.
- The Database Statistics report results format is updated and no longer causes any issues in viewing accurate results for users with large databases.
- SQLdm no longer displays and error message when a user attempts to schedule an email delivery of the Top Tables by Fragmentation report.
- The Repository now properly saves Query Monitor data while the grooming job runs.
- A failure in the SQLdm replication monitoring no longer prevents otherwise successful scheduled refreshes from saving to the repository.
- The Table Fragmentation (Percent) alert now changes to gray shading if the alert is raised for more than 12 hours.
- Extremely long queries no longer cause the Query Monitor to generate OutOfMemory exceptions on the Collection Service. These exceptions could previously cause a service crash.
- Users no longer experience an error during installation that caused an error when attempting to validate the user account used to perform the install.
- Performance updates including query aggregation and grooming changes improve SQLdm data collection and processing efficiency.
- SQL Server authentication no longer occasionally fails for users attempting to use a password containing the quotation mark (" special character.
- When you delete a database, the system is groomed to removes all references to that database. Now the feature also grooms the deleted database from the [SQLServerDatabaseNames] table.

### 7.1.1 New Features

There are no new features for SQLdm 7.1.1.

### 7.1.1 Fixed Issues

- Improvements now allow SQLdm to return data for the Session Details page more quickly and avoid a timeout due to slow or no response.
**New SQL Server dashboard**

SQLdm now features a completely redesigned dashboard, which dramatically expands the display of your performance data and simplifies the view of the overall health of a SQL Server instance. For additional information about the new dashboard, see the SQL Server performance overview.

**New Tempdb monitoring**

You can now quickly identify and resolve tempdb performance-related issues with the new tempdb monitoring and diagnostic views. For additional information about the new tempdb monitoring functionality, see tempdb status summary.

**New SQLdm Mobile views**

The following new views in SQLdm Mobile make diagnosing performance issues even easier while you're on the go:

- Blocking Chain View
- Sessions > Top Wait Customer View
- Tempdb Monitoring View

For additional information about SQLdm Mobile, see SQLdm Mobile Help.

**View and copy SQL Agent Job messages from the Job History list**

SQLdm now allows users to view SQL Agent Job messages from the Job History list. Right-click the appropriate job, and then select View Message. A copy feature allows you to copy the existing message and then paste it into a different application. For more information about SQL Agent jobs, see how to Monitor SQL Agent Jobs.

**7.1 Fixed Issues**

- SQLdm Repository timeouts no longer cause the SQLdm services to stop for some users.
- This release of SQLdm fixes an issue that caused SQLdm to display an error message when some users attempted to configure alerts.
- SQLdm performance improvements decrease the amount of time spent waiting for page loading and processes to complete.
- The CLR Enabled alert now properly retains settings when the user changes the alert state to Informational.
- SQLdm no longer experiences an issue when a user adds servers to a custom view.
- Improved collection functionality corrects an issue causing some users to see an incorrect SQL CPU Usage statistic.
- Refreshing the SQL Agent Jobs view no longer deselects all selected rows.
- The SQLdm Service Availability chart on the Services tab no longer incorrectly displays, "Unable to monitor" when the server is available.
- SQLdm no longer displays an error message when users who upgrade to SQLdm 7.0 attempt to edit their custom counters.
- SQLdm no longer displays the error message, "An INSERT EXEC statement cannot be nested," for some users who attempt to run a report when SQLdm Application Security is enabled.
- SQLdm users monitoring SQL Server instances by IP address instead of name no longer receive the error message, "The server you are trying to access is not monitored by Idera SQL diagnostic manager," when attempting to view history for a mirrored database.
- Users who have clustered servers no longer receive the error message, "Error interpreting Services Collector: Invalid length parameter passed to the substring function" when attempting to use the Show Real Time View function.

**7.0 New Features**

**New SQLdm Mobile interface**

SQLdm now offers a Web application (SQLdm Mobile) 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.

**New Alert Templates**

SQLdm now offers alert templates, which allow a user to configure generic alert settings to apply to monitored SQL Servers or groups of servers all at once. You can create and assign alert templates to any of your monitored SQL server instances.

**New Informational Alerts**

Informational alerts allow you to set a threshold that 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.

**Improved Alert Response Usability**

Alert Response rules, which configure how SQLdm responds when an event triggers an alert, are easier to create and manage due to an updated interface and process. Users can now apply an alert response rule to all metrics quickly and easily.
**Improved Notification Times for Unresponsive SQL Server Instances**

SQLdm can now alert you more quickly when one of your monitored SQL Server instances becomes unresponsive for any reason. Although the default is set to 30 seconds, you can adjust the interval at which the Collection Service checks whether your SQL Server instances are “up” or available. If a connection is not confirmed within the specified time, SQLdm creates an Unable to Connect alert.

**Improved Metrics Collection**

The Collection Service now:

- Uses a different SQL Server metric to determine the CPU Busy Value for instances running SQL Server 2005 or later. This change prevents timeout errors during the metric collection process.
- Uses separate processes to collect table growth and fragmentation metrics. This approach ensures quiet time data collection does not delay alerts about other metric thresholds.
- Is able to efficiently collect error log metrics for SQL logs that are large or infrequently cycled.

**Improved Repository Performance**

The SQLdm Repository database will perform more efficiently due to the following enhancements:

- The Repository grooming job now:
  - Calls the index maintenance in a separate step. Using SQL Server Management Studio, you can now disable this step and then later run your own index maintenance script after the grooming job completes.
  - Better handles I/O and CPU demands when processing data.
  - Several stored procedures used by SQLdm Repository were updated.

**New Reports**

SQLdm now offers the following new reports:

- The Metric Thresholds report is for users who want to view a list of metric thresholds for a monitored SQL Server instance. The report includes the metric name, description, and threshold for their informational, warning, and critical alerts.
- The Disk Details report lists key disk metrics for a specific SQL Server instance. This report offers Disk Time Per Read, Disk time Per Transfer, Disk Time Per Write, Disk Reads Per Second, Disk Transfers Per Second, and Disk Writes Per Second.

**7.0 Fixed Issues**

**Alerting Updates**

SQLdm now offers the following updates to alerting:

- SQLdm now accesses the correct database when a user double-clicks an associated Fragmentation alert on the Active Alerts tab of the SQLdm Today page. When you use this feature, SQLdm displays the affected database information on the Tables & Indexes pane of the Databases tab.
- Editing an alert response no longer causes SQLdm to display an error message when attempting to save your changes. This issue affected users who manually changed XML detail in the repository, and then attempted to save additional alert response changes using the SQLdm Console.
- SQLdm now properly captures customer error detail when Error: 0, Severity: 19, State: 0 occurs. This issue affected users using the SQL Server error log alert.
- The SQLdm scheduled alert refresh is no longer dependent on table statistics before completing a full refresh and possibly triggering alerts. Because table growth and fragmentation can be long-running processes, SQLdm was delayed in delivering alerts. These items were removed from the scheduled alert refresh for a more responsive alert system.
- SQLdm now raises an alert when a monitored SQL Server instance does not respond within the default 30 seconds. You can configure the response time in the Server Properties.
- SQLdm now checks to make sure that the WMI services are running before trying to connect to a monitored SQL Server instance. If SQLdm detects that previous attempts to connect are causing multiple connections, it will stop monitoring that server and triggers an alert. All real time views in SQLdm will display an error message explaining the issue. SQLdm automatically resumes monitoring once the number of connections drops below the default of 30 connections.

**Fragmentation Monitoring Updates**

SQLdm now offers the following updates to your table fragmentation monitoring:

- SQLdm now accurately collects table fragmentation data on tables within the entered size range. An issue causing SQLdm to collect table fragmentation data on tables less than the entered minimum size no longer occurs.
- Large tables no longer cause a block when collecting table fragmentation on a monitored SQL Server instance. This issue occurred when multiple very large tables were involved. Fragmentation data for populating the reorganization panel on all tables in a database is not collected. The reorganization query in SQLdm 6.1 and SQLdm 6.2 can timeout before table statistics are collected. For more information, refer to Idera Solution 1583.
**SQL Server Statistics Updates**

SQLdm now offers the following updates to the functionality that collects and analyzes your SQL Server statistics:

- The Sessions view now retains any custom filtering when viewing your Historical Snapshots. Custom filtering across all snapshots helps you quickly determine changes needing your attention.
- The History Browser on the Queries tab now allows you to scroll when you have large amounts of data.
- The SQL Server Physical I/O chart on the Resources tab now displays values in seconds instead of raw values. Displaying results in seconds provides you with more accurate real-time analysis.
- Users with mirrored databases no longer receive an error message that the mirror is not monitored when the database is actually monitored. This issue affected users viewing their monitored mirrored database information in the real time view on the Mirroring pane of the Databases tab.
- SQLdm now includes a new counter for collecting CPU stats. This new counter include a number of improvements over the previous counter, which occasionally suffered an error due to reaching maximum capacity cumulative CPU time.
- SQLdm no longer displays an error message when a user stops the Session Detail Trace and then refreshes the view. Previously, users received the error message, “Error starting Session Details collector System.InvalidOperationException: BeginExecuteReader: CommandText property has not been initialized.”
- Users working on diagnostics while troubleshooting can now add a configuration file entry to limit the procedure cache row count. The new key is maxRowCountProcedureCache, which has a default value of 10000.
- This release includes improvements to the repository grooming process, which fixes grooming of the ServerActivity table and resolves an issue causing occasional deadlocks during grooming.

**Reporting Updates**

SQLdm now offers the following updates to reporting:

- Unauthorized users can no longer access data for restricted SQL Server instances when viewing a report through SQL Server Reporting Services. This issue affected all users who restrict access to specific instances using Application Security features. This update improves data security and prevents confusion when a user unwittingly views data for the wrong instance.
- Users can now select up to 10 Windows, SQL Server, and/or custom counters when creating a custom report. These additional counters provide more flexibility and detail in your reports.
- If a custom counter contains special characters (<>&%), then your custom report generation fails with the error: Field names must be CLS-compliant identifiers. A CLS-compliant Custom Counter name uses the following rule: the first character of an identifier can be an uppercase or lowercase, titlecase, modifier, or other letter, or letter number. The subsequent characters can be any of the previously-mentioned characters, plus non-spacing or spacing combining marks, decimal numbers, connector punctuation, and formatting codes.

---

**Known issues**

Idera strives to ensure our products provide quality solutions for your SQL Server needs. If you need further assistance with any issue, please contact Support.

In this section you can find information on:

- Issues known in this version.
- Issues known in recent releases.
- Issues known in releases previous to SQLdm 7.5.

**Known issues in version 9.0**

**General**

**SQL diagnostic manager fails to update fields in the VM Configuration window after the virtualization host server is edited**

Users with SQL diagnostic manager version 8.5 and later may find that fields such as the Virtual Machine name, Type, and Computer Name in the VM Configuration window are not updated correctly after the virtualization host server is changed.

**Web Application Service**

**The Idera Dashboard's Web Application Service fails to restart after reboot**

Users with SQL diagnostic manager version 9.0 may find that the Idera Dashboard's Web Application Service does not automatically restart after a machine reboot.
Alerting

SQL diagnostic manager fails to display one alert response when multiple metric thresholds are breached

SQL diagnostic manager version 8.5 and later is designed to trigger an alert response only after multiple metric thresholds are breached when the “Where all metrics are in specified list” condition is selected. However, an issue in the “Send a Network Management (SNMP) Trap Message”, “Send an email to recipient”, and “Run the following SQL Script” actions of the Alert Response Window cause these actions to execute several times, resulting in multiple alerts.

The action “Run the following program” fails to execute when configuring automated responses to alerts

Users with SQL diagnostic manager version 8.0 and later may find that when creating an alert response with the associated action “Run the following program” in the Alert Response window, the specified program does not run once the condition is met.

SQL diagnostic manager returns PoweredOff alerts for the VM Power State metric when servers are not turned off

Some SQLdm users may view PoweredOff alerts when reviewing the generated “VM Power State” alerts for their servers linked to vCenter or Hyper-V host servers. These alerts display even though the servers are not turned off.

The actions “Write an event to the Windows event log” and “Send an email to recipient” in the Alert Response Window fail to deliver multiple alert statuses

Users with SQL diagnostic manager version 8.5 and later may find that when creating an alert response with multiple metrics as a condition and associated actions “Write an event to the Windows event log” and/or “Send an email to recipient” in the Alert Response window, the feature fails to deliver complete alert information and displays only one alert status when condition is met.

Previous known issues

Known issues in version 8.6

Alerting

SQL diagnostic manager fails to update statistics for some monitored SQL servers and triggers a persistent alert ‘Unable To Monitor’ for the same

Users with SQL diagnostic manager version 8.0.1 and later may find that some servers are not returning any statistics due to an issue in the Activity Monitor settings that causes the entire refresh of servers to fail.

SQL diagnostic manager fails to display one alert response when multiple metric thresholds are breached

SQL diagnostic manager version 8.5 and later is designed to trigger an alert response only after multiple metric thresholds are breached when the “Where all metrics are in specified list” condition is selected. However, an issue in the “Send a Network Management (SNMP) Trap Message”, “Send an email to recipient”, and “Run the following SQL Script” actions of the Alert Response Window cause these actions to execute several times, resulting in multiple alerts.

The action “Run the following program” fails to execute when configuring automated responses to alerts

Users with SQL diagnostic manager version 8.0 and later may find that when creating an alert response with the associated action “Run the following program” in the Alert Response window, the specified program does not run once the condition is met.

The actions “Write an event to the Windows event log” and “Send an email to recipient” in the Alert Response Window fail to deliver multiple alert statuses

Users with SQL diagnostic manager version 8.5 and later may find that when creating an alert response with multiple metrics as a condition and associated actions “Write an event to the Windows event log” and/or “Send an email to recipient” in the Alert Response window, the feature fails to deliver complete alert information and displays only one alert status when condition is met.

SQL diagnostic manager returns PoweredOff alerts for the VM Power State metric when servers are not turned off

Some SQLdm users may view PoweredOff alerts when reviewing the generated “VM Power State” alerts for their servers linked to vCenter or Hyper-V host servers. These alerts display even though the servers are not turned off.

SQL diagnostic manager fails to stop alert generation after the Availability Group Preferred Replica alert is un-checked

Users with SQL diagnostic manager version 8.6 that have the Availability Group Preferred Replica alert configured may find that when un-checking the alert in the Alert Configuration window to stop raising the alert, SQLdm continues to generate the alert which also displays in the Alerts View. Un-checking the configured database in the Configuration tab of the Alert Configuration window solves this issue.

General

SQL diagnostic manager fails to update fields in the VM Configuration window after the virtualization host server is edited

Users with SQL diagnostic manager version 8.5 and later may find that fields such as the Virtual Machine name, Type, and Computer Name in the VM Configuration window are not updated correctly after the virtualization host server is changed.

Known issues in version 8.5
**General**

The Query Monitor trace in SQL diagnostic manager does not return queries in progress

Users with SQL diagnostic manager version 8.0 and later may find that when using the Query Monitor functionality, the queries returned and displayed do not include queries in progress.

**SQL diagnostic manager fails to update fields in the VM Configuration window after the virtualization host server is edited**

Users with SQL diagnostic manager version 8.5 may find that fields such as the Virtual Machine name, Type, and Computer Name in the VM Configuration window are not updated correctly after the virtualization host server is changed.

**Alerting**

**SQLdm returns Blocking Session Wait Time alerts with session 0 and missing information when blocking-process node is empty**

Some users may experience Blocking Session Wait Time alerts with missing information and session 0 when the blocking-process node is empty in SQL Server. In SQLdm 8.01 and later versions, this issue may occur only if the trace that is collecting blocking data doesn’t return a SPID for a blocking process.

**SQL diagnostic manager fails to display one alert response when multiple metric thresholds are breached.**

SQL diagnostic manager version 8.5 and later is designed to trigger an alert response only after multiple metric thresholds are breached when the “Where all metrics are in specified list” condition is selected. However, an issue in the “Send a Network Management (SNMP) Trap Message”, “Send an email to recipient”, and “Run the following SQL Script” actions of the Alert Response Window cause these actions to execute several times, resulting in multiple alerts.

**The action “Run the following program” fails to execute when configuring automated responses to alerts.**

Users with SQL diagnostic manager version 8.0 and later may find that when creating an alert response with the associated action “Run the following program” in the Alert Response window, the specified program does not run once the condition is met.

**The actions “Write an event to the Windows event log” and “Send an email to recipient” in the Alert Response Window fail to deliver multiple alert statuses.**

Users with SQL diagnostic manager version 8.5 and later may find that when creating an alert response with multiple metrics as a condition and associated actions “Write an event to the Windows event log” and/or “Send an email to recipient” in the Alert Response window, the feature fails to deliver complete alert information and displays only one alert status when condition is met.

**SQL diagnostic manager returns PoweredOff alerts for the VM Power State metric when servers are not turned off.**

Some SQLdm users may view PoweredOff alerts when reviewing the generated “VM Power State” alerts for their servers linked to vCenter or Hyper-V host servers. These alerts display even though the servers are not turned off.

**Known issues in version 8.0**

**Installation and configuration issues**

The SQLdm Setup Wizard is interrupted after selecting the SQL server instance on which you want to install the SQLdm Repository

Some users may experience an interruption due to memory corruption in their SQLdm installation after selecting the SQL Server instance on which they want to install their SQLdm Repository. Note that the SQLdm installer does not complete on machines where memory corruption prevents connections to SQL Server. Furthermore, in these types of environments other applications such as SQL Server Management Studio also do not function.

**Negative values received from vCenter as a warning message do not impact VM data collection**

Some users may receive negative values from vCenter as a warning message. The message is logged as in the SQLdm Services. Note that this does not have any impact in terms of VM data collection and users are able to view VM and VM Host data.

**Console crashes when upgrading to SQLdm 8.0 while logged in the Idera Newsfeed account**

Users who remain logged in the Idera Newsfeed account and upgrade to SQLdm 8.0 may experience console crashes. Note that an upgrade from SQLdm 8.0 to 8.0.1 will solve this issue.

**Known issues in version 7.5**

**General**

SQLdm 7.5 no longer supports Microsoft Windows 2000 or the .Net 2.0 framework

SQLdm 7.5 does not support Windows 2000 or the .Net 2.0 framework. While SQLdm 7.2 and prior versions continue to operate with Windows 2000, SQLdm 7.5 and later require the .Net 4.0 Full framework to take advantage of the additional features. For additional
information about supported versions, see SQLdm requirements.

Server Overview

Timeline refresh causes a memory leak

SQLdm experiences a memory leak when you are viewing the Timeline and click Refresh. To free the memory, you must restart the SQLdm Management Console.

Queries

Queries view filter may not properly filter in some situations

In environments with large databases, the Queries view filter may not properly filter if you attempt to change filter properties during a refresh.

Resources

File Activity displays an unhelpful error message when WMI credentials are invalid

SQLdm displays the error message, “An error occurred while refreshing the active view,” when you attempt with access the Resources > File Activity view if your WMI credentials are invalid. If this message appears, verify your WMI access credentials and try again.

Installation and configuration considerations

SQLdm Repository computer must run Microsoft SQL Server 2005 or later

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

SQLdm Console only users cannot upgrade to the full version without uninstalling SQLdm

Users who have an installation of SQLdm Console only on a computer can use the upgrade to update their current version of the desktop client console, but cannot use the upgrade program to install the full version of SQLdm.

Registering a SQL Server instance that uses an Alias Name requires an alias on the host computer

When registering a SQL Server instance using the alias name, an alias must first be created on the computer that hosts the SQLdm Management and Collection Services. If the instance uses Named Pipes, Named Pipes must be enabled in the SQL Server network configuration of the aliased SQL Server instance.

Management Service does not start when the account permissions are incorrect

If the account you specified for the SQLdm services does not have System Administrator privileges on the SQL Server instance hosting the Repository database, the setup program cannot start the SQLdm Management Service. This issue only occurs during installation.

Some users may receive an error message when opening the SQLdm stating that there is no valid license available

To resolve this issue, use the Management Service Configuration wizard to specify credentials for a Windows account or SQL Server login that has System Administrator privileges on the SQL Server instance hosting the Repository database. You can start the Management Service Configuration wizard from the Programs menu.

SQLdm Reporting does not support Microsoft Reporting Services 2000

SQLdm only supports Microsoft Reporting Services 2005 and later.

SQLdm requires Predictive Analytics for baseline-related results

SQLdm 7.5 requires the Predictive Analytics Service in order to display baseline-related results. During installations and upgrades, SQLdm installs the Predictive Analytic service with the other included services. If Predictive Analytics is not detected or is disabled, SQLdm cannot generate baseline data for the requested report. The following reports require the Predictive Analysis Service to include baseline-related results:

- Baseline Statistics
- CPU Statistics
- Memory Statistics
- Session Statistics
- Disk Statistics
- Replication Statistics

Phase out SQLdm mobile manager

Idera is announcing end of life for SQL mobile manager (SQLmm). On October 1, 2012, all support for SQLmm will stop and the SQLmm trial site will be decommissioned. Also, any remaining installations of SQLmm 3.6 or earlier that use the Idera redirectors will no longer work.
To continue using a Web-based monitoring solution that is accessible from your smart phones or tablet devices, upgrade to SQLdm 7.0 or later, and deploy SQLdm Mobile. Idera discontinued distribution of SQL mobile manager in October 2011.

**SQLdm Mobile does not support Microsoft IIS Express**

SQLdm Mobile requires the standard version of the Microsoft IIS Web Server, version 7.0 or later. For more information, see the SQL dm Mobile requirements.

**SQLdm Mobile & Newsfeed installation cannot be changed or repaired using Add/Remove Programs**

To successfully change or repair your SQLdm Mobile & Newsfeed installation, uninstall the application and components, and then run the setup program to reinstall your deployment.

**SQLdm may not properly link to vCenter from some monitored virtual SQL Server instances**

The SQLdm VM Configuration window displays a list of your monitored virtual SQL Server instances, but the link between an instance and the associated vCenter may not work properly. To access this window, click Tools > VM Configuration.

**VM Host Server Change alert not always triggered when a VM moves from one host server to another**

Even if the circumstances warrant SQLdm to trigger the VM Host Server Change alert, users who have virtual machines running on an ESX host managed by vCenter 5.0 or later may not receive the alert when they move a VM from one host to another.

**Negative values received from vCenter as a warning message do not impact VM data collection**

Some users may receive negative values from vCenter as a warning message. The message is logged as in the SQLdm Services. Note that this does not have any impact in terms of VM data collection and users are able to view VM and VM Host data.

**Query Monitor disregards filter for SQL Server 2000 users**

Some SQL Server 2000 users may notice that SQLdm Query Monitor includes SQL diagnostic manager queries in the results even if the Exclude SQL diagnostic manager queries checkbox is checked in the Advanced Query Monitor Configuration window.

**SQLdm Mobile Server Dashboard charts display “Data not available” error**

This error indicates that SQLdm Mobile was not able to collect all the performance metrics tracked by these charts. This issue is more likely to occur with the I/O and Per Read/Write charts on the Disk dashboard. To correct this issue, use the Microsoft IIS Manager console to restart the SQLdm Mobile Web site (virtual directory).

**Start and Stop service actions not supported for virtual SQL Server instances**

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.

**Gaps between the real-time view and back-filled historical data can be seen on the CPU view**

This occurs only when the CPU view is accessed via the Server tree. To view the information, without any gaps, access the CPU view via the ribbon.

**Free disk space reported as zero on databases with special characters**

For servers and path names that include special characters such as % or #, SQLdm may incorrectly report the remaining disk space as zero.

**Known issues in versions previous to 7.5**

**General**

**SQLdm displays a refresh error when a user attempts to view data for an instance to which he does not have admin rights**

SQLdm displays a refresh error message when a user tries to view data for a monitored SQL Server instance to which the user does not have administrative rights. The message states that, "An error occurred while refreshing the active view."

**SQLdm displays an error when a user attempts to configure disk collection for an instance to which he does not have admin rights**

SQLdm displays an error message on the Server Properties Disk Drives window if a user’s collection service credentials do not have administrative access to the monitored SQL Server instance, and the user attempts to view the list of attached disk drives on that instance. Note that WMI credentials are not used to retrieve disk drive information on this page.

**Drives may not appear in the Resources Disk view when you switch to direct WMI collection**

Users who select to collect OS metrics using direct WMI collection may notice mounted drives appearing in some areas of SQLdm but not on the Resources Disk view. To refresh this list, you must restart the SQLdm Management Console.
Dashboard

Custom Counters dashboard panel does not return to default after deleting all counters

The Custom Counters dashboard panel includes a drop-down list to allow you to select from the available counters. If you delete all counters, the panel does not return to the default title and continues to display one of the counter names.

Resources

Disk Filter Columns does not work on a blank chart

The Filter Columns selection does not work in the Resources > Disk view if you receive a blank chart due to your WMI credentials being invalid.

Alerts

OS Disk Time (Percent) and OS Disk Time Per Disk (Percent) alerts may not trigger for some users

Users with Windows Server 2008 SP1 and SQL Server 2000 SP4, and who select direct WMI collection for their OS metrics may experience an issue preventing OS Disk Time (Percent) and OS Disk Time Per Disk (Percent) alerts from triggering. If this issue occurs, switch to using OLE automation.

Virtualization-related alerts occasionally raise more than once

In some circumstances, SQLdm may repeatedly raise virtualization-related alerts when there is no change in the status.

Custom Counters

SQLdm does not automatically add unassociated custom counters to an alert template created from an instance

If you use alert templates and create a custom counter, SQLdm automatically adds the new counter to your existing alert templates. However, if you create a custom counter not associated with an instance, and then create an alert template based on the alert configuration of one of your monitored instances, SQLdm does not add the new counter to your alert template nor can you manually add the counter.

To avoid this issue, add the missing custom counter to any SQL Server instance, create an alert template from that instance, and then clear any undesired custom counters from that instance.

SQLdm Mobile

SQLdm Mobile My Servers view filter does not work

The filter on this view will consistently display the message, “You have no favored or followed servers or the servers list has not been cached yet,” regardless of the filter option you choose. This issue occurs even when you are following servers or have marked servers as favorites.

Recommended Idera Solutions

Idera strives to ensure our products provide quality solutions for your database needs. The following Idera Solutions have been recently added to the knowledge base at the Customer Service Portal.

<table>
<thead>
<tr>
<th>Name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>4337</td>
<td>SQLdm alert, &quot;Session 0 has been blocking other sessions...&quot;</td>
</tr>
<tr>
<td>4287</td>
<td>Disk Full Percent fires at or near 100% for a disk that still has plenty of space.</td>
</tr>
<tr>
<td>4281</td>
<td>A SQL diagnostic manager with wait type MSQL XP uses constant CPU resources and never finishes on a monitored server.</td>
</tr>
<tr>
<td>4088</td>
<td>Disk Busy per Disk counter reports 100% when the disk is known to be idle.</td>
</tr>
<tr>
<td>3511</td>
<td>SQLdm always shows “The SQLdm Collection Service has too many outstanding connections” in the overview pane for a specific server, and SPIDs stall on the audited server.</td>
</tr>
</tbody>
</table>
When Upgrading SQL diagnostic manager, you receive the error message "The repository is an invalid version."

CPU Used, Memory Paging and Disk Queue Length show question marks in the Summary view.

What alerting Event Codes are used in SQLdm 5.0 or later?

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

Get started

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.

---

**Upgrade to this release**

Upgrading SQLdm to the most recent version allows you to take advantage of the new features available. An up-to-date list of new features and fixed issues is available in the SQLdm Release Notes.

To upgrade from SQLdm 4.x (or from earlier versions) to SQLdm 9.0, you first must upgrade to SQLdm 5.x. For assistance with your upgrade, contact Idera Technical Support.

---

**Upgrade from SQLdm 5.x or higher to SQLdm 9.0 checklist**

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

<table>
<thead>
<tr>
<th></th>
<th>Follow these steps ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>Back up your SQLdm Repository prior to upgrading.</td>
</tr>
<tr>
<td></td>
<td>Make sure that your environment meets the product requirements for the newer version of SQLdm.</td>
</tr>
<tr>
<td></td>
<td>Verify that no SQLdm desktop clients are running.</td>
</tr>
<tr>
<td></td>
<td>Upgrade your environment to the latest version of SQLdm on the computer where the SQLdm services reside. Note that SQLdm 9.0 integrates with the Idera Dashboard.</td>
</tr>
<tr>
<td></td>
<td>Test your new SQLdm deployment and setup.</td>
</tr>
<tr>
<td></td>
<td>Upgrade your Idera Newsfeed and SQLdm Mobile installations or deploy a new Idera Newsfeed and SQLdm Mobile installation.</td>
</tr>
</tbody>
</table>

---

**Upgrade to SQLdm 9.0**

Customers upgrading from previous SQLdm versions to SQLdm 9.0 need a new license key or it is not possible to continue with the upgrade process. Users can access the Idera Customer Portal at the moment of upgrade from the license management window in SQLdm (Help > Manage Licenses) or directly at https://idera.secure.force.com. Information necessary to generate a new license key is provided in the license management screen or the portal.
Upgrade SQLdm in a clustered environment

Use the following steps to upgrade your SQLdm installation in a clustered environment.

SQLdm 9.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database for all installations.

Upgrade in a Microsoft Windows Server 2003 Clustered Environment

To upgrade SQLdm in a clustered environment:

1. Log on to the cluster node using the Administrator account.
2. In Cluster Administrator, select the resource group that contains the SQLdm services.
3. Take the SQLdm Management and Collection services off line.

Perform the following steps 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 start, and then restart the services.

Once the previous steps are complete, perform the following steps on each cluster node:

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

Upgrade in a Microsoft Windows Server 2008 Clustered Environment

To upgrade SQLdm in a clustered environment:

1. Log on to the cluster node using the Administrator account.
2. In the Failover Cluster Management tool, select the resource group that contains the SQLdm services.
3. Take the SQLdm Management and Collection services offline.

Perform the following steps 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 start, and then restart the services.

Once the previous steps are complete, perform the following steps on each cluster node:

1. Bring the SQLdm Management Service on line.
2. Use the Management Service Configuration wizard to force the services to re-register on the active node.

Perform a distributed upgrade

Use the following procedures to upgrade a distributed SQLdm 9.0 environment.

Upgrade your SQLdm Repository and/or SQLdm Services

Install the SQLdm Repository database on a SQL Server instance that meets the product requirements. You do not need to install your Repository in the same domain as the SQLdm Console, but a two-way trust between the target domains must exist. You can install the SQLdm services on the Repository computer (recommended) or a different computer.

SQLdm 9.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database for all installations.

To upgrade the SQLdm Repository and/or Services:
1. Use an administrator account to log on to the computer on which you want to upgrade the SQLdm Repository database and/or Services.
2. Run **Setup.exe** in the root of the installation kit.
3. On the Idera SQL diagnostic manager Quick Start window, click **SQL diagnostic manager** under the Install heading.
4. Read the Upgrade Wizard Welcome window, and then click **Next**.
5. Upgrade your Repository by verifying the authentication method and connection credentials, and then click **Next**. This account must belong to the sysadmin role on the Repository database and each instance you plan to monitor.
6. Click **Upgrade** to begin your upgrade.
7. Click **Finish** to exit the upgrade program.

**Upgrade your SQLdm Console**

You must have a full installation of SQLdm already running in your environment before attempting to upgrade the SQLdm desktop client console.

**To upgrade the SQLdm Console:**

1. Use an administrator account to log on to the computer on which you want to upgrade the SQLdm Console.
2. Navigate to the location where the SQLdm Console file resides in the distribution media.
3. **If you use an x64 operating system**, run **SQLDiagnosticManager-x64_c.exe**. **If you use an x86 OS**, run **SQLDiagnosticManager-c.exe**.
4. On the Welcome window of the setup program, click **Next**.
5. Click **Upgrade** to begin your upgrade.
6. Click **Finish** to exit the upgrade program.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

**Upgrade SQLdm console only**

SQLdm allows users to install a full version, which includes the SQLdm Repository, Management Service, and Management Console, or you can install the SQLdm desktop client console by itself when you already have a full version of SQLdm installed in your environment.

Users who have an installation of SQLdm Console only on a computer can use the upgrade to update their current version of the desktop client console, but cannot use the upgrade program to install the full version of SQLdm.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

**Test 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 identifies and resolves SQL Server performance problems before they happen. Learn more >>

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

**Welcome to SQLdm**

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. SQL diagnostic manager now integrates with the Idera Dashboard, a common technology framework, designed to support the Idera product suite. Users are able to obtain an overview of the status of their SQL Servers and hosted databases all in a consolidated view and navigate to individual product dashboards for details. The Idera Dashboard provides a central set of services for managing users, product registry, instance registry, aggregated alerts across Idera applications, a central web server, and tags for grouping instances. For additional information, see Navigate the Idera Dashboard.
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** identifies and resolves SQL Server performance problems before they happen. Learn more >>

---

### What is SQLdm?

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 the Idera Dashboard?

The Idera Dashboard is a common technology framework designed to support the entire Idera product suite. The Idera Dashboard allows users to get an overview of the status of their SQL Servers and hosted databases all in a consolidated view, while providing users the means to drill into individual product dashboards for details. The Idera Dashboard supports multiple copies of SQL diagnostic manager installations.

---

### What is the SQLdm web console?

The SQLdm web console is a web application designed to provide quick access to the overall status of your SQL Server environment to help diagnose any issues. The web console provides key information simply through dashboards and high level views to a large user base such as IT, helpdesk, and DBAs. You can access the web console from any connected browser and grant access as you deem necessary. To learn more about the SQLdm web console, visit Navigate the web console dashboard.

The SQLdm web console is integrated with the Idera Dashboard.

---

### What is SQLdm Mobile?

SQL diagnostic manager (SQLdm) 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.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

---

### 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.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

**Use the SQLdm interface**

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.

**Navigation pane**

The Navigation pane displays on the left side of the SQLdm Console as shown in the following image.

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 Tags section allows you to manage tags or key words that you can use to quickly find the appropriate monitored SQL Server instances, custom counters, and permissions. 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 links to configure alert templates, actions, and responses.

**Newsfeed**

If installed, the Idera Newsfeed provides collaboration regarding your most critical SQL Server issues. If the Idera Newsfeed is not installed, this page guides you to where you can get additional information regarding the Idera Newsfeed.
Reports

Allows you to select the report you want to generate. SQLdm provides reports for monitoring, analyzing, and planning how to best use the SQL Server instances within your environment.

Administration

Allows you to control access to SQLdm, add custom counters for monitoring, and keep track of key actions performed in your SQLdm environment.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more »

View SQLdm Today

The SQLdm Today page provides a current overview of all the monitored SQL Server instances in your environment and offers shortcuts to:

- your five most critical servers
- the five servers you recently monitored
- all active alerts in your SQL Server environment
- the most recent server status updates and coworker posts from the Idera Newsfeed

Access the SQLdm Today page

You can open the SQLdm Today page from the SQLdm Console menu by clicking the SQLdm Today button as shown in the following image.

Use SQLdm Today to view the general health of your 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.

Get the most from SQLdm
The Get Started tab of the SQLdm Today page provides links for you to access content in the SQL diagnostic 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.

**View your recent active alerts**

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.

If you are aware of an alert and do not 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. Select the appropriate number of minutes you want to alert to snooze, and then click OK.

**To snooze an alert on the Active Alerts tab:**

1. On the Active Alerts tab of the SQLdm Today page, right-click anywhere on the row containing the alert you no longer want to view.
2. Select Snooze Alert.
3. Type or scroll to select the number of minutes you want the alert to snooze, and then click OK.

**Use Idera Newsfeed to track your 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.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more >>]

| Idera Website | Products | Purchase | Support | Community | About Us | Resources | Legal |

**Refresh and pause views**

To let you control the information refresh on a particular view, the SQLdm Toolbar includes the Refresh and Pause buttons as shown in the following image.

You can edit the scheduled refresh of your views for more current information. SQLdm offers the following refresh controls:

**Refresh button**

The Refresh button tells SQLdm to immediately collect data and refresh the current Console view. Depending on the view, the refresh could take several seconds to complete. SQLdm displays the current refresh status in the lower-right corner of the SQLdm Console.

**Pause button**

The Pause button tells SQLdm to stop refreshing the console. If you click Pause during a refresh, the pause occurs after the refresh completes. To continue refreshing, click Pause again. Pausing the refresh allows you to maintain the currently-monitored state so you can see what is happening on your SQL Server instance without incoming information overwriting the data. This feature also
gives you time to export or print the displayed statistics before retrieving the next data set.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Filter views

On a number of views, SQLdm includes filtering options ranging from ribbons allowing you to select items shown on the view, to a Filter Settings window that provides a variety of options to customize how SQLdm presents your view.

Use the Filter options to customize your view

The Filter box contains the filtering options available for that view, meaning that the available options are not consistent across views because of the data displayed in each view. You can select the options you want displayed in the menu by using the Filter Settings window. Click the Filter icon on the ribbon to display the Filter Settings window.

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 you select the field.

Use the Group By feature to organize your view

The Group By feature allows you to group the information shown in your views. To group your information by a column, check the Group By Box option, and then drag one or more column headings into the provided area as shown in the following image.

Charts

Charts are populated from the point at which you navigate to the window with the chart. This allows you to view real-time charts for the data point selected. Real-time viewing is dependent on your refresh interval. Once populated, you can manipulate the type of chart you want to display and export the chart or copy it to the clipboard to paste into other programs.

In the Server Details view, click the Maximize button on the upper-right corner of a chart to display a large version of the chart. The Restore button returns the chart to the default view.
Understanding your chart data and refresh interval

SQLdm charts provide data whether collected in real time or as part of a scheduled collection event. When you open a view, SQLdm begins populating the chart with real-time data collected on the interval specified in the **Server View Refresh** field on the **Console Options window (Tools > Console Options)**.

If you pause or leave this view and then return during the same Console session, SQLdm displays the real-time data points as previously viewed, followed by a period of data points from any scheduled collections that occurred while you were away. The chart then returns to populating with new real-time data points, as shown in the following image.

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. In addition, the time entered in the **Keep data for the last** field is the amount of historical data that you can scroll back to in your charts.

Customize your charts

SQLdm contains charts that are populated by selecting the appropriate data items collected. Once populated, you can manipulate the type of chart you want 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 the chart and selecting **Toolbar** from the menu. The chart toolbar contains the following options:

- **Copy To Clipboard**
  - The **Copy To Clipboard** button allows you to copy the chart to your Windows clipboard in various formats.

- **Print**
  - The **Print** button allows you to print the chart.

- **Gallery**
  - The **Gallery** menu includes a number of options that allow you to change the type of chart used to represent your data.

- **Anti-Aliasing**
  - The **Anti-Aliasing** button smooths the lines between your data points to more easily view your data.

- **Palette Selector**
  - The **Palette Selector** menu includes a number of color choices for you to change color scheme of your chart.

- **3D/2D**
  - The **3D/2D** toggle lets you choose to view your results in three- or two-dimensional plots. Additional options become active based on your choice and allow you to rotate your view or cluster your results on the z-axis.

- **Axes Settings**
  - The **Axes Settings** menu lets you choose how to view each axis, whether to display labels, and additional properties for your view.

- **Point Labels**
  - The **Point Labels** toggle lets you choose to view the actual data at each point.

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

- **Legend Box**
  - The **Legend Box** toggle lets you choose to view a legend for your displayed data.
The **Zoom** button allows you to zoom in on a point in your chart.

**Properties**

The **Properties** button displays additional properties for further customization of your chart.

---

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

Monitor multiple SQL Server 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.

**Access My Views**

My Views is available at all times as part of the Navigation pane as shown in the following image.

![My Views Image](image)

**Use the group list**

SQLdm updates the data shown in the thumbnails view of the group list 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.

By default, the group list displays basic statistics for each of the SQL Server instances monitored by SQLdm. This information includes the following statistics for each instance:

**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.

**SQL Memory Usage/Allocated**
Refers to amount of memory in use by the monitored SQL Server instance.

**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.

**Customize the monitors in your group list**

SQLdm allows you to customize the monitors displayed in group lists so that you display only those key statistics and 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, SQLdm displays the individual chart of the data item you selected. To revert back, select **Summary** from the Thumbnail list.

**Use the Server Thumbnails, Details, or Configuration Properties window**

SQLdm allows you to monitor multiple instances using different views, such as thumbnails of mini-monitors, a list of instances and their important details, or a list of instances and their server properties. SQLdm defaults to the thumbnails view. To access a different view, click the appropriate button in the upper-right corner of the window as shown in the following image.

![Server Thumbnails, Details, or Configuration Properties window](image)

If any of the metrics are missing from this view, you may need to enable OS metrics collection.

These views list all the data collected at the last refresh for all your monitored SQL Server instances. The Details and **Server Configuration Properties** views can assist you with comparing the status of a specific metric or property between two or more of your SQL Server instances.

**Add a custom 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. *If you want to create a new view from the File menu*, click File > Create View. *If you want to create a new view from My Views*, right-click within the My Views section, and then select **Create View**.
2. Type a name for the view.
3. Select the monitored SQL Server instances you want to include in this view.
4. Click **Add** to move the selected instances from **Monitored Instances** to **View Instances**.
5. Click **Create** to create your new view.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Server Configuration Properties**

The Server Configuration Properties view provides the current configuration settings for each of your monitored SQL Server instances and allows
for some changes directly in this window rather than leaving this view to access the properties page. The grid layout allows you to quickly scroll down in the column for a specific setting, and verify that the setting is correct in all of your instances.

You can sort your results in ascending or descending order for each column by clicking the column header. Right-click a column header to select other column options, including grouping by the selected column, hiding a column, or opening the Group By box for additional options.

Access the Server Configuration Properties view

SQLdm provides two paths to access the Server Configuration Properties view. The first access path is by clicking All Servers, and then clicking the Show Server Configuration Properties icon on the far right. The second access path is by right-clicking All Servers, and then clicking View All Server Properties. By default, the All Servers view displays your monitored SQL Server instances in the thumbnail view.

Performing general tasks in this view

The Server Configuration Properties view provides the following actions by right-clicking a row, and then selecting the task you want to perform:

**Open**

Opens the Dashboard view for the selected monitored SQL Server instance.

**Refresh Alerts**

Gathers the alert status and refreshes the tree view information for the selected monitored SQL Server instance.

**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.

**Collapse All Groups**

Allows you to collapse the contents of a group of monitored SQL Server instances.

**Expand All Groups**

Allows you to expand the contents of a group of monitored SQL Server instances.

**Print**

Allows you to print the grid of your monitored SQL Server instances and their configured properties.

**Export To Excel**

Allows you to export the grid of your monitored SQL Server instances and their configured properties to Microsoft Excel.

**Properties**

Opens the Monitored SQL Server Properties window for the selected instance.

Server Configuration Properties view contents

When you select the Server Configuration Properties view, SQLdm displays a grid containing each of the settings for all of your monitored SQL Server instances. This information is updated each time the SQLdm Collection Service runs. Some cells allow you to make changes on this view while others are managed in other areas in SQLdm, which are accessible by double-clicking the cell you want to change. SQLdm displays the Change icon in the Changed? column for the row in which you made the change. Make sure you click Save Changes once you made the necessary change. Click Reset Changes to revert all unsaved changes you made on this view. The Change icon helps remind you to save your changes as shown in the following image.

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

---

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

---
The Server Configuration Properties view contains the following properties:

**Active Cluster Node**
- Displays the name of the active cluster node on this monitored SQL Server instance. For more information about this setting, see Cofigure cluster settings.

**Alert Refresh**
- Displays the interval (in hours, minutes, and seconds) between times when the service collects diagnostic data and raises the associated alerts. Use a value between 30 seconds and 30 minutes. Lower values result in a faster triggered alert, but also cause more frequent refreshes, which may increase monitoring overhead. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click Save Changes. For more information about this setting, see Set general server options.

**Authentication Mode**
- Displays the authentication mode used by the SQLdm Collection Service to collect diagnostic data from the monitored SQL Server instance. For more information about this setting, see Set general server options.

**Activity Monitor Enabled**
- Displays whether monitoring of non-query activities is enabled. The monitoring of non-query activities captures autogrow events, deadlocks, and blocks. For more information about this setting, see Set activity monitor options.

**Baseline Date Range**
- Displays the period of time over which the server baseline is calculated. For more information about this setting, see Set server baseline options.

**Baseline Time Period**
- Displays the date range over which the server baseline is calculated. For more information about this setting, see Set server baseline options.

**Changed?**
- Displays an edit icon to indicate that this row includes a change from the last time the data was saved. Click Save Changes to retain any changes made in this view. Click Reset Changes to revert to the most-recently saved server configuration properties.

**Check Server Accessibility**
- Displays the interval (in hours, minutes, and seconds) between times when the server availability is verified. Use a value between 30 seconds and 10 minutes. If a select 1 query cannot execute in this timeframe, then the server is considered unresponsive. Setting your server availability check to a very low value may result in false positive alerts. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click Save Changes. For more information about this setting, see Set general server options.

**Database Statistics Refresh**
- Displays the interval (in hours, minutes, and seconds) between times when the database space-related data is collected and associated alerts are raised. Use a value between one minute and 24 hours. Setting your database statistics refresh to a low value may result in false positive alerts. A low setting also causes refreshes to occur more often, which increases the monitoring overhead. In environments with a large number of databases whose sizes do not change rapidly, setting this refresh to a long interval can greatly reduce your monitoring footprint. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click Save Changes. For more information about this setting, see Set general server options.

**Deadlock Monitoring**
- Displays the current state of deadlock monitoring, which raises alerts for deadlocked sessions on your monitored SQL Server instance. Deadlock monitoring is supported on monitored servers running SQL Server 2005 or greater and is dependent on enabling the monitoring of non-query activities in the Activity Monitor window. For more information about this setting, see Set activity monitor options.
**Disk Collection Settings**

Displays whether the connected disk drives are discovered automatically. For more information about this setting, see [Customize disk statistics](#).

**Error Log Alerting Enabled**

Displays whether error log alerting is enabled, causing a specific error log collector to run. Error log alerting is considered enabled if either the SQL Server Agent Log or SQL Server Error Log alert is enabled. Note that these collectors may negatively affect performance. You can reduce the impact of error log reading by regularly cycling the server error logs. For more information about error log alerting, see [Configure alerts](#).

**Fragmentation Min. Table Size (KB)**

Displays the minimum size, in kilobytes, a table must meet before fragmentation statistics are collected. Use a value between 0 KB and 999999 KB. Fragmentation statistics are only gathered on tables with clustered indexes. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click [Save Changes](#). For more information about this setting, see [Edit table statistics options](#).

**InputBuffer Limiter**

Displays the limit of the executions performed by the DBCC Inputbuffer. This command retrieves the actual input command for the Session Details view, among others. Note that on busy servers, limiting this value can reduce monitoring impact. For more information about this setting, see [Set general server options](#).

**Job Alerting Enabled**

Displays whether job alerting is enabled, causing a specific job alerts collector to run. Job alerting is considered enabled if one of the following alerts is enabled:

- SQL Server Agent Job Completion
- SQL Server Agent Job Failure
- SQL Server Agent Long Running Job (Minutes)
- SQL Server Agent Long Running Job (Percent)

For more information about job alerting, see [Configure alerts](#).

**Last Fragmentation Collection**

Displays the date and time of the most recent, successful table fragmentation collection. This timestamp is local to the monitored SQL Server instance. For more information about this setting, see [Edit table statistics options](#).

**Last Table Growth Collection**

Displays the date and time of the most recent, successful table statistics collection. This timestamp is local to the monitored SQL Server instance. For more information about this setting, see [Edit table statistics options](#).

**Linked Custom Counters (Count)**

Displays the total number of custom counters monitored on this SQL Server instance. For more information about this setting, see [Link or tag custom counters](#).

**Linked Custom Counters (List)**

Displays the name for each custom counter monitored on this SQL Server instance. For more information about this setting, see [Link or tag custom counters](#).

**Maintenance Mode**

Displays the occasions when this server is in maintenance mode. Options include Never, Until further notice, Recurring every week at the specified time, and Occurring once at the specified time. For more information about this setting, see [Schedule maintenance mode](#).

**OS Metrics Collection**

Displays whether SQLdm uses OLE automation or a direct connection to Windows Management Interface (WMI) to collect operating system and disk statistics. Disabling this setting prevents the collection of certain metrics in SQLdm, but may lessen the monitoring overhead. For more information about this setting, see [Configure OS metrics](#).

**Preferred Cluster Node**

Displays the name of the preferred cluster node on this monitored SQL Server instance. You can modify this setting directly in this view. Click in the appropriate field, make your change, and then click [Save Changes](#). For more information about this setting, see [Configure cluster settings](#).

**Query Monitor Enabled**
Displays whether query monitoring is enabled. The Query Monitor collects poorly-performing queries such as SQL batches, statements, stored procedures, and triggers. Low thresholds for query monitoring may impact performance. For more information about this setting, see Set Query Monitor options.

Query Monitor Thresholds
Displays the various threshold types for query monitoring. Note that low thresholds for the query monitoring impacts performance. For more information about these settings, see Set Query Monitor options.

Query Waits
Displays the date and time when the query-level wait statistics are collected or whether they are collected indefinitely. By default, these statistics are available only when the Query Waits view is open. Collection of query waits is a performance-intensive operation. For more information about this setting, see Configure wait monitoring.

Query Waits Filters
Displays the filter configuration of the query-level wait statistics for this monitored SQL Server instance, including which applications, databases, and SQL text you want to exclude from the query wait statistics collection. Collection of query waits is a performance-intensive operation and excluding unnecessary data can reduce the monitoring footprint. For more information about this setting, see Configure advanced query wait options.

Replication Monitoring Disabled
Displays whether replication statistics collection is disabled. Replication monitoring can cause excess monitoring overhead in some environments and is not necessary for all servers. You can modify this setting directly in this view. Click in the appropriate field to make your change, and then click Save Changes. For more information about this setting, see Disable replication statistics collection.

Server Name
Displays the name of the monitored SQL Server instance.

Session History Browser Collection Disabled
Displays whether extended session data collection is disabled, including collection of session details, locks, and blocks. Collection of this data allows you to view past session detail, lock, and block information, but incurs extra monitoring overhead to provide this data. This data also requires extra space in the SQLdm Repository. You can modify this setting directly in this view. Click in the appropriate field to make your change, and then click Save Changes. For more information about this setting, see Set general server options.

SQL Edition
Displays the edition of SQL Server software used in this instance.

SQL Server ID
Displays the identification number of the monitored SQL Server as used internally by SQLdm. You cannot modify this value.

SQL Version
Displays the version of SQL Server software used in this instance.

Table Fragmentation Collection
Displays the days of the week and time of day when an attempt at table statistics collection occurs. For more information about this setting, see Edit table statistics options.

Table Growth Collection
Displays the days of the week and time of day when an attempt at table statistics collection occurs. For more information about this setting, see Edit table statistics options.

Table Statistics Excluded Databases (Count)
Displays the total number of databases excluded from table statistic collection. For more information about this setting, see Edit table statistics options.

Table Statistics Excluded Databases (List)
Displays the name of each database excluded from table statistic collection. For more information about this setting, see Edit table statistics options.

Virtualization Host Name
Displays the name of the host server on which this monitored SQL Server instance is running. This field applies only to SQL Server instances running on a virtual machine. For more information about virtualization configuration, see Configure virtualization.

VM Name
Displays the name of the virtual machine on which this monitored SQL Server instance is running. This field applies only to SQL Server instances running on a virtual machine. For more information about virtualization configuration, see **Configure virtualization**.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more > >](https://idera.secure.force.com/)

### Use the SQLdm web console

The SQLdm web console helps users identify and diagnose problems quickly in their SQL Server environment. The SQLdm web console provides key overview information simply through dashboards and high level views.

- Access the overall status of your SQL Server environment from any connected browser. **No SQLdm Desktop Console required.**
- Provide an easy to use web console without giving access to the entire management console.
- Quickly view status of a single instance or instance groups (tags) and drill down for details.
- View all active alerts in your SQL Server environment and access detailed information.
- Select different sub-views to diagnose performance issues faster.
- Monitor overall health of queries and drill down through advanced query views for details.
- View your top values for selected metrics.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more > >](https://idera.secure.force.com/)

### Find answers

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 (https://idera.secure.force.com/).

**View help topics**

SQLdm Help supports the following Web browser minimum requirements:

- Internet Explorer 8
- Mozilla Firefox 4
- Google Chrome 6

**Print help topics**

You can print a help topic by selecting the topic in the table of contents, and then clicking the **Print this topic** icon on the right side of the Help window.

Use the PDF if you want to print more than a single topic. To print, click the PDF tab, and then click the **Print file** button. In the Print dialog box, select the pages you want to print before clicking **Print**.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more > >](https://idera.secure.force.com/)

### Customer support portal

The Idera Customer Support Portal offers 24 x 7 worldwide support to all customers who hold a current maintenance contract with Idera and provides an easy and convenient way to:

- Submit or track technical support cases.
- Search the Idera Knowledge Base.
- Download the latest version of your Idera products.
- Access information about your support plan and licenses.
- Access Idera resources such as the community site, blog, product videos, and white papers.

**Access the Idera Customer Support Portal**

Idera SQL Customers click this link to enter the portal: [https://idera.secure.force.com/](https://idera.secure.force.com/)

For any questions regarding access to the Customer Support Portal, contact your Idera Sales representative.
About Idera

At Idera, we deliver a new generation of tools for managing, administering, and securing your Microsoft Windows Servers, including SQL Server, 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.

Contact 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 the Idera website.

Sales
713.523.4433
1.877.GO.IDERA (464.3372)
(only in the United States and Canada)

Support
713.533.5144
1.877.GO.IDERA (464.3372)
(only in the United States and Canada)

Website
www.idera.com

Legal notice

Idera, Inc. (“Idera”) makes information and products available on this web site, subject to the following terms and conditions. By accessing this web site, you agree to these terms and conditions. Idera reserves the right to change these terms and conditions, and the products, services, prices, and programs mentioned in this web site at any time, at its sole discretion, without notice. Idera reserves the right to seek all remedies available by law and in equity for any violation of these terms and conditions. THIS WEB SITE MAY INCLUDE TECHNICAL OR OTHER INACCURACIES. CHANGES ARE PERIODICALLY MADE TO THE INFORMATION HEREIN. HOWEVER, IDERA MAKES NO COMMITMENT TO UPDATE MATERIALS ON THIS WEB SITE.

Trademark

Azure SQL Database Backup, Hyper-V VHD Explorer, Linux Hot Copy, PowerShellPlus, PowerShell Scripts for SQL Server, SQL admin toolset, SQL Backup Status Reporter, SQL Business Intelligence Manager, SQL check, SQL comparison toolset, SQL compliance manager, SQLcompliance, SQLcm, SQL defrag manager, SQL diagnostic manager, SQLdm, SQL doctor, SQL Elements, SQL Enterprise Job Manager,
SQL Fragmentation Analyzer, SQL Integrity Check, SQL Job Manager, SQL mobile manager, SQL Permissions Extractor, SQLsafe, SQLsecure, SQLtool, SQL toolbox, SQL Traffic Accelerator, SQL virtual database, SQLvdb, SQL XEvent Profiler, virtual database, Idera, BBS Technologies and the Idera logo are trademarks or registered trademarks of Idera, 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. Elements of this web site are protected by trade dress or other laws and may not be imitated or reproduced in whole or in part.

Copyright

The information on this web site is protected by copyright. Except as specifically permitted, no portion of this web site may be distributed or reproduced by any means, or in any form, without Idera’s prior written consent.

Use of the Software

The software and accompanying documentation available to download from this web site are the copyrighted work of Idera. Use of the software is governed by the terms of the License Agreement, which accompanies such software. If no license accompanies the download, the terms of the license, which accompanied the original product being updated, will govern. You will not be able to use, download, or install any software unless you agree to the terms of such License Agreement.

Use of web site information

Except as otherwise indicated on this web site, you may view, print, copy, and distribute documents on this web site subject to the following terms and conditions:

1. The document may be used solely for informational, personal, non-commercial purposes;
2. Any copy of the document or portion thereof must include all copyright and proprietary notices in the same form and manner as on the original;
3. The document may not be modified in any way; and
4. Idera reserves the right to revoke such authorization at any time, and any such use shall be discontinued immediately upon notice from Idera.

Documents specified above do not include logos, graphics, sounds or images on this web site or layout or design of this web site, which may be reproduced or distributed only when expressly permitted by Idera.

Warranties and Disclaimers; Liability Limitations

EXCEPT AS EXPRESSLY PROVIDED OTHERWISE IN A WRITTEN AGREEMENT BETWEEN YOU AND IDERA, ALL INFORMATION AND SOFTWARE ON THIS WEB SITE ARE PROVIDED “AS IS” WITHOUT WARRANTY OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

IDERA ASSUMES NO RESPONSIBILITY FOR ERRORS OR OMISSIONS IN THE INFORMATION OR SOFTWARE OR OTHER DOCUMENTS WHICH ARE REFERENCED BY OR LINKED TO THIS WEB SITE.

IN NO EVENT SHALL IDERA BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER INCLUDING WITHOUT LIMITATION, THOSE RESULTING FROM: (1) RELIANCE ON THE MATERIALS PRESENTED, (2) COSTS OF REPLACEMENT GOODS, (3) LOSS OF USE, DATA OR PROFITS, (4) DELAYS OR BUSINESS INTERRUPTIONS, (5) AND ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF INFORMATION) WHETHER OR NOT IDERA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Submissions

With the exception of credit card numbers for the purchase of products and services, Idera does not want to receive confidential or proprietary information through its web site.

Any information sent to Idera, with the exception of credit card numbers, will be deemed NOT CONFIDENTIAL. You grant Idera an unrestricted, irrevocable license to display, use, modify, perform, reproduce, transmit, and distribute any information you send Idera, for any and all commercial and non-commercial purposes.

You also agree that Idera is free to use any ideas, concepts, or techniques that you send Idera for any purpose, including, but not limited to, developing, manufacturing, and marketing products that incorporate such ideas, concepts, or techniques.

Idera may, but is not obligated to, review or monitor areas on its web site where users may transmit or post communications, including bulletin boards, chat rooms, and user forums. Idera is not responsible for the accuracy of any information, data, opinions, advice, or statements transmitted or posted on bulletin boards, chat rooms, and user forums.

You are prohibited from posting or transmitting to or from this web site any libelous, obscene, defamatory, pornographic, or other materials that would violate any laws. However, if such communications do occur, Idera will have no liability related to the content of any such communications.

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

You agree that all matters relating to your access to, or use of, this web site and these terms and conditions shall be governed by the laws of the state of Texas. You agree and hereby irrevocably submit to the exclusive personal jurisdiction and venue of the state courts of Texas located in Harris County, Texas, and the United States District Court for the Southern District of Texas, with respect to such matters.

Idera makes no representation that information on this web site are appropriate or available for use in all countries, and prohibits accessing materials from territories where contents are illegal. Those who access this site do so on their own initiative and are responsible for compliance with all applicable laws.

Export Control Laws

Certain Idera products, including software, documentation, services, and related technical data, available on the Idera and other web sites are subject to export controls administered by the United States (including, but not limited to, the U.S. Department of Commerce Export Administration Regulations (“EAR”)) and other countries including, controls for re-export under European Union, the Singapore Strategic Goods Control Act, and the import regulations of other countries. Diversion contrary to U.S. or other applicable law of any Idera product or service is prohibited. Export, re-export or import of products and services may require action on your behalf prior to purchase and it is your responsibility to comply with all applicable international, national, state, regional and local laws, and regulations, including any import and use restrictions. Idera products and services are currently prohibited for export or re-export to Cuba, Iran, North Korea, Sudan, Syria, or to any country then subject to U.S. trade sanctions. Idera products and services are prohibited for export or re-export to any person or entity named on the U.S. Department of Commerce Denied Persons, Entities and Unverified Lists, the U.S. Department of State’s Debarred List, or the U.S. Department of Treasury’s lists of Specially Designated Nationals, Specially Designated Narcotics Traffickers, or Specially Designated Terrorists. Idera products and services are prohibited from use with chemical or biological weapons, sensitive nuclear end-users, or missiles, drones or space launch vehicles capable of delivering such weapons. By downloading or using any product from this web site, or purchasing any service, you are acknowledging that you have read and understood this notice and agree to comply with all applicable export control laws. You are also representing that you are not under the control of, located in, or a resident or national of any prohibited country, and are not a prohibited person or entity. This notice is not intended to be a comprehensive summary of the export laws that govern the products and services. It is your responsibility to consult with a legal adviser to ensure compliance with applicable laws.

United States Government Rights

All Idera products and publications are commercial in nature. The software, publications, and software documentation available on this web site are "Commercial Items", as that term is defined in 48 C.F.R.§2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation", as such terms are defined in 48 C.F.R. §252.227-7014(a)(5) and 48 C.F.R. §252.227-7014(a)(1), and used in 48 C.F.R. ?12.212 and 48 C.F.R. 227.7202, as applicable. Pursuant to 48 C.F.R. §12.212, 48 C.F.R.§252.227-7015, 48 C.F.R. §227.7202 through 227.7202-4, 48 C.F.R. §52.227-19, and other relevant sections of the Code of Federal Regulations, as applicable, Idera's publications, commercial computer software, and commercial computer software documentation are distributed and licensed to United States Government end users with only those rights as granted to all other end users, according to the terms and conditions contained in the license agreements that accompany the products and software documentation, and the terms and conditions herein.

© 2003-2014 Idera, Inc., all rights reserved.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

Document conventions

<table>
<thead>
<tr>
<th>Element</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Titles</td>
<td>View the query monitor signature mode,</td>
</tr>
<tr>
<td>Views</td>
<td>The Dashboard view, the Server</td>
</tr>
<tr>
<td></td>
<td>Configuration Properties view, Details view,</td>
</tr>
<tr>
<td></td>
<td>Configuration view, Active Alerts view.</td>
</tr>
<tr>
<td>Objects</td>
<td>Deleted but preserved all collected data for</td>
</tr>
<tr>
<td></td>
<td>the monitored SQL Server.</td>
</tr>
<tr>
<td>States</td>
<td>For each scheduled refresh on a monitored</td>
</tr>
<tr>
<td></td>
<td>SQL Server instance where any metric</td>
</tr>
<tr>
<td></td>
<td>severity is Critical and able to query.</td>
</tr>
<tr>
<td>Level 1 titles</td>
<td>Heading 3 Style/Bold/Red font</td>
</tr>
<tr>
<td></td>
<td>Only the first letter capital unless there is a product name.</td>
</tr>
<tr>
<td><strong>Level 2 titles</strong></td>
<td><strong>Paragraph style/Bold/Red Font</strong></td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><strong>Level 3 titles</strong></td>
<td><strong>Paragraph style/Bold/black Font</strong></td>
</tr>
<tr>
<td><strong>Level 4 titles</strong></td>
<td><strong>Paragraph style/Bold/Italics</strong></td>
</tr>
<tr>
<td><strong>Path</strong></td>
<td><strong>Bold</strong></td>
</tr>
<tr>
<td><strong>Text</strong></td>
<td><strong>Paragraph style</strong></td>
</tr>
<tr>
<td><strong>Lists</strong></td>
<td>Use bullets to list several items Use numbers when there is a set of steps to be followed in a particular order.</td>
</tr>
<tr>
<td><strong>Field labels</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Buttons</strong></td>
<td>Use bold when referring to buttons that initiate commands</td>
</tr>
<tr>
<td><strong>Commands</strong></td>
<td>Use bold when referring to commands</td>
</tr>
<tr>
<td><strong>Warning macro</strong></td>
<td>Use warnings for important advice that should be taken into account by the user. If the user doesn’t follow this advice then some potential problems could appear.</td>
</tr>
<tr>
<td><strong>Note macro</strong></td>
<td>Use notes to give the user advice that will be useful under certain conditions (not always available or critical), this is less critical than a warning.</td>
</tr>
<tr>
<td><strong>Tips macro</strong></td>
<td>Use tips to give users advice that will help them make processes or tasks more easily.</td>
</tr>
<tr>
<td><strong>Panel macro</strong></td>
<td>Use the panel macro for examples. Border should be 'dashed'.</td>
</tr>
<tr>
<td><strong>Info</strong></td>
<td>Use Info to inform about some fact that is good for the user’s general knowledge, but that doesn’t represent any critical issue.</td>
</tr>
<tr>
<td><strong>Gallery Macro</strong></td>
<td>The gallery macro is used for a group of screenshots that don’t occupy a full size image in the page. This macro allows to view the images full size when you click on them and then you can continue to the next ones without having to return to the original page.</td>
</tr>
<tr>
<td><strong>Anchor Macro</strong></td>
<td>The Anchor macro allows you to link to a specific point in a page. You can use this macro in the same page or also with specific sections of other pages.</td>
</tr>
</tbody>
</table>
**Newtablink Macro**

You can use this macro to insert hyperlinks that will open on a different browser tab. This is used mostly for external hyperlinks. On the url field, type the link for the external site. On the alias field, type the words that will be clickable for the hyperlink.

Have not tried this macro yet, but I agree to its use.

**Page History**

The page history of each page should contain only the current version of the documentation. This option is available to all users and the Info dev should manually delete all past versions.

Sure can do.

**Attachments**

The attachments in each page (images) should be the same as the ones displayed in the page. Other attachments should be deleted and users should not be able to see them.

The info dev should make sure that any attachments uploaded to the page are actually inserted in the wiki page and if they won’t be used, they must be removed.

Ok this is a good point, have not done this earlier. Sometimes we have an image of an earlier build and update images later on. But yes, this sounds like a good practice.

---

**Installation and deployment**

You can install SQLdm and SQLdm Mobile in an environment of any size.

**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 or Idera Newsfeed deployment**

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

**Get ready for your SQLdm Console only deployment**

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

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

**Supported installation scenarios**

You can install and deploy SQLdm, SQLdm Mobile, and the Idera Newsfeed to meet your unique SQL Server environment needs. SQLdm also offers a SQLdm Console-only installation in environments that already have SQLdm fully installed. For additional information about this type of installation, see deploy SQLdm Console only.

The following installation scenario descriptions assume you are deploying SQLdm, SQLdm Mobile, and Idera Newsfeed in your environment.

In larger environments that contain more than one SQLdm installation, consider dedicating a SQLdm Mobile deployment per each SQLdm application.
SQLdm supports monitoring virtual SQL Server instances, such as those using VMWare or other virtualization software. For additional information about how SQLdm works with virtualization, see How SQLdm works with a virtual environment.

**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 and 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 and 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 to make sure 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 and 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 work groups.

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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Choose your deployment strategy

Before you begin installing SQLdm, it is important to consider your deployment choices.

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 each SQLdm component on a separate, dedicated computer. The Idera evaluation license is fully functional and monitors up to 15 SQL Server instances anywhere on your network for a period of 15 days.

**Production deployment**

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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

How SQLdm works with a virtual environment

Virtual environments, such as those provided by Hyper-V or VMware, allow you to run multiple instances of computers or “virtual machines” on a single, physical host server. These VMs create a virtual environment that enables IT shops to more fully utilize the hardware they already have, allowing administrators to maintain fewer physical servers which then consume less power, take up less real estate, and generate less heat.

While there are many advantages to using a virtual environment, there are some disadvantages. One is that a virtual machine currently is not as fast or efficient as a standalone physical machine. Administrators must closely monitor and manage their virtual environments because over-utilized hardware can quickly and severely impact the performance of their servers, which are constantly waiting for physical resources to become available to process their workloads.

SQLdm includes VM monitoring to allow administrators to get a glimpse of what is going on inside the virtual environment in which their SQL Servers are running, to make sure that their VMs and their host machines are running at peak efficiency. This topic is a general outline of how SQLdm works in a virtual environment and includes links to other areas where you can find additional VM-specific information.

Using the hypervisor to manage resources

On the physical server hosting a virtual environment, there is a process called the hypervisor or virtualization manager running in the background. The hypervisor manages the interactions between the virtual machines and the physical hardware, manages resources, and then distributes the physical resources as required by each of the VMs it is hosting.

An administrator can intentionally over commit resources on a physical host with the idea that it is unlikely that all of the VMs need to utilize all of the resources on the host at the same time. The hypervisor is also affected by this shift, and manages any occasion when all of the active VMs need more physical resources than is available from the host. Note that there are ramifications for the extra processing that is occurring when the hypervisor is taxed. This additional processing is the overhead for a virtual environment and consumes CPU, memory, and network resources.

How SQLdm collects and presents metrics involving a virtual environment

The guest operating system on a VM operates no differently than it does when running on a physical machine with full access to all of the resources at its disposal. But the machine hosting the VM operates differently because of the virtual environment. The physical resources of the host are shared with the other VMs on the host. Due to the additional overhead of virtualization, and the hypervisor managing contention for resources, the metrics that the guest OS reads as utilization of its resources are not entirely accurate as they do not take into account that virtualization overhead, nor does it measure that it is waiting on physical resources to process its workload.

The image below shows a typical virtualized environment with VMWare. In this environment, the ESX/ESXi hypervisor on each individual host server collects the performance statistics and reports the data back to a central vCenter Server. SQLdm then collects that information from the vCenter Server in order to show those statistics along with the other SQL related performance information.
In virtualized environment with Hyper-V, the process of how SQLdm collects and presents metrics is slightly different from that of a VMware setup. In this environment, SQLdm collects all the required performance statistics from the Hyper-V host server and virtual machines using the WMI interface.
SQLdm displays virtual machine and host metrics as collected from the perspective of the physical host. In other words, these are the “true” physical metrics. Whereas, the metrics that SQLdm displays for the OS on a virtualized machine are what Microsoft Windows calculates as using without regard to any overhead. For example, SQLdm displays the OS CPU Usage at 35%, but metric displayed for the VM CPU Usage is 38%. This discrepancy results from the additional overhead within the hypervisor used to run the VM, and the VM is charged for the additional processing.

**Link a monitored virtual SQL Server instance to the vCenter Server or Hyper-V host server**

To link a monitored virtual SQL Server instance to the host server that manages your VMs, click **Tools > VM Configuration** to access the VM Configuration window. SQLdm displays the current relationships between your available virtualization host servers and your monitored SQL Server instances. For more information about linking a monitored virtual SQL Server instance to a vCenter Server or Hyper-V host server, see **Configure your virtual machine connections**.

**VM data collection**

The VM collector connects to your virtualization host server (vCenter Server or Hyper-V host server) during the scheduled refresh for each of your monitored virtual SQL Server instances. If the SQLdm Management Console is open to the Overview Dashboard or any Resources view for a specific instance, data is also collected based on the Server View Refresh setting in the Console Options page. In this situation, it is possible that the collector connects to the virtualization host server and virtual machines every few seconds.

You can adjust the scheduled refresh rate for a monitored virtual SQL Server instance by right-clicking the instance name, and then selecting **Properties**.

**Differences in statistics**

You may notice some difference in your statistics, such as SQLdm displaying a higher OS CPU usage than your VM CPU usage. While the hypervisor manages resources efficiently, it is possible that the demand for physical resources is greater than what is available. For example, a guest OS on a virtual machine submits a batch of work to a CPU. However, all of the physical CPUs are committed to other work, so there is no
physical CPU available on the host to process the work. In this situation, the VM waits for a CPU to become available to process its work. While it is waiting, the guest OS is unaware of the wait and assumes that it is simply taking longer for the CPU to process this batch of work, and the OS thinks it is using more CPU power than it is actually using.

Remember that SQLdm uses the additional overhead of the hypervisor to calculate the VM metrics. In this situation, the hypervisor knows the VM is just waiting for a CPU, so it does not charge the VM for CPU processing power while it is waiting. The result is that the guest OS reports that it is using the additional processing power, however the hypervisor simply reports what is actually used. In this situation, your CPU usage metrics that are reported by Windows are higher than what is reported by VMware or Hyper-V for that VM.

**Available metrics related to your VMs**

A monitored virtual SQL Server instance appears in the Servers pane in the same way as a monitored physical SQL Server instances. Click the appropriate virtual instance, and SQLdm displays the same dashboard panels as for a monitored physical SQL Server instance, but also include specific metrics for the VM and host server.

The following dashboard panels include VM- or host-related information:

- **CPU panel.** Includes VM and Host usage metrics. The information available in this panel comes from the Server Details and Resources Summary views.
- **Memory panel.** Includes the VM Memory Usage and Host Memory Usage charts. The information available in this panel comes from the Server Details and Resources Summary views.
- **Network panel.** Includes the VM Network Usage Throughput and Host Network Usage Throughput charts. The information available in this panel comes from the Server Details and Resources Summary views.
- **VM Disk Usage panel.** Includes the amount of data read from and written to disk for this VM during the sample interval. The information available in this panel comes from the Resources Disk view.
- **Host Disk Usage panel.** Includes the amount of data read from and written to disk for this host server during the sample interval. The information available in this panel comes from the Resources Disk view.
- **Virtualization panel.** Includes the Virtual Memory Usage and Virtual Disk Usage charts plus the CPU Ready Wait Time gauge. The information available in this panel comes from the Server Details view.

Use the available virtualization alerts and VM-related reports to get the most information regarding your monitored virtual SQL Server instances and host servers.

**Available VM-related reports**

- Virtualization Statistics Report
- Virtualization Summary Report

**Available virtualization alerts**

- Host CPU Usage Alert
- Host Memory Swap Detected alert
- Host Memory Usage Alert
- Host Power State Alert
- VM CPU Ready Wait Time Alert
- VM CPU Usage Alert
- VM Host Server Change Alert
- VM Memory Swap Delay Detected Alert
- VM Memory Usage Alert
- VM Power State Alert
- VM Reclaimed Ballooned Memory Alert
- VM Resource Configuration Change Alert
- VM Resource Limits Detected Alert

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

**Deploy the Idera Dashboard and SQLdm**

Use the following links to prepare for your Idera Dashboard and SQLdm deployment:

- Check the supported installation scenarios
- Learn about the components and architecture
- Review system requirements for the Idera Dashboard and SQLdm
- View the installation instructions
- Log in the Idera Dashboard and SQLdm web console

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
Product components and architecture

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

- Learn about the Idera Dashboard’s components and architecture.
- Learn about the SQLdm’s components and architecture.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

Idera Dashboard components and architecture

The Idera Dashboard provides web and back end services, shared across multiple Idera products. To learn more about what the Idera Dashboard is and how it works, see Navigate the Idera Dashboard.

The Idera Dashboard consists of the following components:

- Web Application Service
- Core Service
- Core Repository

**Web Application Service**

The Web Application Service is a Windows service that wraps Apache Tomcat server. The Web Application Service serves up dashboard (Idera Dashboard) and SQLdm views and widgets that are displayed in the web console. The Web Application Service requires two ports:

- Standard HTTP port (by default 9290)
Core Service

The Core Service is a C# (.NET 4.0 Framework) based Windows service that hosts dashboard and SQLdm REST APIs that are used by the Web Application Service to configure and retrieve data. In addition, the Core Service handles product registration, security, configuration, product data, and event aggregation.

The Core Service uses two ports, one for REST API and the other for .NET remoting:

- Core Service REST API port (by default 9292)
- .NET remoting port (by default 9293)

Core Repository

The Core Repository is a database where all dashboard's configuration and aggregated data is stored. The Core Repository database is hosted on a SQL Server instance and is accessed by the Core Service to retrieve data.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

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

The Predictive Service: 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 Plug-in

When SQLdm is registered with the Idera Dashboard, the product plug-in module is deployed. The SQLdm plug-in consists of web views and widgets and a .NET based add-in module (SQLdm add-in). The web views and widgets are deployed in the Web Application Service of the Idera Dashboard, and the SQLdm add-in in the Core Service of the same.

The Web Application Service dynamically loads in SQLdm's views and widgets and makes them available to web console users. The views and widgets use the SQLdm add-in REST APIs to retrieve data. Likewise, the SQLdm add-in retries data from the product services and Repository.

SQLdm Web Console integrated to the Idera Dashboard

The Idera Dashboard and SQLdm web console are automatically installed upon upgrade or during installation of the 9.0 or higher versions.

To access the web console:
1. Open your selected Browser, make sure it is compatible with the SQLdm web console requirements.
2. Type the SQLdm product URL: http://<machinename>:<port> where <machinename> is the name of your host or machine, and <port> is the port specified during installation. The default URL is http://<localhost>:9290.
3. When the SQLdm web console launches on your browser, use your Windows user account <domain\user> with the respective password to log into the product.

To learn about the new components and architecture, hardware and software requirements, and the required accounts and permissions you need to run the Idera Dashboard and SQLdm web console in your environment, see Deploy the Idera Dashboard and SQLdm.

**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.

**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 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 document, Authentication in SQL Server.

**Product requirements**

The Idera Dashboard and SQLdm consist of a light, unobtrusive architecture that easily runs in your SQL Server environment with minimal configuration. Review system requirements for the Idera Dashboard and SQLdm.

- Learn about the Idera Dashboard's requirements.
- Learn about the SQLdm's requirements.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

Idera Dashboard and web console requirements

To successfully install the Idera Dashboard and SQLdm web console you need to comply with the following requirements:

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
</table>
### Port requirements:

The Idera Dashboard and SQLdm web console use the following ports:

- Idera Dashboard Core Services port: **9292**
- Idera Dashboard Application Service port: **9290**
- Idera Dashboard Web Application SSL port: **9291**

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more ➔

---

### SQLdm requirements

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

- **SQLdm 9.0** requires Microsoft SQL Server 2005 or above running on the computer that hosts the SQLdm Repository database for all installations.

**Tempdb monitoring is supported on SQL Server 2005 and above servers only.**

### Sample installations in large environments

While this topic provides the minimum requirements for installing SQLdm, users with large environments can benefit from the sample customer installations listed below.

**Installs 51-100 servers**

<table>
<thead>
<tr>
<th>Type</th>
<th>In-House Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>In-House Use</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Number of monitored servers</td>
<td>52</td>
</tr>
<tr>
<td>Services and Repository installed on same machine</td>
<td>Hyper-V</td>
</tr>
<tr>
<td>Services and Repository installed on same machine</td>
<td>Hyper-V</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 120 GB</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 120 GB</td>
</tr>
<tr>
<td>Monitoring features</td>
<td>Monitor Windows and SQL Server performance and error log monitoring</td>
</tr>
<tr>
<td>Monitoring features</td>
<td>Monitor Windows and SQL Server performance</td>
</tr>
</tbody>
</table>

### Installs 201-300 servers

<table>
<thead>
<tr>
<th>Type</th>
<th>In-House Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of monitored servers</td>
<td>90</td>
</tr>
<tr>
<td>Services machine</td>
<td>2 vCPU</td>
</tr>
<tr>
<td>Services machine</td>
<td>2 vCPU</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 100 GB</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 100 GB</td>
</tr>
<tr>
<td>Monitoring features</td>
<td>Monitor Windows and SQL Server performance</td>
</tr>
</tbody>
</table>

### Installs 201-300 servers

<table>
<thead>
<tr>
<th>Type</th>
<th>In-House Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of monitored servers</td>
<td>234</td>
</tr>
<tr>
<td>Services and Repository installed on same machine</td>
<td>Dell 2950</td>
</tr>
<tr>
<td>Services and Repository installed on same machine</td>
<td>Dell 2950</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 140 GB</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 140 GB</td>
</tr>
<tr>
<td>Monitoring features</td>
<td>Monitor Windows and SQL Server performance and use Query Waits for specific periods of diagnostics</td>
</tr>
<tr>
<td>Integrations</td>
<td>Send alerts to HP Open View via a command line program (opcmsg)</td>
</tr>
</tbody>
</table>

### 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>Number of monitored servers</td>
<td>274</td>
</tr>
<tr>
<td>Services and Repository installed on same machine</td>
<td>4 quad core processors</td>
</tr>
<tr>
<td>Services and Repository installed on same machine</td>
<td>4 quad core processors</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 1.5 TB</td>
</tr>
<tr>
<td>Data storage and grooming options</td>
<td>Repository size: 1.5 TB</td>
</tr>
<tr>
<td>Monitoring features</td>
<td>Monitor Windows and SQL Server performance</td>
</tr>
</tbody>
</table>

**Trial requirements**

Consider the following requirements when installing all SQLdm components on one computer.
SQL Server Standard or Enterprise Edition
- SQL Server 2005 SP1 or later
- SQL Server 2008 or later
- SQL Server 2008 R2
- SQL Server 2012
- SQL Server 2014

Hard Disk Space
200 MB for all SQLdm components

Windows Permissions
Your Windows Logon account includes local Administrator permissions

SQL Server Privileges
Either a Windows user account or SQL Server login that includes System Administrator privileges on the monitored SQL Server instances

**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 XP SP2 or later</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2003 SP2 or later</td>
</tr>
<tr>
<td></td>
<td>- Windows Vista SP2 or later</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2008 SP2</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2008 R2</td>
</tr>
<tr>
<td></td>
<td>- Windows 7</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2012</td>
</tr>
<tr>
<td></td>
<td>- Windows 8</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2012 R2</td>
</tr>
<tr>
<td></td>
<td>- Windows 8.1</td>
</tr>
<tr>
<td>64-bit Operating System</td>
<td>- Windows XP SP2 or later</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2003 SP2 or later</td>
</tr>
<tr>
<td></td>
<td>- Windows Vista SP2 or later</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2008 SP2</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2008 R2</td>
</tr>
<tr>
<td></td>
<td>- Windows 7</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2012</td>
</tr>
<tr>
<td></td>
<td>- Windows 8</td>
</tr>
<tr>
<td></td>
<td>- Windows Server 2012 R2</td>
</tr>
<tr>
<td></td>
<td>- Windows 8.1</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 includes Read Access to the SQLdm Repository. You can specify a SQL Server login instead.</td>
</tr>
<tr>
<td>Software Framework</td>
<td>Microsoft .NET 4.0 Full</td>
</tr>
<tr>
<td>Monitor Resolution</td>
<td>1280 x 1024 with small text for the best viewing of the SQLdm Dashboard</td>
</tr>
</tbody>
</table>

**SQLdm Services**

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
</table>
### Disk Space
- 50 MB

### Collection Service Permissions
- System Administrator privileges on the monitored SQL Server instances

### Management Service Permissions
- Read and Write privileges on the SQLdm Repository database

#### SQLdm Repository

SQLdm 7.2 and later includes a repository autogrow setting at 32 MB, which is sufficient for trial and small environments in production. Environments with larger, multi-server installations should change the repository autogrow setting to 250 MB or higher to prevent frequent data file autogrow, which can cause performance problems. Using percentage-based autogrow is not recommended.

<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>
<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></td>
<td>• SQL Server 2012</td>
</tr>
<tr>
<td></td>
<td>• SQL Server 2014</td>
</tr>
</tbody>
</table>

#### Monitored SQL Server Instances

<table>
<thead>
<tr>
<th>Type</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Server Standard or Enterprise Edition</td>
<td>• SQL Server 2000 SP4 or later</td>
</tr>
<tr>
<td></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></td>
<td>• SQL Server 2012</td>
</tr>
<tr>
<td></td>
<td>• SQL Server 2014</td>
</tr>
</tbody>
</table>

#### SQLdm Management Pack and SCOM support requirements

The Idera SQLdm Management Pack for integration with SCOM is designed for System Center Operations Manager 2007 R2 and above. For more information about SQLdm MP and SCOM installation, see [Install and configure the SQLdm Management Pack for SCOM](https://www.idera.com/support/). SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. [Learn more >>](https://www.idera.com/support/)

---

### How to install the Idera Dashboard and SQLdm

This procedure guides you through the installation of the Idera Dashboard and SQL diagnostic manager 9.0.

In a fresh installation, the installer automatically deploys the Idera Dashboard first and then SQLdm.

#### Start your Idera Dashboard installation

You can install the Idera Dashboard and SQLdm 9.0 on any computer that meets or exceeds the product requirements.

**To start installing the Idera Dashboard and 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 SQL diagnostic manager** on the **Quick Start** window.
4. On the **Welcome** window of the setup program, click **Next**.
5. Select an option to run the Idera Dashboard. You can install this framework locally or register with a previously installed copy on another server.
6. Specify a service account and password.

7. Run the Idera's Dashboard installer, and the Welcome to the Idera Dashboard Setup wizard displays.

8. Review the license agreement. To accept this license agreement, click I accept the terms in the license agreement, and then click Next.
Select an installation location and user access

You can use the default install location or specify a different path (Change). For your first install, we recommend using the default location. Click Next to continue with the installation.

Select a Windows user account to use for the service account

The Idera Dashboard uses this account to connect, discover, and gather configuration information from SQL Servers in your Business environment. The installer grants the "Log on as a Service" right to the account that you specify.

To specify a service account:
1. On the Service Account window, type appropriate credentials in the provided fields under which the services run.
2. Click Next.

**Select service ports to use for the Idera Dashboard**

The Idera Dashboard uses ports **9290**, **9291**, and **9292** for the web application and dashboard services respectively. However, you can specify different service ports.

**To specify a service port:**

1. On the Service Ports window, specify the ports you want the web application and dashboard services to use.

**Service Ports**

Specify the ports to be used by Idera Dashboard services.

The Idera Dashboard services will use these ports.

Specify the ports:

- Idera Dashboard Core Services Port: **9292**
- Idera Dashboard Application Service Port: **9290**
- Idera Dashboard Web Application SSL Port: **9291**

If Firewall is enabled make sure it allows TCP traffic through these ports.
2. Verify that the Firewall allows TCP (Transmission Control Protocol) traffic through your specified ports.
3. Click Next.

Select a SQL Server instance to use for the Idera Dashboard Repository database

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

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

1. On the Repository Database screen, specify the host SQL Server instance you want to use for the Idera Dashboard Repository database.
2. Type the name for the Idera Dashboard Repository database.

3. If your SQL Server environment uses SQL Server Authentication, select the check box and click Change to enter SQL Server credentials. By default, the Idera Dashboard uses your Windows credentials when connecting to the Repository database.
4. If you use Microsoft SQL Server Authentication, a new window displays where you can type the respective SQL Server login and password.
5. Click Next.

Grant access to the Idera Dashboard

The user that installs the Idera Dashboard has access to this application by default as a Dashboard Administrator. You can choose whether you want other users to have access to the Idera Dashboard in the Administration view. To continue with the installation, click Next.
1. Click **Install** to indicate that you are ready to complete your installation with the configurations you specified. If you want to review or change any of your settings, click **Back**.

2. Click **Finish** to exit the Idera Dashboard Setup Wizard.
3. Once you have configured SQLdm and the Idera Dashboard, click **Finish** to start the SQLdm installation and prompt its corresponding **Welcome Setup wizard**. To review or change any of your settings click **Previous**.

The Welcome to the Setup Wizard for Idera SQL diagnostic manager displays automatically.

**To continue installing SQLdm:**

1. On the **Welcome** window of the setup program, click **Next**.
2. Review the license agreement. To accept this license agreement, click I accept the terms in the license agreement, and then click Next.

Select an installation location and user access

You can use the default install location or specify a different path (Change). For your first install, we recommend using the default location. Click Next to continue with the installation.

Begin configuring your install

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

By default, the setup program installs the Management Service, Collection Service, and Predictive Service on the local computer where you are running the setup program.

Select a SQL Server instance to use for the Repository database

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

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 check box and click **Change** to enter SQL Server credentials. By default, SQLdm uses your Windows credentials when connecting to the Repository.

3. Click **Next**.

SQLdm 9.0 and later requires Microsoft SQL Server 2005 running on the computer that hosts the SQLdm Repository database for all installations.

**Select a Windows user account 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 verifies the connection before continuing.

**Complete the install**

Click **Install** to indicate that you are ready to complete your install with the configurations you specified. The SQLdm console automatically launches. To access the Idera Dashboard and the SQLdm web console, see **Log in the Idera Dashboard and web console**.

**Decide whether you want to deploy SQLdm Mobile**

SQLdm Mobile includes the Idera Newsfeed and several dashboard views that define your server health and expose critical alerts. You can access the SQLdm Mobile Web application 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.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

---

**Log in the Idera Dashboard and web console**

Once you have installed and configured your Idera Dashboard and SQL diagnostic manager deployments, you can login to the web console by doing the following:

1. Open your selected Browser, make sure it is compatible with the Idera Dashboard console requirements.
2. Type the Idera Dashboard product URL: **http://<machinename>:<port>** where **<machinename>** is the name of your host or machine, and **<port>** is the port specified during installation. The default URL is **http://<localhost>:9290** or **http://<machinename>:9290**.
3. When the Idera Dashboard web console launches on your browser, use your Windows user account **<domain\user>** with the respective password to log into the product.

---

The Idera Dashboard Web Application service comes with SSL already set up. For more information on running the Idera Dashboard over SSL, see Run the Idera Dashboard over SSL (HTTPS)

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

---

**Deploy SQLdm Mobile or Idera Newsfeed**
SQLdm Mobile is a Web-based application that includes the Idera Newsfeed technology and several dashboard views that define your server health and expose critical alerts. You can access the SQLdm Mobile Web site from most smart phones and mobile devices. The Idera Newsfeed is available quickly as a tab on the SQLdm Today page.

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

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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

SQLdm Mobile components and architecture including Idera Newsfeed

SQLdm Mobile is an easy-to-use, Web-based application that seamlessly integrates into your existing diagnostic and monitoring workflow. The SQLdm Mobile Web site includes many metrics and alerts panels 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 and 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 and Newsfeed Service

The SQLdm Mobile and 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 and Newsfeed setup program. Consider SQLdm Mobile and 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 panels so you can quickly check server performance when you are away from your desk.

Architecture
SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

### SQLdm Mobile and Idera Newsfeed requirements

You can easily 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 and Newsfeed Service account requires the following permissions. By default, setup program assigns the Local System account to the SQLdm Mobile and 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 and 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 and 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 and Newsfeed Service</td>
<td>Allows the SQLdm Mobile and 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>51665167</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>5168</td>
<td>Computer running the SQLdm Mobile and Newsfeed Service</td>
<td>SQLdm Mobile and 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 HTML 5, such as Google Chrome or Mozilla Firefox. SQLdm Mobile does not support Microsoft Internet Explorer 8 or earlier.

*If you plan to use IE 9 or later to access the SQLdm Mobile Web site*, make sure that you disable the Display intranet sites in Compatibility View setting. To change this setting, 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>
</tbody>
</table>
| Microsoft Internet Information Services (IIS) | Version 7.0 or later  
By default, IIS 7 installs with Windows Server 2008. You can enable this role using the native Server Manager. |
To successfully run the SQLdm Mobile Web application, make sure that you enable the **Common HTTP Features** setting and the following service roles 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 2005 and above

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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

### How to install SQLdm Mobile and Idera Newsfeed

This procedure guides you through the installation of SQLdm Mobile and Newsfeed. Before installation, 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

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

If you are interested in installing only the Idera Newsfeed without the additional components used with SQLdm Mobile, see How to install Idera Newsfeed only.

### Install or upgrade to SQLdm 9.0

SQLdm Mobile and the Idera Newsfeed technology are supported by SQLdm 7.0 or later versions. To successfully install and deploy SQLdm Mobile, first install or upgrade to SQLdm 9.0.

### Start your SQLdm Mobile and Newsfeed install

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

**To start installing SQLdm Mobile and Newsfeed:**

1. Log on with an administrator account to the computer on which you want to install the SQLdm Mobile and 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 **SQLdm Mobile and Newsfeed** on the **Quick Start** 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 in the license agreement, and then click **Next**.

### Begin configuring your install

We recommend installing the Mobile and 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 and Newsfeed Service is also installed in the same location.

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

**Add the Windows credentials to use for the SQLdm Mobile and Newsfeed Service account**

You can specify the Local System account or a Windows user account. Make sure that the specified account includes 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.

Select the SQL Server instance to use for the Repository database
Store the published status updates and other configuration settings in the SQLdm Mobile Repository.

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

To specify on which SQL Server instance you want to install the SQLdm Repository:
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.

Add the authentication credentials to create the Repository database
You can use either Windows or SQL Authentication to create the Mobile Repository database. Make sure the specified Windows account includes the required permissions.

To specify a service account:
1. Choose the authentication that you want to use, 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.

Add the authentication credentials to connect to the Repository database at run time
You can use either Windows or SQL Authentication to connect to the Mobile Repository database. Make sure the specified Windows account includes the required permissions.

To specify a service account:
1. Choose the authentication that you want to use, 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.

Select a location for the SQLdm Management Service
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. Make sure 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 and Newsfeed setup program and navigate to the Custom Setup window.
3. Choose the Mobile Web components, and then click Next.

Select the Web Server settings you want the SQLdm Mobile Web application to 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.

Specify the name of the computer that hosts the SQLdm Mobile and Newsfeed Service so that you can send communications between the SQLdm Mobile components and SQLdm.
Make sure your Web Server meets the SQLdm Mobile requirements.

To configure the Web Server settings:
Choose whether to install the SQLdm Mobile Web application as a new Web site.
Specify the name of the computer on which you installed the SQLdm Mobile and Newsfeed Service.
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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

How to install Idera Newsfeed only

This procedure guides you through the installation of Idera Newsfeed only and none of the additional components used with SQLdm Mobile. Use this solution in a larger environment where your SQL Server team or environment is spread across multiple locations and you need an easier, more streamlined way to troubleshoot issues. The Idera Newsfeed can help you capture or bookmark the team resolutions for future reference. You can implement this option without need to allot the additional hardware and IIS resources necessary to deploy a complete, mobile-available solution.

Before installing Newsfeed, review the associated architecture and components.

Install or upgrade to SQLdm 9.0

The Idera Newsfeed technology is supported by SQLdm 7.0 and above versions. To successfully install and deploy Idera Newsfeed, first install or upgrade to SQLdm 9.0.

Start your Idera Newsfeed install

Before installing Newsfeed, review the associated architecture and components. You can install the Idera Newsfeed on any computer that meets or exceeds the product requirements. By default, the SQLdm Mobile and Newsfeed setup program installs the SQLdm Mobile and Newsfeed Service to the local computer.

To start installing Idera Newsfeed:

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

Begin configuring your install

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 and Newsfeed Service is also installed in the same location.

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

Add the Windows credentials to use for the SQLdm Mobile and Newsfeed Service account

You can specify the Local System account or a Windows user account. Make sure the specified account includes 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.

Select the SQL Server instance to use for the Repository database

The Idera Newsfeed Repository is where the published status updates and other configuration settings are stored.

SQLdm Mobile requires Microsoft SQL Server 2005 or above running on the computer that hosts the Newsfeed 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.
Add the authentication credentials to create the Repository database

You can use either Windows or SQL Authentication to create the Newsfeed Repository database. Make sure the specified Windows account includes the required permissions.

To specify a service account:

1. Choose the authentication that you want to use, 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.

Add the authentication credentials to connect to the Repository database at run time

You can use either Windows or SQL Authentication to connect to the Newsfeed Repository database. Make sure the specified Windows account includes the required permissions.

To specify a service account:

1. Choose the authentication that you want to use, 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.

Select a location for the SQLdm Management Service

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

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

Complete the Newsfeed install

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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Deploy SQLdm Console only

Idera provides a solution for administrators who have SQLdm deployed within their environment, and want to provide their users with the SQLdm desktop client console without exposing other SQLdm components. Administrators can use this process to prevent a user from unknowingly installing an additional service component, which can cause issues with SQLdm within your environment.

Store the SQLdm Console-only executable

You can store the SQLdm Console-only executable (.exe) files on many types of media for distribution, such as a portable or network drive. Your end users can install the SQLdm desktop client console on any computer that meets or exceeds the product requirements.

The following files reside in the location where you installed SQLdm. Copy one or both of these files to your distribution media. Note the first file is for x64 users and the second is for x86 users.

```
c:\Program Files\Idera\SQLdm x64 Installation Kit\x64\SQLDiagnosticManager-x64_c.exe
c:\Program Files\Idera\SQLdm x86 Installation Kit\x86\SQLDiagnosticManager_c.exe
```

How to install SQLdm Console only

This procedure guides you through the installation of the SQLdm Console only.

You must have a full installation of SQLdm already running in your environment before attempting to install the SQLdm desktop client console.

Perform your install
You can install SQLdm desktop client console on any computer that meets or exceeds the product requirements.

To install SQLdm Console only:

1. Log on with an administrator account to the computer on which you want to install the SQLdm desktop client console.
2. Navigate to the location where the SQLdm Console file resides in the distribution media. If you use an x64 operating system, run `SQLDiagnosticManager-x64_c.exe`. If you use an x86 OS, run `SQLDiagnosticManager_c.exe`.
3. On the Welcome window of the setup program, click Next.
4. Review the license agreement. To accept this license agreement, click I accept the terms in the license agreement, and then click Next.
5. Select the default installation location or specify a different path.
6. Choose whether you want any user or only the current user to access this application, and then click Next.
7. Verify that SQLdm displays Console Only as selected, and then click Next.
8. Click Install to indicate that you are ready to complete your install with the configurations you specified. Click Finish to exit the setup program.

Complete the post-installation steps

When you first open the SQLdm desktop client console, you must complete the fields necessary to connect to your existing SQLdm Repository. It is important that you have the information available to connect to the proper SQL Server instance and SQLdm Repository database using the proper authentication credentials.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Deploy SQLdm in a high availability group

This procedure guides you through the installation of SQLdm in a high availability group. SQLdm does not support direct installations under this scenario; however, you can deploy the SQLdm Repository in one of the nodes that belongs to a specific availability group and configure an availability group listener that serves as a global link between nodes.

Make sure to restart the SQLdm services, apply the newly-generated license using the Manage Licenses button before attempting to apply the license directly using the Management Console or you may encounter issues.

1. Deploy the SQLdm Repository targeting a node that is part of an availability group. For example, `Node 1\Instance A`.
2. Move the SQLdm Repository to an availability group in the primary replica with Listener Y and port 300 (as example). To review how to create an availability group, see Microsoft document Creating and Configuring a New Availability Group.
3. A new license key for Listener Y, Port 300 is required. See Capturing an availability group's listener name and port designation to send your request to Idera Sales.
4. Using the SQLdm Management Service Configuration wizard modify the target repository to Listener Y, port 300.
5. Save changes, and then restart the SQLdm services. This process takes approximately 5 minutes.

6. Upon opening the SQLdm Console, the system recognizes that you are trying to access the selected node without the proper license and throws an error message.
6. Click the "Manage Licenses" button of the error message and type the new license for Listener Y, Port 300. Open the SQLdm Console, go to Help > Manage Licenses in the Toolbar menu, and the License Keys window displays. Type the new license key in the available space, and then click Enter.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

7. AlwaysOn availability group listener name and port designation

This procedure guides you through the steps necessary to capture an availability group's listener name and port designation. This information is essential for Idera Sales to generate a special license that allows you to host the SQLdm Repository in an availability group. As a pre-requisite the SQLdm Repository needs to be hosted in an availability group within the primary replica.

1. Within SSMS, go to node that contains the availability group that hosts the SQLdm Repository and is the primary replica.
2. Expand the **AlwaysOn High Availability** folder and then the **Availability Groups** folder.

3. Expand the availability group that is the primary replica.

4. Expand the **Availability Group Listeners** folder.
5. Right-click the listener, select Properties, and the Availability Group Listener Properties dialog displays.

6. Send Idera Sales the **Listener DNS Name** and **Port designation** information to obtain a new license to host the SQLdm Repository in an availability group.
Configure your deployment

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

Connect to the SQLdm Repository

The first time you open the SQLdm Management Console, SQLdm displays the Connect to SQLdm Repository window. Use this window to select the SQL Server instance, SQLdm Repository database name, and enter the authentication details the SQLdm Console uses to communicate with the SQLdm Repository.

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

Access the Connect to SQLdm Repository window
You can open the Connect to SQLdm Repository window 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

---

**Add your SQL Server instances**

The Add Servers wizard allows you to add 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.

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.

**Access the Add Servers wizard**

You can open the Add Servers wizard from the SQLdm Console menu by selecting **File > Manage Servers**, and then clicking Add in the Manage Servers window.

**Other configurations that need attention when adding a virtualized SQL Server instance**

If the monitored SQL Server instance you want to add resides on a virtual machine, add the instance using the Add Servers wizard, and then access the VM Configuration window to associate the instance with the correct virtual machine and host.

**About alert templates**

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.

**Delete a SQL Server instance**

You can easily delete a SQL Server instance at any time by right-clicking the appropriate instance, and then selecting Delete. SQLdm displays a confirmation message that asks, “Would you like to retain the data collected for the selected instance?” Click Yes and SQLdm retains data for the instance and includes it in some reports, such as the Server Inventory report. Click No and SQLdm deletes the SQL Server instance and all related data.

---

**Configure authentication**

The Configure Authentication window of the Add Servers wizard allows you to select the authentication credentials SQLdm uses when collecting performance and statistics information from your SQL Server instances. You can also set encryption to secure the connection.

After selecting a connection type and encryption, click Next to continue.

**Access the Add Servers wizard**

You can open the Add Servers wizard from the SQLdm Console menu by selecting **File > Manage Servers**, and then clicking Add in the Manage Servers window.

**Select a connection type**

Windows Authentication is the recommended collection type as SQLdm uses a number of SQL trace statements that require a trusted connection. When you select and use Windows Authentication, the account the SQLdm Collection Service is running as is used to connect to the SQL Server instance.

**Select encryption**

SQLdm allows you to designate encryption methods used to encrypt data between the SQLdm Collection Service and the monitored SQL Server instance. The options include SSL (Secure Sockets Layer) or SSL with Trust Service Certificate.
Encrypt Connection (SSL)

The Encrypt Connection (SSL) option adds a flag in the connection properties that triggers SSL use when the SQLdm Collection Service connects to a monitored server that uses SSL to encrypt the data transfer. For this option to work correctly, you must configure the monitored SQL Server instance to use SSL encryption.

Trust Server Certificate (Bypass Certificate Validation)

The Trust Server Certificate (bypass certificate validation) option is available only after you select Encrypt Connection (SSL). This option allows you to skip certificate validation 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, then the connection is rejected unless you select the Trust Server Certificate option.

Select instances to monitor

The Select Servers to Monitor window of the Add Servers wizard allows you to select one or more of the SQL Server instances you want to monitor using SQLdm. If the name of the SQL Server instance you want to monitor does not appear in the list, type the instance name in the space provided in the Available Servers area. 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.

Configure data collection

The Configure SQL diagnostic manager Collection window of the All Servers wizard allows you to set the interval at which you want to collect diagnostic data and raise alerts, and whether you want to enable query monitoring.

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.

The Query Monitor is a standard SQL Server trace that collects events occurring on your SQL Server instance over a period of time.

After making any changes to the collection interval or enabling the Query Monitor, click Next to continue.

Monitoring your environment by collecting query activities

When you experience query timeouts or other performance issues, you can collect query-related activities through the Query Monitor. Note that enabling these options can degrade performance on the monitored SQL Server instance because of the additional data collection and storage. To help alleviate degradation, SQLdm allows you to select only those types of queries you want to monitor such as batches, statements, procedures,
and triggers. Additional threshold fields allow you to set the levels at which you consider a query as performing poorly.

**Using advanced Query Monitor options**

Query Monitor includes advanced options that allow you to filter applications, databases, and SQL text from collection. Use semicolons ( ; ) to separate multiple entries. You can also use the percent character (%) as a wildcard.

You can access this window by clicking the **Advanced** button in the Query Monitor tab of the monitored SQL Server Properties window.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

**Configure OS metric collection**

The Configure OS Metric Collection window of the Add Servers wizard lets you select which type of operating system metrics collection you want to use or whether you do not want to collect any OS metrics from a SQL Server instance. SQLdm uses OLE automation or a direct connection to the Windows Management Interface (WMI) when collecting OS metrics and service status data on clustered SQL Servers. Collect Operating System data using direct WMI is selected by default when adding a new SQL Server instance.

Select **Do not collect Operating System data** if you do not want to collect these statistics. For additional information about OS metrics collection, see Configure OS metrics monitoring.

SQLdm retains your previous settings upon upgrade if OLE automation is the default for your monitored SQL Server instance.

After selecting your method for OS metrics collection, click Next to continue.

**Access the Add Servers wizard**

You can open the Add Servers wizard from the SQLdm Console menu by selecting File > Manage Servers, and then clicking **Add** in the Manage Servers window.

**Permissions necessary for direct WMI collection**

It is preferred that the WMI user is a local administrator on the monitored SQL Server instance. However, if you do not want to grant administrator access, use the following steps to configure remote WMI access in Microsoft Windows:

1. Add the user account to the Performance Log Users and Distributed COM users groups.
2. Grant privileges to WMI.

You also may need to add the WMI user account to the following policies:

- Act as part of the operating system
- Log on as a batch job
- Log on as a service
- Replace a process level token

For more information about using a direct WMI connection, see the Microsoft document, Securing a Remote WMI Connection.

**Disabling OS metric collection**

SQLdm does not require OS metrics collection. However, if you disable this feature, SQLdm ceases to collect OS metrics and does not raise any previously-associated alerts. You can easily enable OS metric collection at any time if you decide that you want to continue collecting these metrics.

**Using direct WMI when the collection service is on a different domain than the SQL Server**

In environments where cross-domain trust security is enforced, data from asynchronous collection is not returned from an untrusted domain to a trusted domain. As a result, the collection service does not receive the data.

If you select Collect Operating System data using direct WMI and the machine used for your collection service is in a different domain than your monitored SQL Server instance, you must make some configuration changes on your collection service machine. If you do not make these changes, change your OS metrics collection to Collect Operating System data using OLE automation if you want OS and disk statistics plus the auto discovery of mount points.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

Select an alert template and tags
The Select Alert Template and Tags window lets you assign an alert template and tags while you add an instance. Alert templates allow you to configure generic alert settings to apply to your servers or groups of servers. Tags are labels that help you organize server instances into meaningful groups.

To apply an alert template, select the appropriate template from the list. If you do not have an existing custom alert template, 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 tags by clicking Add Tag.

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.

After selecting your baseline collection, click **Next** to continue.

**Access the Add Servers wizard**

You can open the Add Servers wizard from the SQLdm Console menu by selecting **File > Manage Servers**, and then clicking **Add** in the Manage Servers window.

**Using tags**

Tags are customizable to suit 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).

You can assign multiple tags to each instance so that the instance appears in several different groups.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

**Configure your virtual machine connections**

The VM Configuration window allows you to associate your monitored virtual SQL Server instances with the virtualization host server that manages your VMs and the VM on which the monitored SQL Server instance is running.

Virtual machines appearing in the list are based on the permissions for the user account typed into the Virtualization Host Configuration dialog box. To view a specific VM, you must have the View permission on all parent folders.

Remember to renew your link after changing the name of a virtual machine. Once the name change is complete, simply open the VM Configuration window and wait until after SQLdm is done contacting the vCenter server or Hyper-V Host server to get a list of your VM names. Note that your updated VM name already appears in the list. Click OK to save the new name to the list of VMs. If you do not update the name, this action occurs again the next time you access the VM Configuration window.

**Access the VM Configuration window**

To open the VM Configuration window, select **Tools > VM Configuration**, or click VM Configuration from the Manage Servers window. You can also access this window from the Virtualization tab of the Server Properties window by clicking the VM Configuration button.

**Add a host server**

SQLdm allows you to add a new host server through the VM Configuration window. Click New, and then complete the access information for the server. When you complete all the available fields, SQLdm allows you to verify connectivity with this server. Click Test to make sure you typed the correct access credentials. Click OK after a successful test.
Verify your connection to the host server

You can test the connection to your listed host servers by selecting the appropriate server from the list, and then clicking Test. SQLdm displays an icon with your current connection status. If the connection fails, click View/Edit and verify your connection details. Make the necessary changes to access your host server, click Test, and then click OK to save your changes after a successful test.

Associate a monitored SQL Server instance to a VM

Once you add your SQL Server instances to SQLdm for monitoring, you must associate them with their host server and virtual machine.

To associate a monitored SQL Server instance to a virtual machine:
1. Verify that the instance you want to associate to the host appears in the list.
2. Click the Virtualization column for the appropriate virtual instance, and then select the associated host server from the drop-down list.
3. Click the Virtual Machine column, and then select the name of the virtual machine where this instance resides. SQLdm populates this drop-down list based on the servers on the selected host server.
4. Click Link Instances to VMs.
5. Click OK to save your changes and close this window.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Configure your VM host server connection

The vCenter Connection Configuration window allows you to add or update connection details for your virtual machine host servers. Complete the access information for the server, and the click Test to make sure you typed the correct access credentials. Click OK after a successful test.

Occasionally, a connection test fails. Make sure you typed accurate information in the appropriate fields, and then click Test. If the test fails again, there may be an issue with the server to which you are trying to connect.

Access the vCenter Connection Configuration window

To access the vCenter Connection Configuration window, first access the VM Configuration window by selecting Tools > VM Configuration, or by clicking VM Configuration from the Manage Servers window. You can also access this window from the Virtualization tab of the Server Properties window. Once in the VM Configuration window, click New.

vCenter Server port requirements

The vCenter Server requires that you open the ports included in the following table. For the most current list of ports, see the VMware Knowledge Base article regarding network port requirements.

<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>Required for direct HTTP connections. Note that this port redirects requests to HTTPS Port 443.</td>
</tr>
</tbody>
</table>
Active Directory Services for the vCenter Server group.

Lists for connection from the vSphere Client, vSphere Web Access Client, and other SDL clients.

SSL port of the local instance for vCenter Linked Mode.

Used to send data to managed hosts.

Used by the vSphere Client to display virtual machine consoles.

CIM transaction traffic (Hardware Status).

vCenter Management Webservices HTTP.

Secure connection for vCenter Management Webservices HTTPS.

Used to stream inventory object changes to SDK clients. Firewall rules for this port on the vCenter Server can be set to block all, except from and to localhosts if the clients are installed on the same host as the vCenter Server service.

---

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Configure console options**

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**
  - The General tab allows you to set how often the console refreshes data, hide the Console when minimized, select how SQLdm manages chart data, and choose what colors you want SQLdm to use.

- **Alerts**
  - The Alerts tab allows you to limit the number of historical alerts displayed in the Alerts view, which can improve the performance of the view when there is a large number of alerts.

- **Notifications**
  - The Notifications tab allows you to configure how SQLdm notifies you when alert data crosses a threshold.

**Define common properties**

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

**Access the Console Options window General tab**

You can open the General tab of the Console Options window from the SQLdm Console by selecting **Tools > Console Options**. The General tab is available by default.

**Set your 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:

- **Server 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 is 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 is between one and 60 minutes.

**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.

**Use chart options**

Chart options allow you to determine how much data is shown in your real-time charts. The entry in the Keep data for the last field is always greater than the entry in the Show data for the last field.

- **Keep data for the last**
  
  Determines the length of time data is stored in the SQLdm Repository.

- **Show data for the last**
  
  Determines the time period displayed in real-time charts.

**Use a color scheme**

The Color scheme field allows you to select the background color you prefer for the SQLdm Management Console. By default, the scheme is set to Black.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Configure the Alerts view**

The Alerts tab of the Console Options window allows you to enter the number of rows you want displayed in the historical Alerts view. While this restriction limits the historical view, there is no limit to the number of rows used for your active alerts. The default entry is 0, which does not limit the number of rows in your historical view.

**Access the Console Options window Alerts tab**

You can open the Alerts tab of the Console Options window from the SQLdm Console by selecting Tools > Console Options, and then selecting the Alerts tab.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Configure notification settings**

The Notifications tab of the Console Options window allows you to configure when and how the SQLdm Console displays alert notifications. When you select an option to show notifications, SQLdm displays a pop-up notification in your task bar each time it reaches a threshold.

To see an example of a console alert, click Preview, and the alert pop-up appears as shown in the following image.
Access the Console Options window Notifications tab

You can open the Notifications tab of the Console Options window from the SQLdm Console by selecting Tools > Console Options, and then selecting the Notifications tab.

Selecting visual notifications

SQLdm allows you to select when and for which alert states you receive alert notifications. You can select from the following options regarding when you want SQLdm to display notifications:

- **Always show console alert notifications**
  Provides alert notifications occur after every refresh.

- **Only show console alert notifications when a state transition occurs**
  Limits the alert notifications you receive on refreshes to only those that change states.

- **Never show console alert notifications**
  Prevents the occurrence of all console notifications, but you continue to receive all email alerts. This setting is dependent upon your particular needs and you can modify the alert at any point. If you receive too many notifications, consider limiting notifications to occur only after an alert state transition. However, if you monitor a problematic SQL Server and want to know when a threshold is met, consider monitoring all alert states.

Select from the following options regarding which alert statuses trigger notifications:

- **Show all alert statuses**
  Display notifications for all alert state changes.

- **Show Warning and Critical alerts**
Display notifications only when an event triggers a Warning or Critical alert.

**Show Critical alerts only**

Display notifications only when an event triggers a Critical alert.

**Selecting audible notifications**

The options in the When should SQLdm play sounds section allow you to toggle whether you want to hear an audible notification along with the alert in SQLdm whenever an event triggers a critical alert. The default For all console notifications option helps to let you know of any notification within SQLdm. Select Only for critical notifications and SQLdm plays the audible notification only when an event triggers a critical alert.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

### Set your SQLdm Repository grooming schedule

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.

- 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.

**Access the Grooming Options window**

To open the Grooming Options window, select Tools > Grooming Options from the toolbar menu.

**Available Grooming options**

You can set the time when SQLdm grooms the Repository and how many days your metrics, session details and queries, change log data and inactive alerts 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.

---

### Use custom counters to track metrics

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 used for historical trending, on-demand monitoring, and schedule refresh alerting.

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

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
- VM and host server performance counters accessible through the host Server

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.

**Access the Custom Counters view**

To open the Custom Counters view, click Administration > Custom Counters from the navigation pane.
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.

Using custom SQL scripts

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

SQLdm does not scrub custom SQL batches for SQL Injection protection. Use caution when entering custom SQL scripts as it is an advanced feature. The go command is not a Transact-SQL statement and you cannot use this command in custom SQL batches. The go command is a batch separator for Management Studio, Query Analyzer, sqlcmb, and osql.

Locate your custom counters

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. You can access this window by selecting details on the Server view or by double-clicking Custom Counters in the Servers tree.

Custom Counters table

SQLdm provides 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 or select from a list of virtualization performance counters if you have monitored servers hosted on a virtual machine. 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.

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking Administration > Custom Counters, and then clicking Add in the Custom Counters view.
The Select Counter Type window of the Add Custom Counter wizard allows you to choose the method you want SQLdm to use when collecting the counter. SQLdm offers 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. Note that your script must return a single numeric value.

**Virtual Machine Counter**
Select this option to pick from counters collected using an API from VMware or Hyper-V.

**Access the Add Custom Counter wizard**
You can open the Add Custom Counter wizard by clicking Administration > Custom Counters, and then clicking Add in the Custom Counters view.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

Select windows system counter

The Select Windows System Counter window of the Add Custom Counter wizard 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.

**Access the Add Custom Counter wizard**
You can open the Add Custom Counter wizard by clicking Administration > Custom Counters, and then clicking Add in the Custom Counters view.

**Select a 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 populates (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.

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

**Manually add a counter**

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 does not work correctly.

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

Win32_PerfFormattedData_PerfOS_Processor

**You can find custom counter programmatic names on Microsoft TechNet.**

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
Select SQL Server system counters

The Select SQL Server System Counter window of the Add Custom Counter wizard allows you to manually enter counter information or select counters from a list populated from those currently available on an existing monitored SQL Server.

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 does not work correctly.

To select a SQL Server system counter:
1. Select the appropriate SQL Server instance from the drop-down list.
2. Select the object name.
3. Select the name of the specific counter you want to add.
4. Select the instance, if applicable. The instance name is what distinguishes it from other similar objects.

If no counters appear in the drop-down list, either the counters are not installed or they are disabled, possibly by a recent Windows Service Pack. Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking Administration > Custom Counters, and then clicking Add in the Custom Counters view.

How to calculate this metric

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.

Understand why the SQL Server counter grows

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 always shows the total number of page lookups since the server was first started, while when in the value per second mode, it shows the number of lookups per second during the monitored period.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >

Provide custom SQL script

The Provide Custom SQL Script window of the Add Custom Counter wizard allows you to enter or paste a T-SQL script for your custom counter. For more information about 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 is executed as entered, which could result in a negative impact on the linked monitored SQL Server instances.
- The go command is not a Transact-SQL statement and you cannot use this command 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 two 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 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

Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking Administration > Custom Counters, and then clicking Add in the Custom Counters view.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >
The Select Virtual Machine System Counter window of the Add Custom Counter wizard allows you to select the counter you want from a list populated by those counters currently available on an existing monitored SQL Server host server, or you can enter the counter information manually.

You can select virtual machine counters for both the Hyper-V and VMware server virtualization platforms.

**Access the Add Custom Counter wizard**

You can open the Add Custom Counter wizard by clicking Administration > Custom Counters, and then clicking Add in the Custom Counters view.

**Select the counter from the list**

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

**Add Custom Counter**

![Add Custom Counter window](image)

Select Virtual Machine System Counter

Select the Virtual Machine System Counter you would like to monitor.

- Select a counter from an existing monitored SQL Server host (Recommended)
- Manually enter counter configuration

<table>
<thead>
<tr>
<th>Select Counter</th>
<th>SQL Server: DM-HT-3w2012R2</th>
<th>Object Name: VM - CPU</th>
<th>Counter Name: &lt;Select a counter&gt;</th>
<th>Instance Name: &lt;Select a counter&gt;</th>
</tr>
</thead>
</table>

1. Pick the monitored 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.

**Add a counter manually**

If you have the counter information available, you can enter the information quickly by selecting Manually enter counter configuration, and then simply pasting or typing the name of the object, counter, and SQL Server instance into the associated fields. The counter information entered must be exact or the counter does not work correctly.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more ->
Specify counter value

The Custom Counter Value window of the Add Custom Counter wizard 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.

**Access the Add Custom Counter wizard**

You can open the Add Custom Counter wizard by clicking Administration > Custom Counters, and then clicking Add in the Custom Counters view.

**Scale factor examples**

The following table covers some conversion factors that allows 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>0.00000953674316</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>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>

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
Specify counter definition

The Customize Counter Definition window of the Add Custom Counter wizard is where you name, assign to a category, and enter a description of your new custom counter. The information entered on the Custom Counter Definition window appears in various places in the SQLdm interface.

You cannot change the name or category of a counter once you create it.

**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.

### Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters** and then clicking **Add** in the Custom Counters view.

**Configure alert settings for this counter**

The Configure Alert Settings window of the Add Custom Counter wizard allows you to configure how SQLdm uses and monitors your new custom counter information.

**Configure Alert Evaluation**

This option allows you 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**.

### Access the Add Custom Counter wizard

You can open the Add Custom Counter wizard by clicking **Administration > Custom Counters** and then clicking **Add** in the Custom Counters view.

**Configure a custom response when SQLdm generates an alert**

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

To configure a custom response when an alert is generated, select **Tools > Alert Actions and Responses** from the console toolbar.

### Edit a custom counter

You cannot change the name or category of a counter once you create it.
The Edit Custom Counter wizard allows you to edit your custom counter settings. To edit a custom counter, select the counter from the list on the Custom Counters window and click the Edit button.

You cannot change the custom counter name and category once you create a Counter.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

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.

Test a custom counter

The Test Custom Counter window allows you to verify that the custom counter is what you want to collect 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.

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 to monitored instance

The 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. 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 exist.
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.
Manage tags

The Manage tags window allows you to Add, Edit and Remove Tags. For more information on tags and their functionality, see Working with tags.

Add a new tag

The Add Tag window allows you to create and assign a tag while adding an instance. Tags are labels that allow you to organize server instances into meaningful groups.

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. You can assign multiple tags to each instance so that the instance appears in several different groups.

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

Edit a tag

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.
Navigate the Idera Dashboard

What is the Idera Dashboard?

The Idera Dashboard is a common technology framework designed to support the entire Idera product suite. The Idera Dashboard allows users to get an overview of the status of their SQL Servers and hosted databases all in a consolidated view, while providing users the means to drill into individual product dashboards for details. The Idera Dashboard supports multiple copies of SQL diagnostic manager installations. Click image to view full size.

In the Idera Dashboard all products show a common Administration tab, granted the logged-in user has administrator privileges. Selecting this tab, displays the Administration view which hosts a range of widgets for performing administration-related actions.

What actions can be performed in the Administration view of the Idera Dashboard?

The Administration view of the Idera Dashboard provides a central set of services related to specific actions such as:

- Security
- Product registry
- Instance registry

For more information on each service and what configuration settings are available, visit each respective section.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Managing users in the Idera Dashboard

The Users widget of the Administration view, allows users to grant access to other team members or groups, and manage their roles. Users with administrative privileges are divided in:

- Dashboard administrators - capability to manage access over Dashboard functions as well as individual products' functions.
- Product administrators - capability to grant access to individual products for which they have administrative rights.

To add new users, edit their details (name, subscription, or email address), or remove them, select Manage Users in the Administration view, and the Manage Users window displays:
Adding a user in the Idera Dashboard

In the Idera Dashboard access is granted to Windows users or groups. To add users follow these steps:

1. Click the Add User / Group option and the Add User/Group dialog displays.

2. Type the name of the user you want to grant access to. You should enter a Windows user name in the following format: `<domain\user>`.
3. Select User or Group in the Account Details field.
4. Check the Do not timeout the browser session for this account check box to stay logged in.
5. Check the Send welcome email check box to provide new user with information about the product and URL for the Idera Dashboard console.
6. Type the email address where you want the user to receive alert emails.
7. In the Product field you can select to add user to the Idera Dashboard or SQLdm.
8. If you select the first option (Idera Dashboard), in the Role field you can assign user the Dashboard Administrator or Dashboard guest roles.
9. If you select the second option (SQLdm), in the Role field you can assign user the Product Administrator, Product user or Product guest roles.
10. Click SAVE.

Editing a user in the Idera Dashboard

This option allows you to edit the account name, change the email address where user receive alerts, disable his/her account, and add new permissions. To edit a user or group follow these steps:

1. Select one user from the list of users, click the Edit icon and the Edit User/Group dialog displays:
Change the necessary setting.

2. Click **SAVE**.

**Removing a user from the Idera Dashboard**

This option allows you to remove a user from access to the Idera Dashboard. To delete a user or group follow these steps:

1. Select one user from the list of users, click the **Remove** icon (as seen above).
2. A warning that requires a confirmation whether you want to delete the selected user or group displays.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

Managing products in the Idera Dashboard

The Idera Dashboard hosts Idera products that register with the dashboard. The Products widget of the Administration view, allows users to view and manage registered products.

To edit or remove registered products, select **Manage Products** in the Administration view, and the Products window displays:
To edit a product, follow these steps:

1. Select one product from the list of products, click the **Edit** icon (as above) and the Edit Product Information displays:

   ![Edit Product Information](image)

   - **Product Name**: SQLDm
   - **Instance Name**: null
   - **Version**: 3.0.0.0
   - **Registered**: 13/01/2015 03:11 AM
   - **Location**: cr-2kfr\SQLDmRepository
   - **Connection User**: simpson\adminstrator
   - **Connection Password**: To change the saved password, click here.
   - **Short Name**: SQLDm

2. Make the necessary modifications.
3. Click **SAVE**.

To remove a product, follow these steps:

1. Select one product from the list of products, click the **Delete** icon.
2. A warning that requires a confirmation whether you want to unregister the selected Idera product displays.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

---

**Managing instances in the Idera Dashboard**

The Idera Dashboard tracks SQL Server instances, discovered and managed by different Idera products. The Instances widget of the Administration view, allows users to view and manage registered instances.

To view coverage or remove registered instances that no longer exist in your SQL Server environment, select **Manage Instances** in the Administration view, and the Managed Instances window displays. The View filter allows you to select from:

- **All**: Lists all instances discovered in your SQL Server environment and network.
- **Managed**: Lists only those managed instances in various Idera products.
- **Unmanaged**: Lists instances discovered on the network but not registered.
SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Navigate the web console dashboard

The SQLdm web console interface boasts an easy-to-navigate dashboard comprised of several views, sub-views, tabs, and widgets that provide a complete synopsis of your SQLdm environment. Click the image to view full size.

On the dashboard you can find the following sections:

- Views by Status, Tags, or Critical Instances
- List, thumbnail and heatmap sub-views
- Active alerts roll up view
- Top servers by alert count and response time widgets
- Top databases by alerts
- Top alert categories
- Top X Lists tab
- Alerts tab

Additional options

Search

In the SQLdm web console dashboard, you can search for a particular instance by name. Use the percent (%) wildcard character if you do not remember the full name of the instance that you are looking for.

Group by

In the SQLdm web console dashboard, you can group returned information from the different views by Severity and Tags.

Refresh

The Refresh button allows you to manually refresh displayed data.

System menu

The system menu provides options for help and navigation to Idera sites such as the knowledge base, community forum, and customer support.
Use views to get an overview of your environment

In the Views section of the web console's dashboard you can see at a glance if there are any problems in your monitored SQL Server instances and drill-down for details. Each view is expandable and information is grouped By Status, By Tags, or by Most Critical Instances. SQLdm displays this information in the panel to the right.

**By Status**

In the By Status view you can easily identify the status of your monitored SQL Server instances and take action.

Instances are grouped as follows:

- All Instances
- OK
- Warning
- Critical
- Maintenance Mode

If you select all instances with an **OK** status in the By Status view, the panel displays information only pertaining to those instances.

**By Tags**

In the By Tags view, you can view the status of different tags or group of instances, and drill-down to details. For example, verify if problems occur in the **AlwaysOn, Mirroring** or **SQL 2005** servers (click image to view full size).
If you select the **Mirroring** tag in the **By Tags** view, the panel displays information only pertaining to those instances. For additional information on tags, see **Work with tags**.

### By Most Critical Instances

The Most Critical Instances view lists all the critical SQL servers in your environment and provides links to the single instance dashboard so you can quickly review the issue. The instance with the largest number of alerts shows up at the top of the list.

If you click any of the listed critical instances, the **single instance dashboard** displays.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more > >]

Use sub-views to identify and diagnose issues

The web console’s main dashboard is comprised of three main sub-views:

- List sub-view
- Thumbnail sub-view
- Heatmap sub-view
These sub-views display all data collected at the last refresh for all your monitored SQL Server instances. To access any sub-view, click the appropriate button in the upper-right corner of the dashboard as shown in the following image:

**List sub-view**

The list sub-view is the view by default in the SQLdm web console's dashboard and offers a means to quickly identify the status of instances, as well as the possibility to navigate to the single instance dashboard when clicking on a specific row.

The list sub-view provides you with the following SQL Server information:

**Instance Name**
Displays the name of the monitored SQL Server instance.

**SQL Server Version**
Displays the edition and version of SQL Server software used in this instance.

**SQL Server Status**
Displays the status of the SQL Server service such as running, stopped, and paused.

**SQL Agent Status**
Displays the status of the SQL Server Agent service such as running or stopped.

**DTC Status**
Displays the status of the Distributed Transaction Coordinator service such as running or stopped.

**Available Memory (MB)**
Indicates how much available memory is allocated for SQL Server.

**Blocked Sessions**
Displays blocked process sessions information on the SQL Server instance.

**CPU Activity**
Displays the percentage of CPU consumed by your SQL Server, virtual machine, and host server.

**Actions**

**Sort**
Instances on the list sub-view are sorted by severity. To re-organize information, click a column header or select the option to group by severity or tags (Group By).

**Export**
You can export instance information in PDF, XLS, or XML by clicking EXPORT at the top of the list sub-view.

**Show/Hide instances**
You can hide/show instances information by clicking HIDE INSTANCES at the top of the list sub-view.

**Thumbnail sub-view**
The thumbnail sub-view provides you with a graphical replication of your monitored SQL servers. Access additional details in the single instance dashboard by clicking on a thumbnail's instance name or easily spot categories that have active alerts as marked by a critical icon. Navigate to an instance’s specific category view in case it has active alerts by clicking on the marked category icon.
The thumbnail sub-view displays basic statistics about your SQL Servers:

**SQL Server version**
Displays the edition and version of SQL Server software used in this instance.

**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.

**SQL Memory Usage**
Refers to amount of memory in use by the monitored SQL Server instance.

**SQL Disk I/O**
Refers to the number of physical reads and physical writes made by the SQL Server instance between refreshes.

**Category icons and actions**
In the thumbnail sub-view, use your mouse cursor to hover over a category icon in the left of the thumbnail and SQLdm displays two linkable options: **Show Alert** and **Go To Category**.

If the category icon contains active alerts, you can choose the **Show Alert** link to access the Alerts Detail screen. If there are no active alerts, you can click a category icon and you are directed to a specific category view in the single instance dashboard.

The table below shows the relationship between the category icons and their corresponding views:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Sessions Icon]</td>
<td>Sessions tab of the single instance dashboard</td>
</tr>
<tr>
<td>![Queries Icon]</td>
<td>Queries tab of the single instance dashboard</td>
</tr>
<tr>
<td>![Resources Icon]</td>
<td>Resources tab of the single instance dashboard</td>
</tr>
</tbody>
</table>
Heatmap sub-view

The heatmap sub-view provides you with a visual overall status of your SQL Server environment. Through its color scheme, you can easily identify which of your monitored instances need the most attention. From each heatmap, you can access basic statistics and specific instance category views.

The following colors in the heatmap sub-view are associated with a status and action within SQLdm:

<table>
<thead>
<tr>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Acceptable threshold where SQLdm does not generate an alert.</td>
</tr>
<tr>
<td>Yellow</td>
<td>Warning threshold where SQLdm generates a warning alert.</td>
</tr>
<tr>
<td>Red</td>
<td>Critical threshold where SQLdm generates a critical alert.</td>
</tr>
<tr>
<td>Orange</td>
<td>Monitored SQL Server in maintenance mode where SQLdm does not generate an alert.</td>
</tr>
</tbody>
</table>

To access basic statistics about your monitored SQL Servers, first expand the heatmap of your choice by clicking the instance name. The information displayed includes:

- **SQL Server version**
  - Displays the edition and version of SQL Server software used in this instance.

- **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.

- **SQL Memory Usage**
  - Refers to amount of memory in use by the monitored SQL Server instance.

- **SQL Disk I/O**
  - Refers to the number of physical reads and physical writes made by the SQL Server instance between refreshes.

**Category icons and actions**

In the heatmap sub-view, use your mouse cursor to hover over a category icon in the left and SQLdm displays two linkable options: **Show Alert** and **Go To Category**.
If the category icon contains active alerts, you can choose the **Show Alert** link to access the **Alerts Detail screen**. If there are no active alerts, you can click a category icon and you are directed to a specific category view in the single instance dashboard.

The table below shows the relationship between the category icons and their corresponding views in the heatmap sub-view:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Sessions" /></td>
<td>Sessions tab of the single instance dashboard</td>
</tr>
<tr>
<td><img src="image" alt="Queries" /></td>
<td>Queries tab of the single instance dashboard</td>
</tr>
<tr>
<td><img src="image" alt="Resources" /></td>
<td>Resources tab of the single instance dashboard</td>
</tr>
<tr>
<td><img src="image" alt="Databases" /></td>
<td>Databases tab of the single instance dashboard</td>
</tr>
<tr>
<td><img src="image" alt="Alerts Detail" /></td>
<td>Alerts Detail screen</td>
</tr>
<tr>
<td><img src="image" alt="Overview" /></td>
<td>Overview tab of the single instance dashboard</td>
</tr>
</tbody>
</table>

Instances in heatmap sub-view display according to severity.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more](#)

---

**View active alerts in your SQL Server environment**

The Active Alerts panel provides you with up to date information on all the active alerts in your SQL Server environment. The panel allows you to see if more than one alert is active at the time of receiving an alert notification, making it easier to determine the underlying problem. For example, analyzing a group of poor response time and CPU alerts, you can deduce that the high CPU usage is affecting your environment and triggering more alerts.

Alerts’ details include time of alert inception, alert description (summary), instance name, and alert category. The most recent alert always shows up at the top of the panel.
In the Active Alerts panel it’s possible to take two actions to further obtain information. You can click on any of the linked instances and the single instance dashboard displays. You can also click an individual alert or Show Details and the Alert Detail screen displays.

Columns are sortable, just click the column header of your choice.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Use widgets for tailored information

The widgets section of the dashboard provides you with practical access to specific environment information for your day-to-day operations. Click the image to view full size.

This section is comprised of the following widgets:

- Top servers by alert count
- Top servers by response time
- Top databases by alerts
- Top alert categories

Top servers by alert count

The Top servers by alert count widget lists the instances with the highest number of alerts in descending order. This widget displays the following information:

- Instance name
- Alert count

If you click any instance name of a row, the Databases tab of the single instance dashboard displays.

Top servers by response time

The Top servers by response time widget lists the instances with the highest response time in descending order. This widget displays the following information:

- Instance name
- Response time in milliseconds (ms)
If you click any instance name of a row, the Sessions tab of the single instance dashboard displays.

### Top databases by alerts

The Top databases by alerts widget lists the instances that contain the databases with the highest number of alerts in descending order. This widget displays the following information:

- Database name
- Number of alerts

If you click any database name of a row, the Databases tab of the single instance dashboard displays.

### Top alert categories

The Top alert categories widget lists the category with the largest number of alerts and displays the following information:

- Category name
- Number of alerts

Alerts' categories include:

- Custom
- Databases
- Logs
- Operational
- Queries
- Resources
- Services
- Sessions
- Virtualization

If you click any category name, the Alerts Detail screen displays.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>
Navigate the web console single instance dashboard

The single instance dashboard includes an overview section for the selected monitored SQL server and specific category views. SQLdm users are able to obtain details on sessions, queries, resources, and databases. Click image to view full size.

In the single instance dashboard, you can find the following sections:

- Server status
- Server properties
- Service status
- File use
- Databases roll up view
- Active alerts roll up view
- Charts
- Category views

Server status

In this section view the status of your monitored SQL Server and its alerts.

Server properties

In this section view the most important characteristics of your monitored SQL Server instance.

- **Instance Name**
  Displays the name of the monitored SQL Server instance.

- **SQL Version**
  Indicates the current SQL Server version including service pack and build information on this SQL Server instance.

- **SQL Edition**
  Indicates the current SQL Server edition, including Enterprise, Business Intelligence, or Standard, installed on this SQL Server instance.

- **Running since**
  Indicates the timestamp associated with the most recent restart of this SQL Server instance.

- **Clustered**
  Indicates whether this SQL Server instance is part of a cluster. If it is part of a cluster, SQLdm displays the name of the active node.

- **Processors**
  Indicates the total number of processors associated with this SQL Server instance and how many of these processors are currently in use.

- **Host**
  Indicates the name of the server hosting this SQL Server instance.

- **Host OS**
  Indicates the current operating system on the server hosting this SQL Server instance.
Host Memory (GB)
Indicates the total amount of memory on the server hosting this SQL Server instance.

Service status
In this section view the status of the SQL Server services:
- SQL Server Service
- SQL Agent Service
- Distributed Transaction Coordinator (DTC) Service

File Use
In this section view file use for your monitored SQL Server instance.

Data File (Allocated GB)
Displays the allocated data space in GB for this SQL Server instance.

Data File (Used GB)
Displays the used data space in GB for this SQL Server instance.

Log File (Allocated GB)
Indicates the allocated log space in GB on the SQL Server instance.

Log File (Used GB)
Indicates the used log space in GB on the SQL Server instance.

Database count
Indicates the number of databases contained in the monitored SQL Server.

Databases roll up view
The Databases roll up view shows the monitored databases for the selected instance. Information provided includes:

Database name
Displays the database name.

Creation Date
Displays database creation date and time.

Data Used (MB)
Displays the size of all data files used on this SQL Server instance.

Log Size (MB)
Indicates the current size of the log files on this SQL Server instance.

Status
Indicates the operational status of the database such as Normal, Offline, or Suspect.

If you click any row of the Database roll up view, SQLdm directs you to the database tab of the single instance dashboard for more details.

To sort information in the Databases roll up view, click any of the column headers.

Active alerts roll up view
The Active alerts roll up view shows the active alerts for the selected instance only. Information provided includes:

- Date and time of alert inception
- Alert description
- Monitored SQL Server instance name
- Database name
- Alert category
If you click any row of the Active alerts roll up view, the Alerts Detail Screen displays.

To sort information in the Active alerts roll up view, click the column header of your choice.

Charts

The single instance dashboard includes the following charts:

Response Time

The Response Time chart displays the response time (in milliseconds) it takes a Select SQL statement to go to the SQL Server instance and back. This chart indicates the current speed and congestion of the network as well as the speed with which SQL Server processes small queries. To obtain additional key diagnostic statistics for sessions on your SQL Server instance, click on the chart and you’re directed to the Sessions > Overview view. This view contains various charts like the Response Time (ms), Session Activity, and Blocked Sessions.

SQL Server CPU Activity

The SQL Server CPU Activity chart displays the percentage of a CPU consumed by your SQL Server, virtual machine, and host server. To obtain additional key diagnostic statistics for monitoring the resources for your SQL Server instance and computer on which it resides, click on the chart and you’re directed to the Resources > CPU view. This view contains various charts like the CPU Usage (%), Processor Time (%), and Processor Queue Length.

OS Memory Usage

The OS Memory Usage chart displays information in GB on the physical and virtual memory in use during the current day. To obtain additional key memory statistics on the computer hosting your SQL Server instance, click on the chart and you’re directed to the Resources > Memory view. This view contains various charts like SQL Memory Usage (MB), Memory Areas (MB), and Page Life Expectancy (sec).

Disk Busy

The Disk Busy chart displays the percentage of time during the last ten minutes that the disk was not idle for all the disks (combined) on the server that the SQL Server instance resides. To obtain additional key statistics on the way your disks are used on the computer hosting your SQL Server instance, click on the chart and you’re directed to the Resources > Disk view. This view contains various charts like the Disk Reads/Second per Disk, Disk Writes/Second per Disk, and the SQL Server Physical I/O.

Category views

The single instance dashboard contains the Sessions, Queries, Query Waits, Resources, Databases, and Alerts tabs where you can see metrics and information on these category views.
From these tabs you can:

- View your SQL Server sessions information.
- View your SQL Server queries information.
- View your SQL Server query waits information.
- View your SQL Server resources information.
- View your SQL Server databases information.
- View your SQL Server alerts.

Access the single instance dashboard

SQLDM provides several paths to access the single instance dashboard. The first access path is by clicking the appropriate monitored SQL Server instance in the list, thumbnail, or heatmap sub-views. The second access path is by clicking on a specific instance from the Active Alerts panel, or the Most Critical Instances view in the main dashboard. It is also possible to access the single instance dashboard by clicking a specific instance or database in the Top servers by alert count, Top servers by response time, and Top databases by alerts widgets.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

View your SQL Server sessions information

The Sessions tab of the single instance dashboard allows you to quickly view key diagnostic statistics for sessions on your SQL Server instance and contains the Overview and Session Graphs views. Click image to view full size.
The **Sessions > Overview** tab contains the following information:

- Session ID
- Host
- Database
- Session status
- Open transactions
- Command
- Application
- Wait time (ms)
- Wait type
- CPU (ms)
- CPU Delta (ms)
- Memory Usage (KB)
- Login time
- Last Batch
- Blocked by
- Blocking
- Blocking count
- Net Address
- Net Library
- Version Store Elapsed (sec)

**Session Graphs**

The **Sessions > Sessions Graphs** tab contains the following charts:

**Response Time (ms)**

The Response Time chart displays the response time (in milliseconds) it takes a Select SQL statement to go to the SQL Server instance and back. This chart indicates the current speed and congestion of the network as well as the speed with which SQL Server processes small queries.

**Session Activity**

The Session Activity 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.

**Blocked Sessions**

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

**Access the Sessions tab**

SQLdm provides three paths to access the Sessions tab. The first access path is by clicking on the Response Time chart (ms) of the single instance dashboard, and you're directed to the **Sessions > Overview** tab. The second access path is by selecting the Sessions tab of the single instance dashboard. You can also click on an alert that belongs to the Sessions category in the Active alerts panel of the main dashboard.

**View your SQL Server queries information**

The Queries tab of the single instance dashboard lists each event that occurs in the Query Monitor and allows you to easily explore query data in multiple ways and get a handle of query performance in your environment.

The Queries tab provides a highly usable and customizable Main Query view along with two low-level views which you can navigate to for root cause analysis:

- Query Signature view
- Query Details view
The Main Query view consists of the following major components that allow you to customize your view:

- View selection and grouping
- Filtering
- Time period
- Data view

Select the type of view that you want displayed

The View Selection of the Main Query view allows you to choose from a variety of views to explore query data. The following views are available:

- Duration (ms)
- CPU Time (ms)
- Reads
- Writes
- I/O (sum of Reads and Writes)
- Blocking Duration (ms)
- Wait Duration (ms)
- Deadlocks

Duration (ms) is the default view of the Main Query view.

Use Group By to organize your view

Once you have selected a view, use the Group By option to choose from the following type of objects for query aggregation:

- Application Name
- Database Name
- User
- Client
- Query Signature
- Query Statement
Use Filtering to customize your view

Filter options in the Main Query view allow you to focus on only the information that you want displayed. You can include or exclude specific applications, databases, users, SQL text and more from the Query Monitor results through filters. The following filter options are available:

- Application
- Databases
- Users
- Clients
- SQL text
- Advanced Filters

Select a time period to display query data

You can choose the time period to display in the Main Query view. This selection limits the queries to only those that run during the specified time period. You can select from a broad range of built-in time periods or specify start and end dates, and times for the period. The **Time** box includes the following options:

- Last 30 minutes
- Last 1 hour
- Last 8 hours
- Last 24 Hours
- Last 7 days
- Last 30 days
- Custom

By default only queries that start within the time period are shown. If you want to modify this behavior, the Advanced Filters option lets you include incomplete queries or overlapping queries. For more information, see Advanced filters of the main query view.

To switch to a shorter time frame, click the larger time bucket. For example, clicking a month in a monthly graph modifies the time period to the selected month and grouping to days.

Analyze your query data

On the right hand side of the overall view, query data is displayed based on your View, Group By, and filtering selections. The Data view is designed to provide good visualization of the selected data and enable you to drill down into problem areas. The Data view is composed of the following elements:

Header

Provides View, Group By, Period, and filter criteria information.
Graph
Chart area which displays the aggregated query results that meet the filtering criteria.

List
Displays data selected for the view in list form. There are several list options according to your Group By selection in the Main Query view.

The table below shows what options you have to drill down from the graph or list and get more specific information:

<table>
<thead>
<tr>
<th>Click the...</th>
<th>Access the...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Name</td>
<td>Query Signature view</td>
</tr>
<tr>
<td>Database Name</td>
<td>Query Signature view</td>
</tr>
<tr>
<td>User</td>
<td>Query Signature view</td>
</tr>
<tr>
<td>Client</td>
<td>Query Signature view</td>
</tr>
<tr>
<td>Query Signature</td>
<td>Query Statement view</td>
</tr>
<tr>
<td>Query Statement</td>
<td>No drill down option</td>
</tr>
</tbody>
</table>

You can return to the higher level view by clicking by the list name or by using the list context menu's Back option.

**Access the Queries tab**

SQLdm provides two paths to access the Queries tab. The first access path is by selecting the Queries tab of the single instance dashboard. You can also click on the query category icon of a specific instance in the thumbnail and heatmap views.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Advanced query signature view**

SQL diagnostic manager provides the ability to view individual SQL Statements or to view Query Signatures. Query signatures are groupings of SQL Statements that match once their literals are stripped. A query signature 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 signature, you can drill into individual queries that make up the signature as needed. For more information on the query signature view, see View the query monitor signature mode.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Query Details view**

The Query Details view provides various views with detailed information on each query. Click image to view full size.

The Query Details view is comprised of the following views:
The Plan Diagram and Plan XML views are only available for instances running SQL Server 2008+.

**Plan Diagram**

The Plan Diagram is a visual representation of the query execution plan available in XML format in SQL Server. The Plan Diagram displays a view of the tree of operations that make up a query. This tree shows individual operation nodes, pertaining graphical execution plan icon, along with basic information such as operator name and operation's percentage of total cost. Click image to view full size.

In the Plan Diagram view you can take the following actions:

- Select individual operators and view the set of properties for each operator.
- Increase size of the image to view more details (Zoom in).
- Reduce image size to show more of view in window (Zoom out).
- Select a size for the view so that it fits in the current view size (Zoom to fit).
- Export image of Plan Diagram to file.

The top three operators based on percentage of total cost in a Plan Diagram are highlighted in yellow.

**Plan XML**

The Plan XML tab of the Query Details window displays the actual XML of the query execution plan. It has a syntax-highlighting (color-coded) XML viewer. Click image to view full size.

In the Plan XML view you can take the following actions:

- Export and save XML file for review in other application such as SSMS or email query execution plan to other DBAs.

**SQL Text**
The SQL Text tab of the Query Details window shows the underlying SQL Text for the query execution plan. In the SQL Text view you can take the following actions:

- Export and save SQL text for review.

**Query Columns**

The Query Columns tab of the Query Details window shows all the referenced columns for the query execution plan. In the Query Columns view you can take the following actions:

- Export and save information to a text file.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

---

**Plan diagram operator properties**

In the Plan Diagram view, hover over an operator to view its full set of properties such as:

- Operator Type
- Operator Description
- List of Properties (varies according to operator type).
- Related Objects (varies according to operator type).

---

**Execution plan graph operators table**

Operators describe how SQL Server executes a query or a Data Manipulation Language (DML) statement. The query optimizer uses operators to build a query plan to create the result specified in the query, or to perform the operation specified in the DML statement. The query plan is a tree consisting of physical operators.

This is the table of descriptions to be used in the properties dialog showing the operator details in the execution plan graphical view.

---
<table>
<thead>
<tr>
<th>Operator</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate</td>
<td>Logical</td>
<td>Computes new value using SQL Server functions MIN, MAX, SUM, COUNT or AVG.</td>
</tr>
<tr>
<td>Arithmetic Expression</td>
<td>Logical</td>
<td>Computes a new value from existing values in a row. Not used in SQL Server 2014.</td>
</tr>
<tr>
<td>Assert</td>
<td>Physical</td>
<td>Verifies a condition.</td>
</tr>
<tr>
<td>Assign</td>
<td>Physical</td>
<td>Assigns a value to a variable.</td>
</tr>
<tr>
<td>Async Concat</td>
<td>None</td>
<td>Used in remote distributed queries to get output rows from remote child nodes to send to the parent node.</td>
</tr>
<tr>
<td>Bitmap</td>
<td>Physical</td>
<td>Applies filtering to parallel query plans.</td>
</tr>
<tr>
<td>Bitmap Create</td>
<td>Logical</td>
<td>Shows where bitmaps are built.</td>
</tr>
<tr>
<td>Bookmark Lookup</td>
<td>None</td>
<td>Uses a bookmark (row ID or clustering key) to look up the corresponding row in the table or clustered index. Not used in SQL Server 2014. The Key Lookup operator also provides this functionality.</td>
</tr>
<tr>
<td>Branch Repartition</td>
<td>Physical</td>
<td>Shows where iterators could be executed by parallel threads.</td>
</tr>
<tr>
<td>Broadcast</td>
<td>None</td>
<td>Sends the set of input rows to multiple consumers.</td>
</tr>
<tr>
<td>Build Hash</td>
<td>None</td>
<td>Indicates that a batch has been built.</td>
</tr>
<tr>
<td>Cache</td>
<td>Logical</td>
<td>Caches a single row of data. Not used in SQL Server 2014.</td>
</tr>
<tr>
<td>Clustered Index Delete</td>
<td>Physical</td>
<td>Deletes rows from a clustered index.</td>
</tr>
<tr>
<td>Clustered Index Insert</td>
<td>Physical</td>
<td>Inserts rows into a clustered index.</td>
</tr>
<tr>
<td>Clustered Index Merge</td>
<td>Physical</td>
<td>Merges a data stream and a clustered index.</td>
</tr>
<tr>
<td>Clustered Index Scan</td>
<td>Physical</td>
<td>Scans a clustered index. Returns only rows matching the WHERE clause and sorts the results based on the ORDERED clause.</td>
</tr>
<tr>
<td>Clustered Index Seek</td>
<td>None</td>
<td>Retrieves rows from a clustered index using the seeking ability of indexes.</td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Clustered Index Update</td>
<td>The <strong>Clustered Index Update</strong> operator updates rows in the clustered index. It only updates rows matching the WHERE clause.</td>
<td></td>
</tr>
<tr>
<td>Collapse</td>
<td>The <strong>Collapse</strong> operator merges separate operations into a single more efficient operation.</td>
<td></td>
</tr>
<tr>
<td>Columnstore Index Scan</td>
<td>The <strong>Columnstore Index Scan</strong> operator scans the columnstore index specified in the query execution plan argument.</td>
<td></td>
</tr>
<tr>
<td>Compute Scalar</td>
<td>The <strong>Compute Scalar</strong> operator returns a computer scalar value from an evaluated expression.</td>
<td></td>
</tr>
<tr>
<td>Concatenation</td>
<td>The <strong>Concatenation</strong> operator returns rows scanned from multiple inputs. This is used for statements like UNION ALL.</td>
<td></td>
</tr>
<tr>
<td>Constant Scan</td>
<td>The <strong>Constant Scan</strong> operator adds rows into a query.</td>
<td></td>
</tr>
<tr>
<td>Convert</td>
<td>The <strong>Convert</strong> operator converts one scalar type to another.</td>
<td></td>
</tr>
<tr>
<td>Cross Join</td>
<td>The <strong>Cross Join</strong> operator is a logical operator that joins each row from input with each row from another input.</td>
<td></td>
</tr>
<tr>
<td>catchall</td>
<td>The <strong>catchall</strong> operator is a placeholder icon when an operator does not match any other query operators.</td>
<td></td>
</tr>
<tr>
<td>Cursor</td>
<td>The <strong>Cursor</strong> operator describes the execution of a query or update that use cursor operations.</td>
<td></td>
</tr>
<tr>
<td>Declare</td>
<td>The <strong>Declare</strong> operator allocates a local variable.</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>The <strong>Delete</strong> operator deletes rows from an object.</td>
<td></td>
</tr>
<tr>
<td>Deleted Scan</td>
<td>The <strong>Deleted Scan</strong> operator scans the deleted table within a trigger.</td>
<td></td>
</tr>
<tr>
<td>Distinct</td>
<td>The <strong>Distinct</strong> operator removes duplicate rows.</td>
<td></td>
</tr>
<tr>
<td>Distinct Sort</td>
<td>The <strong>Distinct Sort</strong> operator removes duplicate rows and sorts the resulting set of rows.</td>
<td></td>
</tr>
<tr>
<td>Distribute Streams</td>
<td>The <strong>Distribute Streams</strong> operator is used to break records from a single input to multiple output streams in a parallel query plan.</td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>The <strong>Dynamic</strong> operator uses a cursor to see all changes made by others.</td>
<td></td>
</tr>
<tr>
<td>Eager Spool</td>
<td>The <strong>Eager Spool</strong> operator stores each row in the input to tempdb so that this cached data can be used if the operator is rewound.</td>
<td></td>
</tr>
<tr>
<td>Fetch Query</td>
<td>The <strong>Fetch Query</strong> operator returns rows from a fetch issued against a cursor.</td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>Result</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Filter</td>
<td>The <strong>Filter</strong> operator returns only rows that match a filter expression.</td>
<td></td>
</tr>
<tr>
<td>Flow Distinct</td>
<td>None</td>
<td>The <strong>Flow Distinct</strong> operator removes duplicate rows but returns each row as it is processed.</td>
</tr>
<tr>
<td>Full Outer Join</td>
<td>None</td>
<td>The <strong>Full Outer Join</strong> operator is a logical operator that implements an OUTER JOIN. Results in all matching rows from two streams plus rows for each row in the streams that did not have matches.</td>
</tr>
<tr>
<td>Gather Streams</td>
<td></td>
<td>The <strong>Gather Streams</strong> operator is used to consume multiple input streams and combine them into a single output stream in parallel query plans.</td>
</tr>
<tr>
<td>Hash Match</td>
<td></td>
<td>The <strong>Hash Match</strong> operator creates a hash table from the rows in the build input.</td>
</tr>
<tr>
<td>Inner Join</td>
<td>None</td>
<td>The <strong>Inner Join</strong> operator returns rows that satisfy the JOIN of the first (top) input with the second (bottom) input.</td>
</tr>
<tr>
<td>Insert</td>
<td></td>
<td>The <strong>Insert</strong> logical operator inserts input rows into the object specified in the Argument column.</td>
</tr>
<tr>
<td>Inserted Scan</td>
<td></td>
<td>The <strong>Inserted Scan</strong> operator scans the inserted table.</td>
</tr>
<tr>
<td>Intrinsic</td>
<td></td>
<td>The <strong>Intrinsic</strong> operator runs an internal Transact-SQL function.</td>
</tr>
<tr>
<td>Iterator</td>
<td></td>
<td>The <strong>Iterator</strong> operator is a placeholder icon used when no match is found for the iterator operation.</td>
</tr>
<tr>
<td>Key Lookup</td>
<td></td>
<td>The <strong>Key Lookup</strong> operator signals a lookup using a bookmark on a table with a clustered index.</td>
</tr>
<tr>
<td>Keyset</td>
<td></td>
<td>The <strong>Keyset</strong> operator uses a cursor that can only see updates and not inserts.</td>
</tr>
<tr>
<td>Language Element</td>
<td></td>
<td>The <strong>Language Element</strong> operator is a placeholder icon when no matching language constructs can be found.</td>
</tr>
<tr>
<td>Lazy Spool</td>
<td></td>
<td>The <strong>Lazy Spool</strong> operator stores each row in the input to tempdb so that this cached data can be used if the operator is rewound. The rows are only copied to tempdb as they are processed.</td>
</tr>
<tr>
<td>Left Anti Semi Join</td>
<td>None</td>
<td>The <strong>LEFT ANTI SEMI JOIN</strong> operator returns rows from the first input where there is no match in the second input.</td>
</tr>
<tr>
<td>Left Outer Join</td>
<td>None</td>
<td>The <strong>LEFT OUTER JOIN</strong> operator returns rows from the first input that satisfy the join with the second input.</td>
</tr>
<tr>
<td>Left Semi Join</td>
<td>None</td>
<td>The <strong>LEFT SEMI JOIN</strong> operator returns rows from the first input that have a match with the second input.</td>
</tr>
<tr>
<td>Log Row Scan</td>
<td></td>
<td>The <strong>Log Row Scan</strong> operator scans the transaction log.</td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Merge Interval</td>
<td>The MERGE INTERVAL operator merges multiple intervals to create a minimal set of non-overlapping intervals.</td>
<td></td>
</tr>
<tr>
<td>Merge Join</td>
<td>The MERGE JOIN operator performs one of the many possible join operations.</td>
<td></td>
</tr>
<tr>
<td>Nested Loops</td>
<td>The NESTED LOOPS operator performs the logical operations to satisfy many join operations that need a loop to search the inner table for rows in the outer table.</td>
<td></td>
</tr>
<tr>
<td>Nonclustered Index Delete</td>
<td>The NONCLUSTERED INDEX DELETE operator deletes rows from a non-clustered index.</td>
<td></td>
</tr>
<tr>
<td>Index Insert</td>
<td>The INDEX INSERT operator inserts rows into a non-clustered index.</td>
<td></td>
</tr>
<tr>
<td>Index Scan</td>
<td>The INDEX SCAN operator scans a non-clustered index. It returns only rows matching the WHERE clause.</td>
<td></td>
</tr>
<tr>
<td>Index Seek</td>
<td>The INDEX SEEK operator retrieves rows from a non-clustered index using the seeking ability of indexes.</td>
<td></td>
</tr>
<tr>
<td>Index Spool</td>
<td>The INDEX SPOOL operator copies input rows to tempdb and builds a non-clustered index for these rows.</td>
<td></td>
</tr>
<tr>
<td>Nonclustered Index Update</td>
<td>The NONCLUSTERED INDEX UPDATE operator updates rows from its input in the non-clustered index.</td>
<td></td>
</tr>
<tr>
<td>Online Index Insert</td>
<td>The ONLINE INDEX INSERT operator is a physical operator indicating that an index create, alter or drop is performed online.</td>
<td></td>
</tr>
<tr>
<td>Parallelism</td>
<td>The PARALLELISM operator performs the logical operations of distribute, gather or repartition streams.</td>
<td></td>
</tr>
<tr>
<td>Parameter Table Scan</td>
<td>The PARAMETER TABLE SCAN operator scans a table that acts as a parameter of the query.</td>
<td></td>
</tr>
<tr>
<td>Partial Aggregate</td>
<td>The PARTIAL AGGREGATE operator is a logical operator that aggregates input rows to prevent writing to disk in parallel plans.</td>
<td></td>
</tr>
<tr>
<td>Population Query</td>
<td>The POPULATION QUERY operator populates a cursor's work table.</td>
<td></td>
</tr>
<tr>
<td>Refresh Query</td>
<td>The REFRESH QUERY operator fetches current data for rows.</td>
<td></td>
</tr>
<tr>
<td>Remote Delete</td>
<td>The REMOTE DELETE operator deletes rows from an remote object.</td>
<td></td>
</tr>
<tr>
<td>Remote Index Scan</td>
<td>The REMOTE INDEX SCAN operator scans a remote index.</td>
<td></td>
</tr>
<tr>
<td>Remote Index Seek</td>
<td>The REMOTE INDEX SEEK operator retrieves rows using a remote index.</td>
<td></td>
</tr>
<tr>
<td>Remote Insert</td>
<td>The REMOTE INSERT operator inserts rows into a remote object.</td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Remote Query</td>
<td>The REMOTE QUERY operator submits a query to a remote source.</td>
<td></td>
</tr>
<tr>
<td>Remote Scan</td>
<td>The REMOTE SCAN operator scans a remote object.</td>
<td></td>
</tr>
<tr>
<td>Remote Update</td>
<td>The REMOTE UPDATE operator updates a remote object.</td>
<td></td>
</tr>
<tr>
<td>Repartition Streams</td>
<td>The REPARTITION STREAMS operator creates multiple out streams from multiple input streams while applying a bitmap filter.</td>
<td></td>
</tr>
<tr>
<td>Result</td>
<td>The RESULT operator is the query plan return data.</td>
<td></td>
</tr>
<tr>
<td>RID Lookup</td>
<td>The RID LOOKUP operator is a bookmark lookup on a heap.</td>
<td></td>
</tr>
<tr>
<td>Right Anti Semi Join</td>
<td>The RIGHT ANTI SEMI JOIN operator returns rows from the second input where there is no match in the first input.</td>
<td></td>
</tr>
<tr>
<td>Right Outer Join</td>
<td>The RIGHT OUTER JOIN operator returns rows from the second input that satisfy the join with the first input.</td>
<td></td>
</tr>
<tr>
<td>Right Semi Join</td>
<td>The RIGHT SEMI JOIN operator returns rows from the second input that have a match with the first input.</td>
<td></td>
</tr>
<tr>
<td>Row Count Spool</td>
<td>The ROW COUNT SPOOL operator returns empty rows for each row in the input stream.</td>
<td></td>
</tr>
<tr>
<td>Segment</td>
<td>The SEGMENT operator uses the value of columns to divide the input set into segments.</td>
<td></td>
</tr>
<tr>
<td>Segment Repartition</td>
<td>The SEGMENT REPARTITION operator marks the boundaries of regions whose iterators can be run in parallel threads.</td>
<td></td>
</tr>
<tr>
<td>Sequence</td>
<td>The SEQUENCE operator executes each input in sequence.</td>
<td></td>
</tr>
<tr>
<td>Sequence Project</td>
<td>The SEQUENCE PROJECT operator adds columns to the input set, divides the input set into segments and outputs one segment at a time.</td>
<td></td>
</tr>
<tr>
<td>Snapshot</td>
<td>The SNAPSHOT operator creates a cursor that can see changes by others.</td>
<td></td>
</tr>
<tr>
<td>Sort</td>
<td>The SORT operator sorts incoming rows.</td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>The SPLIT operator creates a delete and insert operation out of each update operation.</td>
<td></td>
</tr>
<tr>
<td>Spool</td>
<td>The SPOOL operator saves an intermediate query to tempdb.</td>
<td></td>
</tr>
<tr>
<td>Stream Aggregate</td>
<td>The STREAM AGGREGATE operator groups rows by columns and calculates aggregate expressions.</td>
<td></td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Switch</td>
<td>The SWITCH operator copies the appropriate input stream to the output stream by evaluating an expression.</td>
<td></td>
</tr>
<tr>
<td>Table Delete</td>
<td>The TABLE DELETE operator deletes rows from a table.</td>
<td></td>
</tr>
<tr>
<td>Table Insert</td>
<td>The TABLE INSERT operator inserts rows into a table.</td>
<td></td>
</tr>
<tr>
<td>Table Merge</td>
<td>The TABLE MERGE operator applies a merge data stream to a heap.</td>
<td></td>
</tr>
<tr>
<td>Table Scan</td>
<td>The TABLE SCAN operator retrieves rows from a table.</td>
<td></td>
</tr>
<tr>
<td>Table Spool</td>
<td>The TABLE SPOOL operator scans the input and places the rows into tempdb.</td>
<td></td>
</tr>
<tr>
<td>Table Update</td>
<td>The TABLE UPDATE operator updates rows in a table.</td>
<td></td>
</tr>
<tr>
<td>Table-valued Function</td>
<td>The TABLE-VALUED FUNCTION operator evaluates a table-valued function and stores the resulting rows in tempdb.</td>
<td></td>
</tr>
<tr>
<td>Top</td>
<td>The TOP operator returns only the specified number of rows from the input.</td>
<td></td>
</tr>
<tr>
<td>Top N Sort</td>
<td>The TOP N SORT operator returns only the specified number of rows from the input and sorts them.</td>
<td></td>
</tr>
<tr>
<td>UDX</td>
<td>The UDX operator implements XQuery and XPath operations.</td>
<td></td>
</tr>
<tr>
<td>Union</td>
<td>The UNION operator combines multiple inputs and removes duplicates.</td>
<td></td>
</tr>
<tr>
<td>Update</td>
<td>The UPDATE operator updates a specified object from the rows in its input.</td>
<td></td>
</tr>
<tr>
<td>While</td>
<td>The WHILE operator represents a SQL while loop.</td>
<td></td>
</tr>
<tr>
<td>Window Spool</td>
<td>The WINDOW SPOOL operator expands the input rows into sets of rows that represent the window associated with the row.</td>
<td></td>
</tr>
</tbody>
</table>

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Advanced filters of the Main Query view

Advanced filter options allow you to determine the precise Query Monitor data that is displayed. You can find the advanced filter options at the left side of the Main Query view. Click image to view full size.
The Advanced Filters include:

**Show SQL Statements**
Includes queries that are SQL Statements. This filter is enabled by default.

**Show Stored Procedures**
Includes queries that are Stored Procedure calls. This filter is enabled by default.

**Show SQL Batches**
Includes queries that are SQL batches. This filter is enabled by default.

**Include Overlapping Queries**
Includes queries whose total period of execution overlaps with the selected time period. This filter is disabled by default.

**Include Incomplete Queries**
Includes queries whose total period of execution exceeds the selected time period (i.e. query information is incomplete). This filter is disabled by default.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

The list view displays all matching objects for the view selection and grouping in the Main Query view. The following list options are available:

- Application list
- Database list
- User list
- Client list
- Query Signature list

**Application list**
The application list displays data that results from grouping by Application in the View Selection of the Main Query view. The application list provides you with the following information by default:

- Application
- Occurrences
- Total Duration (ms)
- Average Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Total IO
- Total Wait Time (ms)
- Most Recent Completion
- Total Blocking Time (ms)
- Total Deadlocks

You can add or remove columns through the Column Chooser under **Options**.

- Database
- Average Writes
- Average Blocking Time (ms)
- Average CPU Time (ms)
- Reads as % of list
- Average I/O
- CPU as % of list
- Average Reads
- Average CPU Per Second
• Average Deadlocks
• Average Wait Time (ms)

Records
You can display the top 100, 200 or all records by clicking the different Show Top options.

Explanation: CPU as % of List
In the Main Query view, after selecting the CPU Time (ms) by Application view. The CPU as % of list metric provides the CPU as a percentage of the values returned in the application list. For example, if you have three applications returned with a total CPU Time (ms) of 600, then the CPU as % of list metric for each is calculated as follows:

Application 1: CPU Time (ms): 100; \((100/600) \times 100 = 16.7\%\) of total application CPU.
Application 2: CPU Time (ms): 200; \((200/600) \times 100 = 33\%\) of total application CPU.
Application 3: CPU Time (ms): 300; \((300/600) \times 100 = 50\%\) of total application CPU.

The CPU as % of list metric helps to pinpoint which application in your list has the highest CPU Time percentage (Application 3).

To drill down for additional query information, you can click the gear icon in the list, and the following context menu displays:

• Query Signature. Click this option to view query signatures for the selected application.
• Query Statement. Click this option to view query statements for the selected application.

Database list
The database list displays data that results from grouping by Database in the View Selection of the Main Query view. The database list provides you with the following information by default:

• Database
• Occurrences
• Total Duration (ms)
• Average Duration (ms)
• Total CPU Time (ms)
• Total Reads
• Total Writes
• Total IO
• Average Wait Time (ms)
• Most Recent Completion
• Average Blocking Time (ms)
• Average Deadlocks
Additional columns are available through the Column Chooser under Options.

- Average Writes
- Total Blocking Time (ms)
- Average CPU Time (ms)
- Reads as % of list
- Total Wait Time (ms)
- Average IO
- Total Deadlocks
- CPU as % of list
- Average Reads
- Average CPU Per Second
- Application

To drill down for additional query information, you can click the gear icon in the list and the following context menu displays:

- **Query Signature.** Click this option to view query signatures for the selected database.
- **Query Statement.** Click this option to view query statements for the selected database.

**Records**
You can display the top 100, 200 or all records by clicking the different Show Top options.

To access the Query Signature view, you can also click the linkable database name in the database list.

**User list**

The user list displays data that results from grouping by user in the View Selection of the Main Query view. The user list provides you with the following information by default:

- Users
- Occurrences
- Total Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Average Wait Time (ms)
- Most Recent Completion
- Average Blocking Time (ms)
- Average Deadlocks

Additional columns are available through the Column Chooser under Options.

- Database
- Average Writes
- Total Blocking Time (ms)
- Average Duration (ms)
- Average CPU Time (ms)
- Reads as % of list
- Total Wait Time (ms)
- Average I/O Per Second
- Total Deadlocks
- CPU as % of list
- Average Reads
- Average CPU Per Second

To drill down for additional query information, you can click the gear icon in the list, and the following context menu displays:

- **Query Signature.** Click this option to view query signatures for the selected user.
- **Query Statement.** Click this option to view query statements for the selected user.

**Records**
You can display the top 100, 200 or all records by clicking the different Show Top options.

To access the Query Signature view, you can also click the linkable user name in the user list.
Client list

The client list is the data displayed after you selected Group By Clients in the View Selection of the Main Query view. The client list provides you with the following information by default:

- Clients
- Occurrences
- Total Duration (ms)
- Average Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Average Wait Time (ms)
- Most Recent Completion
- Average Blocking time (ms)
- Average Deadlocks

Additional columns are available through the Column Chooser under Options.

- Database
- Average Writes
- Total Blocking Time (ms)
- Average CPU Time (ms)
- Reads as % of list
- Total Wait Time (ms)
- Average I/O Per Second
- Total Deadlocks
- CPU as % of list
- Average Reads
- Average CPU Per Second
- Application

To drill down for additional query information, you can click the gear icon in the list, and the following context menu displays:

- **Query Signature.** Click this option to view query signatures for the selected client.
- **Query Statement.** Click this option to view query statements for the selected client.

**Records**

You can display the top 100, 200 or all records by clicking the different Show Top options.

To access the Query Signature view, you can also click the linkable client name in the client list.

Query Signature list

The Query signature list is the data displayed after you selected Group By Query Signature in the View Selection of the Main Query view. The Query signature list provides you with the following information by default:

- Query ID
- Signature SQL Text
- Occurrences
- Event Type
- Total Duration (ms)
- Total CPU Time (ms)
- Total Reads
- Total Writes
- Average Wait Time (ms)
- Most Recent Completion
- Average Blocking Time (ms)
- Average Deadlocks
- Keep Detailed History Flag

Additional columns are available through the Column Chooser under Options.
To research for additional query details, you can perform the following actions from this list:

- Show Query Text
- Show Query History
- Keep Detailed History
- Print Export to Excel

View your SQL Server query waits information

The Query waits tab of the single instance dashboard allows you to analyze waits over time and by duration, so that 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.

The *Statements by Duration* and *Statements by Wait Time* charts display by default. A drop-down menu allows you to further investigate waits based on Application, Databases, Clients, Sessions, and Users.

For more information on the query waits, see View query waits.

View your SQL Server resources information

The Resources tab of the single instance dashboard contains several views such as CPU, Memory, Disk, and Server Waits. The Resources tab allows you to monitor the system resources on the computer hosting the SQL Server instance as well as what is used by SQL Server.

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

**CPU view**

The Resources > CPU tab contains the following charts:
SQL Server CPU Usage (%)

The SQL Server 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.

Processor Time (%)

The Processor Time chart displays the percentage of CPU time used by all processes on the computer hosting the SQL Server instance.

Processor Queue Length

The Processor Queue Length chart displays the number of ready threads in the processor queue on the computer hosting the SQL Server instance.

Memory view

The Resources > Memory tab contains the following charts:
SQL Memory Usage (MB)

This chart displays the SQL memory usage (SQL Used) compared to what is allocated (SQL Allocated) and the total used (Total Used).

Memory Areas (MB)

This chart displays the memory usage of the Procedure Cache, Connections, Locks, and Database.

Page Life Expectancy (sec)

This chart provides the time in seconds that a page remains in the cache or buffer pool.

Cache Hit Ratios (%)

This chart displays the hit ratios of the Buffer and Procedure cache.

*Disk view*

The Resources > Disk tab contains the following charts:
Disk Reads/Second Per Disk
This chart displays the number of reads per disk per second during a sample interval.

Disk Writes/Second Per Disk
This chart displays the number of writes per disk per second during a sample interval.

SQL Server Physical I/O
This chart displays the number of read and write requests sent to the physical disk the SQL Server instance resides in. Read and write requests are classified in Checkpoint Writers, Read Ahead Reads, Page Writes, Lazy Writer Writes, and Page Reads.

Server Waits view
The Resources > Server Waits tab contains two charts that display information on all waits occurring on your system and show the time different wait types occurred and the length of time (in milliseconds) it took for the waits to clear.

You can select from the following wait categories in the drop-down menu:

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.

**Server Waits panel**

Below the charts, the Server Waits panel includes information on the wait category, wait type, length of time the wait took to clear, total wait time for all the collected wait types, and a definition of the wait type.

**Available wait types**

The SQLdm web console displays information on the following wait types: I/O, Memory, Non-I/O Page Latch, Non-Page Latch, Transaction Log, and Other categories.

For additional information on Server Waits metrics, see Server Waits Panel.

**Access the Resources tab**

SQLdm provides two paths to access the Resources tab. The first access path is by selecting the Resources tab of the single instance dashboard. You can also click on the resources category icon of a specific instance in the thumbnail and heatmap sub-views.

---

**View your SQL Server databases information**

The Databases tab of the single instance dashboard contains the Summary, Tempdb, and Availability Groups views. The Databases tab allows you to view the status of all the databases on the selected SQL Server instance, including the tempdb database and availability groups. Click image to view full size.

**Summary view**

The Databases > Summary tab lists each database on the monitored SQL Server instance and provides a large variety of database statistics:

- Database name
- Status
- Date Created
- Last Backup
- Files
- Data File Size (MB)
- Data Unused (MB)
Log File Size (MB)
Log Used (MB)

**Capacity Usage chart**

You can choose to view how data or the log is used on your database (in megabytes). Click any database in the top panel and the Capacity usage chart displays. This chart provides information on the number of files, Log Used (MB), Log Size (MB), Data Unused (MB), and Data File Size (MB).

**Tempdb view**

The Databases > Tempdb tab allows you to view the status of your tempdb database on the selected SQL Server instance. The Tempdb view contains the following charts:

- Tempdb Space Used over Time (MB)
- Version Store Clean up Rate (KB/Sec)
- Tempdb Space Used by File (MB)

For information on each chart, see Get the tempdb status summary.

**Availability Groups view**

The Databases > Availability Groups allows you to monitor the availability groups in your monitored SQL Server instance. This view displays the following information for your availability groups:

- Group Name
- Replica Name
- Replica Role
- Synchronization Health
- Redo Queue
- Redo Rate
- Log Send Queue
- Log Rate
- Database Status

The Availability Groups view contains the following charts:

- Queue Size (Redo and Log)
- Redo Queue Size

For information on each chart, see Monitor AlwaysOn Availability Groups.

**Access the Databases tab**

The SQLdm web console provides two paths to access the Databases tab. The first access path is by selecting the Databases tab of the single instance dashboard. You can also click the database category icon of a specific instance in the thumbnail and heatmap views.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**View your SQL Server alerts**

The Active alerts roll up view of the single instance dashboard shows the active alerts for the selected instance only. Click image to view full size.
Information provided includes:

**Time**
Indicates date and time of alert inception.

**Summary**
Displays alert description.

**Instance**
Displays the name of the monitored SQL Server instance.

**Database**
Displays the database name.

**Category**
Indicates alert category.

**Additional options**

**Actions**
In the Alerts tab of the single instance dashboard, you can click **Show Details** to view additional information.

**Export**
In the Alerts tab of the single instance dashboard, you can export alert information in PDF, XLS, and XML.

**Show/Hide Alerts**
In the Alerts tab of the single instance dashboard, you can choose to show or hide alerts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

View your top values in the web console

The Top X Lists tab of the SQLdm web console allows you to view the top values in descending order for selected metrics across the instances and databases in your SQL Server environment. Click the image to view full size.

The Top X Lists dashboard contains 18 widgets that help you assess quickly the health and performance of your SQL Server environment. You can expand widgets, drill down for details, and also configure them as follows:

**Records**
Choose to display a particular number of records with a maximum of 50. Access this option through the gear icon on the right of the widget.
On the **Top X Lists dashboard** you can find the following widgets:

### Top servers by alerts

This widget allows you to view the instances with the highest number of alerts during the past seven days in your SQL Server environment. This widget displays the following information:

- Instance name
- Alerts

Clicking a specific server (instance name) in the widget allows you to navigate to the **Overview** tab of the single instance dashboard for that server.

### Top servers by CPU usage

This widget allows you to view the instances with the highest percentage of CPU usage in your SQL Server environment. This widget displays the following information:

- Instance name
- CPU usage in percentage

Clicking a specific instance name in the widget allows you to navigate to the **Resources > CPU** tab of the single instance dashboard for that server.

### Top servers by memory usage

This widget provides you with information on the top instances with the highest percentage of memory usage in comparison to their SQL Server allocated memory. This widget displays the following information:

- Instance name
- Allocated (KB)
- Usage (KB)

Clicking a specific server (instance name) in the widget allows you to navigate to the **Resources > Memory** tab of the single instance dashboard for that server.

### Top Servers by Response Time

This widget provides you with information on the top response times in milliseconds for all instances in your environment. Keeping a tab of the top response times helps monitor congestion in a network and the performance of SQL Server. This widget displays the following information:

- Instance name
- Response time in milliseconds

Clicking a specific server (instance name) in this widget allows you to navigate to the overview tab of the single instance dashboard for that server.

### Top Servers by waits

This widget provides you with information on the top waits for all instances in their environment. This widget displays the following information:

- Instance name
- Waits

Clicking a specific server (instance name) in the widget allows you to navigate to the **Resources > Server Waits** tab of the single instance dashboard for that server.

### Top servers by queries

This widget allows you to view the top instances with the highest number of queries in your SQL Server environment. Keeping a tab of this metric in your environment lets you prevent overloading your server.

This widget displays the following information:

- Instance name
- Queries (ms)

Clicking a specific server (instance name) in the widget allows you to navigate to the main dashboard.

### Top servers by I/O
This widget provides you with information on the top 10 instances with the highest I/O load by SQL Physical I/O (reads and writes) in their SQL Server environment. Keeping tab of the I/O subsystem performance prevents increased timeouts and response times.

This widget displays the following information:
- Instance name
- Physical I/O (per sec)

Clicking on a specific server (instance name) in the widget allows you to navigate to the Resources > Disk tab of the single instance dashboard for that server.

**Top servers by sessions**

This widget allows you to view the top values of the total number of sessions by instance in your SQL Server environment. Keeping a tab of this metric in your environment lets you monitor the system load.

This widget displays the following information:
- Instance name
- Sessions

Clicking a specific server (instance name) in the widget allows you to navigate to the Sessions > Summary tab of the single instance dashboard for that server.

**Top servers by blocked sessions**

This widget allows you to view the top blocked sessions by instances in your SQL Server environment. This widget displays the following information:
- Instance name
- Blocked sessions

Clicking a specific server (instance name) in the widget allows you to navigate to the Sessions > Overview tab of the single instance dashboard for that server.

**Top Servers by active connections**

This widget provides you with information on the top number of most user connections by instance in a SQL Server environment. This widget helps you keep on top of the load on the system and monitor bottlenecks.

This widget displays the following information:
- Instance name
- Connections

Clicking a specific server (instance name) in the widget allows you to navigate to the Sessions > Summary tab of the single instance dashboard for that server.

**Top servers by disk space utilization**

This widget allows you to view the top instances with the highest percentage of disk space utilization in your SQL Server environment. Keeping a tab of this metric in your environment, lets you monitor available and used disk space so that there is enough space to be allocated to databases, databases backups, and any other types of files that you may want to store on the server.

This widget displays the following information:
- Instance name
- Utilization (%)

Clicking a specific database name in the widget allows you to navigate to the Resources > Disk tab of the single instance dashboard for that server.

**Top Servers by tempdb utilization**

This widget allows you to view the top values of the instances with the most tempdb space utilization in your SQL Server environment. High tempdb utilization may lead to costly performance issues. This widget displays the following information:
- Instance name
- Space utilization (MB)

Clicking a specific server (instance name) in the widget allows you to navigate to the Databases > Tempdb tab of the single instance dashboard for that server.

**Top sessions by CPU usage**

...
This widget allows you to view the top values of sessions by CPU Usage in your SQL Server environment. This widget helps you monitor the system load on your SQL Server environment. This widget displays the following information:

- Instance name
- Session ID
- Host
- Status
- Database
- Session CPU Usage

Clicking a specific server (instance name) in the widget allows you to navigate to the Sessions > Summary tab of the single instance dashboard for that server.

**Top queries by execution time**

This widget allows you to view the top longest running queries in your SQL Server environment. Long running queries not only consume system resources that make the server and application run slowly, but they may also lead to table locking and data corruption.

This widget displays the following information:

- Instance name
- Database name
- Query name
- CPU time (ms)
- Physical Reads
- Logical Reads
- Logical Writes
- Query execution time (ms)

Clicking on any row in this widget allows you to navigate to the Queries > Summary tab of the single instance dashboard for that server.

You can configure the Top queries by execution time widget by time frame and records to display. Click the gear icon to select a specific number of records, and choose to display the longest running queries for the Last 24 hours or Last 7 days. Note that the maximum amount of records the widget displays is 50.

**Top databases by size**

This widget provides you with information on the top databases by file size across your SQL Server environment. Note that values represent the actual amount of data used inside a database. This widget displays the following information:

- Database name
- Size (MB)

Clicking a specific database (database name) in the widget allows you to navigate to the main dashboard.

**Top databases by growth (Last 7 days)**

This widget allows you to view the top fastest growing databases by instance in your SQL Server environment. Keeping a tab of this metric in your environment, lets you predict how large databases may grow. This widget displays the following information:

- Database name
- Growth (MB)

Clicking a specific database name in the widget allows you to navigate to the main dashboard.

**Top databases by alerts**

This widget allows you to view the top instances with the highest number of database alerts in your SQL Server environment. This widget displays the following information:

- Database name
- Alerts

Clicking a specific database name in the widget allows you to navigate to the Databases > Summary tab of the single instance dashboard for that server's database.

**Top databases by activity**

This widget provides you with information on the top databases by activity size across your SQL Server environment. This widget displays the following information:
Clicking a specific database name in the widget allows you to navigate to the **Databases > Summary** tab of the single instance dashboard for that database.

*SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >*

---

### View all your alerts in the web console

The Alerts tab of the SQLdm web console allows you to view all alerts for all the servers in your environment. The information displayed in this view includes:

- Alert start date & time
- Summary
- Instance
- Database
- Category

You can organize the alerts in the list using either the **Current View** or the **Filter**. Click image to view full size.

Organize your alerts by the **Current View**

You can organize your alerts by selecting one of the following views from the Current View section.

- **Active**
  
  Organizes all your alerts that are currently active.

- **By Severity**

  Organizes all your alerts by their severity.

- **By Instance**

  Organizes all your alerts by monitored SQL Server instance.

- **By Metric**

  Organizes all your alerts by each of the metrics affected.

- **By Category**

  Organizes all your alerts by their category.

- **Custom**

  Organizes all your alerts by selected time range.

Organize your alerts using the **Filter**

You can organize your alerts using the available filter options. Filter options help pinpoint to data you want to view.

You can filter by SQL Server instance, metric, severity, and time range.

- To apply new filters, select the **clear filter** button first.
### Access alert details

The Alerts Detail screen in the SQLdm web console provides you with detailed statistics on your alerts.

On the Alerts Detail screen you can find the following information:

- Alert Name
- Instance Name
- Value (description)
- Warning Threshold
- Critical Threshold
- Current alert time
- Alert Details

**Additional button options**

**Category**

Click the **SHOW CATEGORY DETAILS** button to access first the Overview tab of a single instance and then specific category views such as sessions, queries, query waits, resources, databases, and alerts.

**Instance**

Click the **SHOW INSTANCE DETAILS** button to access the single instance dashboard for a specific instance.

**Close**
Click the **CLOSE** button to exit the Alerts Detail screen.

**Prev**

Click the **PREV** button to view details for a previous alert.

**Next**

Click the **NEXT** button to view details for the next alert.

**Access the Alerts Detail screen**

SQLdm provides several paths to access the Alerts Detail screen. The first access path is by clicking any row in the alerts roll up view of the Alerts tab. The second access path is by clicking the **Show Alert** link of a category icon that contains active alerts in the thumbnail or heatmap sub-views. It is also possible to access the Alerts Detail screen through the main SQLdm web console dashboard and specific widgets.

**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 the configuration properties of all your monitored SQL Server instances**

The Server Configuration Properties view allows you to view the important configuration settings for all of your monitored SQL Server instances. While you manage these settings in different areas of SQLdm, you can see a current list of all of your instances and their properties settings in one view.

**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](#).

**Monitor instances**

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.
- Timeline view where you can see a timeline of events as they occurred on this instance.
Access the Overview tab

SQLdm provides two paths to access the Overview tab. The first access path is simply by clicking the appropriate monitored SQL Server instance. The Overview tab is the default view when you first click an instance. The second access path is used when you are on a different tab within an instance. In this situation, click the Overview tab.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Navigate to your SQL Server instances

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. Select a tag to show a filtered view of the servers within that tag.

View alerts on the Server tree

SQL Server instances in the Server tree also display status icons. When a SQL Server instance is associated with a warning or critical metric icon, use your mouse cursor to hover over the server name to view the top three associated alerts. This feature allows you to prioritize your SQL Server instances and the work that you need to perform.

Refresh your alerts

A full refresh of alerts excluding table fragmentation occurs in the following circumstances even if longer collection intervals are defined:

- when the collection service starts
- when the collection service receives a new workload, such as redirecting the Management Service to a new Repository
- when you add a SQL Server for collection
- when a SQL Server resumes from Maintenance Mode
- when a user selects Refresh Alerts

This refresh does not cause collection of non-alertable data, nor cause collection of anything that is disabled.

Overview tab features

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

The Overview tab contains the following options:

- **Dashboard**
  - The Dashboard view is where you can see a graphical representation of the most important metrics for the selected SQL Server instance.

- **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 instance.

- **Active Alerts**
  - The Active Alerts view allows you to view and control the active alerts as of the most recent data collection.

- **Timeline**
  - The Timeline view allows you to view all alerts and maintenance mode events as well as the first monitored event for the selected SQL Server instance.

Other Servers tree features

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 monitored SQL Server instance.
Refresh Alerts
Gathers the alert status and refreshes the tree view information for the selected monitored SQL Server instance.

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.

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 Baseline
Allows you to configure the performance baseline for the selected SQL Server instance.

Apply Alert Template
Allows you to select an existing alert template to apply to the selected SQL Server instance.

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

Snooze Alerts
Allows you to snooze all the alerts for the selected SQL Server instance instead of selecting the snooze feature for each alert individually.

Resume Alerts
Allows you to resume any alerts set to snooze on the selected SQL Server instance.

Properties
Allows you to view and edit the collection properties of your SQL Server instance.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Get the SQL Server performance overview
The Dashboard view contains a graphical representation of the activity on your SQL Server instance. This view not only displays the status of your SQL Server instance, but it allows you to diagnose where problems originate.

The Dashboard view is made up of panels, which include additional information for specific metrics in chart form. 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.

You can also customize your Dashboard view to show only those panels that are important to your monitored SQL Server instance. SQLdm also allows you to save your customized view on a per-instance basis, or use the same view for all of your instances. For more information about customizing your Dashboard view, see Customize your Dashboard view.

Access the Dashboard view
You can open the Dashboard view of the SQLdm Overview tab by selecting the appropriate SQL Server instance, and then clicking Overview > Dashboard.

Dashboard view
The Dashboard view is organized with panels that display all the key metrics collected by SQLdm by category.

The available data panels for the Dashboard view include:

Cache
The Cache panel 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 panel** 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
- Processor Queue Length
- Call Rates

**Custom Counters**

The **Custom Counters panel** tracks selected custom counters for your monitored SQL Server instance. You must have at least one custom counter to view data in this panel.

**Databases**

The **Databases panel** 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:

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

**Disk**

The **Disk panel** 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

**File Activity**

The **File Activity panel** includes the top five database files with the highest relative activity since the last refresh of your monitored SQL Server instance. You can select different metrics and display results for all files, or a single database, disk, or file. The available metrics include:

- Reads/sec
- Writes/sec
- Transfers/sec

**Lock Waits**

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

**Memory**

The **Memory panel** 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:

- SQL Server Usage
- SQL Server Paging
- VM Memory Usage
- Host Memory Usage

**Network**

The **Network panel** tracks the performance of the network connection being used by your monitored SQL Server, and includes the following charts and gauge:

- SQL Server Throughput
- VM Network Usage Throughput
- Host Network Usage Throughput
- Response Time

**Server Waits**

The **Server Waits panel** tracks the total number and time spent on waits affecting your monitored SQL Server.
Sessions
The Sessions panel includes session-specific data items. The metrics displayed in Sessions include the following chart and gauge:

- Activity & Blocking
- Clients

Tempdb
The Tempdb panel tracks the status of the tempdb database on your monitored SQL Server instance, and includes the following charts and gauge:

- Tempdb Space Used
- Tempdb Contention
- Version Store Cleanup Rate

Virtualization
The Virtualization panel tracks the performance of the host server that manages your virtual machines and the VM on which the monitored SQL Server instance is running, and includes the following charts and gauge:

- Virtual Machine Memory Areas
- VM Host Memory Areas
- Virtual Machine Disk Usage
- VM Host Disk Usage
- CPU Ready

Understand the Dashboard colors
The following colors are associated with a status for each of the provided panels.

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

Blue
Indicates the status of the data is at an informational level based on your settings.

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.

Blank metrics on the Dashboard
If a metric on the Dashboard has no value, or displays as blank, SQLdm is not able to collect the corresponding counter. Causes of this situation include:

- The SQLdm Console has not captured the metrics. It can take up to two refreshes for this capture to occur.
- OS metrics are disabled. SQLdm requires direct WMI or OLE automation to collect data.
- If using OLE automation, those procedures are unavailable. SQLdm is unable to locate required OLE automation stored procedures.
- If connecting to WMI directly, the WMI Service is not running. SQLdm requires that the WMI service is 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.

Refresh your Dashboard data
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.

Understanding your chart data and refresh interval
SQLdm charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQLdm displays your chart data, see Charts.
Customize your Dashboard view

The Dashboard view is customizable per monitored SQL Server instance by selecting the panels that are important to you for that instance. You can customize your Dashboard view for the selected SQL Server instance or as the default for all SQL Server instances added to SQLdm.

Using the Panel Gallery

When you install SQLdm, a default Dashboard view is applied to all your SQL Server instances. The Panel Gallery allows you to alter your view so that you can monitor the metrics that are most important. You can access the Panel Gallery by selecting the appropriate SQL Server instance from the Servers pane, and then clicking Customize. Once in the Dashboard Designer, click a panel in the Panel Gallery for SQLdm to display additional information about that panel.

Designing a Dashboard view

When you design a Dashboard view, it is important that you select the metric panels that give you the best information for your selected SQL Server instance and the current status of that instance. Because SQLdm allows you to create multiple designs, you can create one design with panels specifically for SQL Server 2000 instances and another design for SQL Server 2005 instances. It allows you to create designs that show the most important panels to monitor when certain events trigger alerts.

SQLdm allows you to choose from several different layouts of columns and rows for your dashboard design. You can then drag and drop panels from the Panel Gallery to the actual location in the layout where you want that panel to reside. You can also click, hold, and drag panels to re-arrange their location. When you drag a panel to another position and release the mouse button, the panel you moved switches locations with the existing panel. For example, your design includes the CPU panel in the upper left corner and the Sessions panel in the lower right corner. Click, hold, and drag the Sessions panel to the upper left corner, and then release. Note that the Sessions panel is in the upper left corner and the CPU panel is now in the lower right corner. For more information on available designs, see Design a dashboard view.

SQLdm includes several options for you to use as your Dashboard view. You can also create a design using the available panels, and save the design for other people in your environment to use as their Dashboard view. For more information on available designs, see Select a dashboard design.

Saving a design

Once you are pleased with the selection and positioning of your panels, you can save your design. In the Dashboard Designer, click Save. SQLdm allows you to name this design, which is then displayed in the Dashboard Gallery. For additional information about saving a dashboard design, see Save a dashboard design.

Understanding how you can apply a dashboard design

SQLdm allows you to use a design for a single session of the SQLdm Management Console, apply a design to a specific monitored SQL Server instance, or apply the layout to any other SQL Server instance including those you add later. You can use a custom Dashboard view in the following situations:

One-time use

SQLdm allows you to view a custom layout of your Dashboard view for a single session of the SQLdm Management Console. For example, an event in one of your instances triggered an alert. You want to watch the metric that includes statistics relating to this alert, so you access the Panel Gallery in the Dashboard Designer. Drag the panel displaying the alert-related statistics to the top-left corner of your layout, and then close the Dashboard Designer. SQLdm displays a message that you made changes to the current design. Click No to use this design only until you close this session of the SQLdm Management Console.

SQLdm keeps that panel in the position of your Dashboard view until you close the SQLdm Management Console session. Even after accessing other areas within SQLdm, when you return to your Dashboard view, the same panel appears in the same location.

When you no longer want that panel foremost in your Dashboard view, simply close the SQLdm Management Console knowing that the default Dashboard view for that instance appears the next time you access SQLdm.

Set as the default design for one or more instances

While the default Dashboard view includes panels that are important to your monitored SQL Server instances, the panels may not be in the order in which you want or this design does not include all of the panels you want to view. If there are no existing designs you want to use, access the Panel Gallery to select and move the panels you want to the positions that most suit your needs.

When you are done with your design, click Save to make sure you can use this design again. Type a unique name for this design, verify that this SQL Server instance appears in the Selected Servers area. You can add any other SQL Server instances by selecting the name of that instance, and then clicking Add > to move the instance to the Selected Servers area. Once your list of selected instances is complete, click OK.

Each time you access the Dashboard view for this instance, SQLdm displays the information using this design.

You can also select an existing dashboard view as the default for one or more monitored SQL Server instances. On the Server Overview tab, click Customize, and then in the Dashboard Designer, click Select a Dashboard. Scroll through the available designs until you find the one you want to use for this monitored SQL Server instance. Click Select and then each time you access this Dashboard view for this instance, SQLdm displays the information in this design.

Set as the default design for all new or unassigned instances
SQLdm allows you to select a design to use as the default for all new or unassigned monitored SQL Server instances so that you have a consistent format across all of your monitored SQL Server instances. When saving a dashboard design, check the Use this dashboard as my default for new or unassigned servers check box.

**Available panels**
- Cache panel
- CPU panel
- Custom Counters panel
- Databases panel
- Disk panel
- File Activity panel
- Lock Waits panel
- Memory panel
- Network panel
- Server Waits panel
- Sessions panel
- Tempdb panel
- Virtualization panel

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

**Design a Dashboard view**

Throughout the day, your monitored SQL Server instances can go through a number of situations including triggered alerts requiring immediate attention. Some events result in multiple alerts, each needing you to keep a close eye on as a resolution is put into place. To help you monitor your SQL Server instances, SQLdm provides a default Dashboard view that includes commonly-monitored metrics for the selected SQL Server instance.

You can improve the Dashboard view by customizing your view to include those panels most important to you. A customized Dashboard view can help you keep your monitored SQL Server instances from triggering alerts by providing you real-time coverage of crucial metrics.

You can view your current Dashboard view by making the appropriate changes, and then closing the Panel Gallery pane for a full view of your design. Click Panel Gallery to toggle the display of the Panel Gallery selection pane. You can also close the Panel Gallery by clicking the close icon in the upper right corner of the pane.

**Identify which Dashboard panels you want to include**

In the Panel Gallery, each panel includes a brief description and a link to access additional information about that panel. To view the description, click the panel for which you want to view details. SQLdm also displays a sample graphic showing the kinds of charts and data available in the panel. Click More Info in the description to access the help topic specifically for that panel.

Note that you can use the same panel more than once in your Dashboard design. Because some panels in the Panel Gallery provide drop-down selections for you to select metrics or other filters, you can use the same panel more than once in your Dashboard. This feature is handy when using panels such as the Databases panel because it allows you to have one Databases panel displaying a selected metric and another Databases panel displaying a different metric. You can so the same with the Disk panel to view the same metrics for different disks at the same time.

**Manage your Dashboard view**

To change your current Dashboard view:

1. Select the appropriate SQL Server instance from the Servers pane.
2. Click Customize. Notice that you are now in the Dashboard tab. SQLdm displays this tab when you use the Customize feature.
3. If you want to change the number of panels to display in your view, click Select Layout, and then select the number of columns and rows.
4. Using a drag-and-drop method, move the panels you want to include from the Panel Gallery to the location within the view where you want that panel to reside. You can place the same panel more than once in your view. This feature is important on panels that allow you to select different metrics, such as the Databases panel which offers different measurements per second.
5. Click Close Designer once you select and move panels to where you want them to reside.
6. On the confirmation message, click Yes to use this design again or apply it to additional SQL Server instances, or click No to use this design for only this session of the SQLdm Management Console. For additional information about saving a dashboard design, see Save a dashboard design.

**Select an existing dashboard design**

SQLdm includes a default Dashboard view and multiple designs with installation. If you want to change your view, you can select each panel specifically for a custom view or you can select from existing layouts that you or another user in your environment created. This feature allows you
to share designs as templates that each user can use to save, and then apply to their monitored SQL Server instances.

Select an existing design for your Dashboard view

To use an existing design for your Dashboard view:

1. Select the appropriate SQL Server instance from the Servers pane.
2. Click Customize. Notice that you are now in the Dashboard tab. SQLdm displays this tab when you use the Customize feature.
3. Click Select a Dashboard.
4. Scroll through the displayed designs until you find the one you want to use. Click the arrows to scroll through the available designs. Click All Dashboards to view designs other users in your environment created and made available for use. This feature allows you to "share" a design that you find effective.
5. Click Select when you find the design you want to use. SQLdm returns you to the Dashboard Designer using your selected design. You can then save this design to use for this SQL Server instance or click Close Designer to use this design only during this session of the SQLdm Management Console. For additional information about saving a dashboard design, see Save a dashboard design.

Delete a Dashboard layout

If you created a layout that you no longer want anyone to use, you can delete that layout from your gallery. Simply click Select a Dashboard, scroll to the layout you want to delete, and then click Delete. You cannot delete any of the layouts supplied with your SQLdm installation.

Save a dashboard design

There are two options when saving a design for your Dashboard view. Check Use this dashboard as my default for new or unassigned servers if you want SQLdm to use this design for all new SQL Server instances and for existing instances that do not have a designated default Dashboard view.

To save a dashboard design:

1. In the Dashboard Designer, click Save.
2. Type the name you want for your new design. If you use the name of the existing design, SQLdm displays a message alerting you to the existing design. You can then type a different name or use the same name, replacing the existing design with this new format.
3. Verify that the correct monitored SQL Server instance appears in the Selected Servers area. You can apply this design to additional instances by selecting the instance, and then clicking Add > to move the instance from Available Servers to Selected Servers.
4. Check Use this dashboard as my default for new or unassigned servers if you want SQLdm to use this design as the default Dashboard view for all of your new monitored SQL Server instances and any instance that does not currently have a designated default design.
5. Click OK.

Cache panel

The Cache panel 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. For additional information, see how to manage your Procedure Cache.

If the database cache size seems to be too small for optimal performance and there is very little available memory on the system, adding more memory to the system may increase performance. If there is a lot of available memory on the system and the database cache size is not growing beyond a certain point, the database cache size may be capped at an artificially low limit. Increasing this limit may increase performance.

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 finds 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 displays the usage in megabytes of each type of cache in use.
Database
Indicates the amount of system memory (in MB) used by the database cache manager to hold commonly-used information from the database file(s) to prevent file operations.

Free
A low amount of free cache indicates a higher risk of performance issues.

Procedure Cache
Indicates the amount of space used to store query plans.

Other
Indicates the amount of space used by other processes.

**Cache 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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**CPU panel**

The CPU panel tracks the performance of the processor 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. For additional information, see how to get your CPU performance details.

**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. For additional information about how SQLdm works with virtual machines and collects metrics, see How SQLdm works with a virtual environment.

<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 can indicate that SQL Server is spending too much time processing queries.</td>
</tr>
<tr>
<td>Total usage</td>
<td>A high total CPU usage can indicate that this server does not have enough resources to adequately process its current workload.</td>
</tr>
<tr>
<td>VM usage</td>
<td>A high virtual machine usage can indicate that the processor power allocated to this VM is insufficient for its current workload.</td>
</tr>
<tr>
<td>Host usage</td>
<td>A high host server usage indicates the processing power on this host is insufficient to handle the workload of the currently active VMs.</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 on 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 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 can indicate 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>
</tbody>
</table>
Differences in statistics

You may notice some difference in your statistics, such as your OS CPU usage being higher than your VM CPU usage. While the hypervisor manages resources efficiently, the demand for physical resources may be greater than what it has to provide. For example, a guest OS on a virtual machine submits a batch of work to a CPU for execution. However, all of the physical CPUs are committed to other work, so there is no physical CPU available on the host to process the work. In this situation, the VM has to wait for a CPU to become available to process its work. While it is waiting, the guest OS is unaware of the wait and assumes that it is simply taking longer for the CPU to process this batch of work, and the OS thinks it is using more CPU power than it is actually using.

Remember that SQLdm uses the additional overhead of the hypervisor to calculate the VM metrics. In this situation, the hypervisor knows the VM is just waiting for a CPU, so it does not charge the VM for CPU processing power while it is waiting. The result is that the guest OS reports that it is using the additional processing power.

Available alerts

- Host CPU Usage (Percent) Alert
- OS Processor Time (Percent) Alert
- OS Processor Queue Length (Count) Alert
- SQL Server CPU Usage (Percent) Alert
- VM CPU Usage (Percent) Alert

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more »

Custom Counters panel

The Custom Counters panel displays the statistics for a selected custom counter on a monitored SQL Server instance. At least one custom counter must exist on the selected instance for this panel to display any data. Custom counters allow you to include more performance metrics than are provided by SQLdm. For additional information about custom counters, see how to use custom counters to track metrics.

Any alerts affecting the Custom Counters panel depend on the selected custom counter and alert threshold settings.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more »

Databases panel

The Databases panel 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. For additional information, see how to get the database status summary.

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

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.

Writes chart (per second)
The Writes/sec chart displays the number of writes per second performed by the databases on your SQL Server instance.

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.
Disk panel

The Disk panel tracks the performance of the disk used by your monitored SQL Server. Unexpected spikes in latency, throughput, or SQL Server IO may warn you about the beginning of a serious performance issue. Chronically high disk metrics can signify an insufficient IO subsystem, or excessive or inefficient IO activity on the server. Use the Disk view on the Resources tab to track several key disk statistics over a period of time. For additional information, see how to get the disk performance details.

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 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 you should optimize the database indexes.</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 you should optimize the database indexes.</td>
</tr>
<tr>
<td>Read ahead reads</td>
<td>A high number of read ahead reads may indicate that you should tune your query execution plan or that you should defragment the disk.</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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
The File Activity panel displays the top five database files with the highest relative activity since the last refresh. Available metrics allow you to view the top Reads/sec, Writes/sec, and Transfers/sec activity for all files or a selected database, disk, or file. Unexpected spikes in the amount of file activity may warn you of ineffective or excessive indexing or out-of-date statistics. For additional information, see how to view file activity.

**I/O stall chart (ms/sec)**

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

**Reads chart (per second)**

The Reads/sec chart displays the number of physical reads per second the disk sub-system is performing to bring SQL Server database pages into the data (buffer) cache. If the reads rate is excessive for a long period of time, consider taking action to avoid performance issues.

**Reads KB chart (per second)**

The Reads KB/sec chart displays the size of the physical reads per second the disk sub-system is performing to bring SQL Server database pages into the data (buffer) cache.

**Total KB chart (per second)**

The Total KB/sec chart displays the total size of activity per second across all databases on the monitored instance.

**Writes chart (per second)**

The Writes/sec chart displays the number of physical database writes per second the disk sub-system is performing. If the writes rate is excessive for a long period of time, consider taking action to avoid performance issues.

**Writes KB chart (per second)**

The Writes KB/sec chart displays the size of the physical database writes per second the disk sub-system is performing.

**Transfers chart (per second)**

The Transfers/sec chart displays the total number of transfers per second across all databases on the monitored instance. Transfers are the sum of reads and writes. A higher transfer rate indicates a higher risk of resource issues.

**When the File Activity panel displays no data**

There are circumstances under which the File Activity panel does not display any data. For example, when you lose connection to the Collection Service or SQLdm Repository. Another reason is the lack of collection data points. On new instances, file activity collection occurs once per hour by default. By increasing the length of time shown on the desktop client views, you may be able to see more data points. You can modify the timeframe of data displayed in SQLdm by clicking Tools > Console Options while in the File Activity view and changing the Keep data for the last and Show data for the last values.

You also can change the database collection interval and grooming options to help you see more data points. Use the Collect and alert on database metrics field in the General tab of the Monitored SQL Server Properties window to manage your database collection. For grooming options, use the Groom Sessions, Queries, Deadlocks, Waits, and History Browser data older than field in the Grooming Options window.

---

Decreasing the collection interval and grooming your data less often can result in performance issues and require additional storage space in your Repository.

---

**Activity for a new monitored SQL Server instance**

The File Activity panel displays results based on activity since the last refresh. You must have two collections to create a point on the graph, three collections to create a line. By default, the database collection interval is set at one hour, which means that the File Activity panel does not show a point until an hour after adding the instance unless you perform a manual refresh. You can refresh and collect data by clicking the Refresh button in the SQLdm Management Console menu bar.

**Activity when using the History Browser**

The History Browser allows you to view the state of your SQL Server instance at the time a snapshot occurs, which is every six minutes by default. When using the History Browser with the File Activity panel, note that data may appear in one snapshot but not in others because no file activity occurred during that interval. Remember that file activity data does not appear for the first two hours of monitoring a SQL Server instance because the metrics are based on the activity since the last database refresh.

---

**Lock Waits panel**

The Lock Waits panel 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. For additional information, see how to analyze locked sessions.
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>File SQL Server 2005 and above only</td>
<td>Indicates the time waiting for a file lock.</td>
</tr>
<tr>
<td>HoBT SQL Server 2005 and above only</td>
<td>Indicates the time waiting for a Heap or BTree lock.</td>
</tr>
<tr>
<td>Key SQL Server 2000 only</td>
<td>Indicates the time waiting for a key lock.</td>
</tr>
<tr>
<td>Key/RID SQL Server 2005 and above only</td>
<td>Indicates the time waiting for a key or RID lock.</td>
</tr>
<tr>
<td>Metadata SQL Server 2005 and above only</td>
<td>Indicates the time waiting for a metadata lock.</td>
</tr>
<tr>
<td>Object SQL Server 2005 and above only</td>
<td>Indicates the time waiting for an object lock.</td>
</tr>
<tr>
<td>Page</td>
<td>Indicates the time waiting for a page lock.</td>
</tr>
<tr>
<td>RID SQL Server 2000 only</td>
<td>Indicates the time waiting for a relative identifier (RID) lock.</td>
</tr>
<tr>
<td>Table SQL Server 2000 only</td>
<td>Indicates the time waiting for a table lock.</td>
</tr>
</tbody>
</table>

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >

Memory panel
The Memory panel tracks the usage and availability of memory on your monitored SQL Server computer. Unexpected spikes in memory usage can 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. For additional information, see how to get the memory performance details.

SQL Server Usage chart
The SQL Server Usage chart compares the amount of memory allocated and used by SQL Server to the total memory consumed on a 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 is allocated to SQL Server from the operating system (SQLOS). Compare this metric to how much memory is actually being used by SQL Server.</td>
</tr>
<tr>
<td>SQL used</td>
<td>Indicates how much memory is currently used by SQL Server from the total allocated (SQL allocated) to this application. 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>Indicates how much physical memory is used by the operating system (SQL Server and other applications). 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>

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

VM Memory Usage chart
The VM Memory Usage chart compares the amount of memory allocated and used by SQL Server to the total memory consumed on this virtual machine. For additional information about how SQLdm works with virtual machines and collects metrics, see How SQLdm works with a virtual environment.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Indicates the amount of memory actively used by this virtual machine.</td>
</tr>
<tr>
<td>Metric</td>
<td>Why it is important</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Active</td>
<td>Indicates the amount of memory actively used by all of your VMs on this host server.</td>
</tr>
<tr>
<td>Ballooned</td>
<td>Indicates the amount of memory temporarily reclaimed by all of your VMs on the host system for use in other areas. A higher ballooned value indicates that the host server is under memory pressure. Review your load balancing for the host server.</td>
</tr>
<tr>
<td>Consumed</td>
<td>Indicates the amount of memory consumed by all of your VMs on this host server. This value does not include shared memory or overhead memory for these VMs.</td>
</tr>
<tr>
<td>Granted</td>
<td>Indicates the amount of physical memory granted by the host to all of your VMs.</td>
</tr>
<tr>
<td>Swapped</td>
<td>Indicates the amount of memory swapped to disk by the hypervisor.</td>
</tr>
</tbody>
</table>

**Host Memory Usage chart**

The Host Memory Usage chart compares the amount of memory allocated and used by SQL Server to the total memory consumed on this host server. For additional information about how SQLdm works with virtual machines and collects metrics, see [How SQLdm works with a virtual environment](#).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>Why it is important</td>
</tr>
<tr>
<td>Active</td>
<td>Indicates the amount of memory actively used by all of your VMs on this host server.</td>
</tr>
<tr>
<td>Ballooned</td>
<td>Indicates the amount of memory temporarily reclaimed by all of your VMs on the host system for use in other areas. A higher ballooned value indicates that the host server is under memory pressure. Review your load balancing for the host server.</td>
</tr>
<tr>
<td>Consumed</td>
<td>Indicates the amount of memory consumed by all of your VMs on this host server. This value does not include shared memory or overhead memory for these VMs.</td>
</tr>
<tr>
<td>Granted</td>
<td>Indicates the amount of physical memory granted by the host to all of your VMs.</td>
</tr>
<tr>
<td>Swapped</td>
<td>Indicates the amount of memory swapped to disk by the hypervisor.</td>
</tr>
</tbody>
</table>

**Available alerts**

- Host Memory Usage (Percent) Alert
- O/S Memory Usage (Percent) Alert
- O/S Paging Alert
- SQL Memory Usage (Percent) Alert
- VM Memory Usage (Percent)

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. [Learn more](#).

**Network panel**

The Network panel 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.

**SQL Server Throughput chart**

The SQL Server Throughput chart plots how many packets this SQL Server sends and receives 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>

**VM Network Usage Throughput chart**

The VM Network Usage Throughput chart plots how much data this virtual machine transmits and receives in KB/second over time. For additional information about how SQLdm works with virtual machines, see [How SQLdm works with a virtual environment](#).
<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitted KB/sec</td>
<td>A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.</td>
</tr>
<tr>
<td>Received KB/sec</td>
<td>A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.</td>
</tr>
</tbody>
</table>

Host Network Usage Throughput chart

The Host Network Usage Throughput chart plots how much data this host server transmits and receives in KB/second over time. For additional information about how SQLdm works with virtual machines, see How SQLdm works with a virtual environment.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why It Is Important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitted KB/sec</td>
<td>A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.</td>
</tr>
<tr>
<td>Received KB/sec</td>
<td>A consistently high rate may indicate that data throughput is saturated. Investigate load balancing your network traffic.</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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Server Waits panel

The Server Waits panel 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. For additional information, see how to view server waits.

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>
Sessions panel

The Sessions panel 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. For additional information, see how to get the sessions performance summary.

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>

Client Computers 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
- Total Deadlocks Alert

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Tempdb panel

The Tempdb panel tracks the status of the tempdb database on your monitored SQL Server instance. Unexpected spikes in the use of tempdb usage and space may warn you to allocate additional space to prevent tempdb issues from causing a performance issue on your server. For additional information, see how to get the tempdb status summary.

Tempdb Space Used chart

The Tempdb Space Used chart provides different views of how your database is used over time based on the object type. Tempdb can fill up quickly when you are low on disk space and or have a low maximum size for database growth. In environments where read committed snapshot isolation is in use, long running transactions may also cause rapid growth in tempdb.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>Indicates the amount of tempdb space used by internal objects.</td>
</tr>
<tr>
<td>Mixed</td>
<td>Indicates the amount of tempdb space used by mixed extents.</td>
</tr>
<tr>
<td>User</td>
<td>Indicates the amount of tempdb space used by user objects.</td>
</tr>
<tr>
<td>Version Store</td>
<td>Indicates the amount of tempdb space used by the version store.</td>
</tr>
</tbody>
</table>

Tempdb Contention chart

The Tempdb Contention chart provides the latch wait time for the allocation pages of your tempdb. These latch waits are associated with performance degradation for the related queries. Latch contention is usually an indication that you should create additional tempdb data files or if you already have multiple files, make adjustments so they are equally sized. For more information about tempdb contention, see Monitor your tempdb database.
GAM Page | Indicates the latch wait time for the global allocation map pages of tempdb.
---|---
PFS Page | Indicates the latch wait time for the page free space pages of tempdb.
SGAM Page | Indicates the latch wait time for the shared global allocation map pages of tempdb.

**Version Store Cleanup Rate**

The Version Store Cleanup Rate gauge displays the current version store cleanup rate and the version generation rate of the data rows necessary to support snapshot isolation. Maintaining a cleanup rate above or equal to the generation rate helps you avoid filling up tempdb. Note that long-running snapshot isolation transactions prevent cleanup of older entries in the tempdb version store, causing growth which can affect performance and cause tempdb to run out of space. When interpreting this gauge, note that version store cleanup occurs only once per minute while generation may be ongoing. For more information about tempdb version store, see Monitor your tempdb database.

**Available alerts**
- Tempdb Contention Alert
- Version Store Generation Ratio Alert
- Version Store Size Alert

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >

---

### Virtualization panel

The Virtualization panel tracks the performance of your virtual machine or host, depending on the selection in the panel. Unexpected spikes in virtual memory or disk usage may warn you about the beginning of a serious performance issue indicating the need for server maintenance to better handle the ongoing workload. For additional information about how SQLdm works with virtual machines and collects metrics, see How SQLdm works with a virtual environment.

Note that you can customize your Dashboard view to include two Virtualization panels and add one for your virtual machine and the other for your host.

---

**Virtual Memory Usage chart**

The Virtual Memory Usage chart displays the amount of memory allocated and used by SQL Server to the total memory consumed on this VM or host server. The Resources > Memory view allows you to track your usage over a period of time, along with other key memory statistics. For additional information about how SQLdm works with virtual machines and collects metrics, see How SQLdm works with a virtual environment.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>Indicates the amount of active memory used by this virtual SQL Server instance.</td>
</tr>
<tr>
<td>Ballooned</td>
<td>Indicates the amount of ballooned memory on this virtual SQL Server instance. A spike in this value alerts you of the low memory status of the host.</td>
</tr>
<tr>
<td>Consumed</td>
<td>Indicates the amount of memory consumed by this virtual SQL Server instance.</td>
</tr>
<tr>
<td>Granted</td>
<td>Indicates the amount of memory granted to this virtual SQL Server instance.</td>
</tr>
<tr>
<td>Swapped</td>
<td>Indicates the amount of swapped memory on this virtual SQL Server instance. A spike in this value alerts you of the low memory status of the host.</td>
</tr>
</tbody>
</table>

**CPU Ready Wait Time gauge**

The CPU Ready Wait Time gauge displays the current gauge value of the vCPU Ready Time metric available on the Server Details view. Use this metric to determine the length of time that virtual CPU in your virtual machine is waiting on a physical CPU on your host to become available. A spike in this value may indicate over committed physical CPUs on the host server.

**Virtual Disk Usage chart**

The Virtual Disk Usage chart displays the amount of data read from and written to disk for the VM or host server during the sample interval. The Resources > Disk view allows you to track your usage over a period of time, along with other key disk statistics.
<table>
<thead>
<tr>
<th>Metric</th>
<th>Why it is important</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Read</td>
<td>A high number of disk reads may indicate that the VM or host server is executing poorly-performing queries.</td>
</tr>
<tr>
<td>Disk Write</td>
<td>A high number of disk writes may indicate that the VM or host server is running low on allocated memory.</td>
</tr>
</tbody>
</table>

**Available alerts**

- Host Memory Swap Detected Alert
- VM CPU Ready Wait Time (ms) Alert
- VM Memory Swap Detected Alert
- VM Reclaimed/Ballooned Memory (KB) Alert

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Get the SQL Server performance details**

The Details view displays a list of properties for the selected SQL Server instance, a list of the collected metrics and their characteristics, and provides a chart for you to graphically display metrics selected from the displayed list. For each metric, SQLdm displays the current value, value range for reference, alert status, metric category, thresholds for each alert, and whether the metric is a custom counter. To add a metric to the Details chart, check the check box in the chart column for the appropriate metric.

All fields highlighted in the State column are associated with alerts. Right-click any of these metrics and select **Configure Alerts** to edit the associated thresholds.

**Access the Details view**

You can open the Details view of the SQLdm Overview tab by selecting the appropriate SQL Server instance, and then clicking Overview > Details.

**SQL Server instance properties area**

The following properties are displayed for the selected monitored SQL Server instance.

- **Version**
  Indicates the current SQL Server version including service pack and build information on this SQL Server instance.

- **Edition**
  Indicates the current SQL Server edition, including Enterprise, Business Intelligence, or Standard, installed on this SQL Server instance.

- **Running**
  Indicates the timestamp associated with the most recent restart of this SQL Server instance.

- **Clustered**
  Indicates whether this SQL Server instance is part of a cluster. If it is part of a cluster, SQLdm displays the name of the active node.

- **Processors**
  Indicates the total number of processors associated with this SQL Server instance and how many of these processors are currently in use.

- **Host**
  Indicates the name of the server hosting this SQL Server instance.

- **Host OS**
  Indicates the current operating system on the server hosting this SQL Server instance.

- **Host Memory**
  Indicates the total amount of memory on the server hosting this SQL Server instance.
Indicates the current number of databases on this SQL Server instance.

Data Size
Indicates the current size of the data on this SQL Server instance.

Log Size
Indicates the current size of the log files on this SQL Server instance.

VM Name
Indicates the name of the virtual machine on which this monitored SQL Server instance is running.

Virtualization Host Server
Indicates the name or IP address of the virtualization host server on which this virtual machine is running.

Host Memory
Indicates the total amount of memory on the host server on which this virtual machine is running.

Host Physical Processors
Indicates the total number of physical processors associated with the host server on which this virtual machine is running.

Host Logical Processors
Indicates the total number of logical processors associated with the host server on which this virtual machine is running. The number of logical processors is different from the number of physical processors only if the CPU on the host server supports Hyper-Threading.

Understanding your chart data and refresh interval
SQLdm charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQLdm displays your chart data, see Charts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Monitor SQL Server configuration settings
The Configuration view allows you to view the SQL Server configuration settings for the selected SQL Server instance. When diagnosing problems with SQL Server performance it is helpful to see how it is configured. In addition, SQLdm offers the ability to edit some configuration settings, depending on the option selected.

Access the Configuration view
You can open the Configuration view of the SQLdm Overview tab by selecting the appropriate SQL Server instance, and then clicking Overview > Configuration.

To edit a configuration setting:
1. Select an option from the list.
2. Click the Edit Value hyperlink in the Details pane.
3. Enter the new value.
4. Click OK.

Edit your configuration value
The Edit Configuration Value window allows you to enter a new value for the selected SQL Server instance configuration option. Any change to the selected value does not take effect until you restart the selected SQL Server instance.

Access the Edit Configuration Value window
You can access the Edit Configuration Value window by selecting an option, and then clicking Edit Value in the Details pane.
View active alerts for this instance

The Active Alerts view allows you to view a list of all the active alerts as of the last data collection. Click any alert in the list to view details at the bottom of the window.

Access the Active Alerts view

You can open the Active Alerts view of the SQLdm Overview tab by selecting the appropriate SQL Server instance, and then clicking Overview > Active Alerts.

Available actions

The Active Alerts view 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 view allows you to perform the following functions via the right-click context menu for each alert:

- Open the Real Time view for more information on the metric that caused the alert.
- Open the Historical view to review the history of the metric that caused the alert. The Show Historical View option is not available for all alerts.
- Open the Show Details view for more information on the metric that caused the alert. The Show Details View option is not available for all alerts.
- Open the SQLdm online help to the topic for the selected alert.
- Clear the selected alert or clear all alerts of the selected type for this SQL Server instance. The Clear Alert and Clear All Alerts of this Type for this Instance options are not available for all alerts.
- Open the alert configuration for the metric.
- Snooze the alert for a set period of time.
- Collapse or expand all of the displayed alert groups. This option is available only when you use the Group By Box functionality.
- You can also print, export results to Microsoft Excel, and copy the alert information to your Clipboard for additional use.

Get a 12-hour forecast of alert activity

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

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

The 12 Hour Forecast is separated into two-hour segments. Each two-hour segment displays the chance of an alert occurrence. 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.

**12 Hour Forecast icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="OK" /></td>
<td>OK - There is a low chance (less than 20%) that a warning or critical alert occurs in the two-hour time segment leading to the forecast time.</td>
</tr>
<tr>
<td><img src="image" alt="Warning" /></td>
<td>Warning - There is a moderate likelihood (less than 50%) that a critical alert occurs, or a high likelihood (50% or higher) that a warning alert occurs, in the two-hour time segment leading to the forecast time.</td>
</tr>
<tr>
<td><img src="image" alt="Critical" /></td>
<td>Critical - There is a very high likelihood (50% or higher) that a critical alert occurs in the two-hour time segment leading to the forecast time.</td>
</tr>
</tbody>
</table>
Alert forecast details

The Alert Forecast Details window displays the Server, Alert Metric, the likelihood the alert occurs, and the type of alert expected in the two-hour window leading up to this forecast time. To sort the list, click any column heading.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Snooze alerts

You have the ability in SQLdm to snooze or pause all alert generation for a specified period of time. For example, you can snooze a specific alert through the Active Alerts tab. Right-click the alert that you no longer want to view, and then select Snooze Alert.

If you want to pause multiple alerts at the same time, you can mass snooze alert generation. To mass snooze, right-click a specific instance, and then select Snooze Alerts. This option is also accessible when right-clicking a specific tag, using certain server views, and when using the Tools menu.

This feature is important if you are working on known issues and you want to avoid unnecessary alert traffic.

View past performance

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. You can use this information 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.

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.

Access the History Browser

To access the History Browser, click the History Browser button on a supported view, such as Overview > Dashboard.

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.

The icons next to each of the snapshots represent the highest level of alert thresholds violated. These icons 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.

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.

Alerts with associated views that support historical snapshots

On the Alerts view, some of the alerts allows you to retrieve the associated view at the time the alert was activated. To access this view, right-click the alerts, and the click Show Historical View in the Details pane.

The following table displays a list of alerts that provide an associated view in SQLdm.

<table>
<thead>
<tr>
<th>Alert</th>
<th>Associated View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Section</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Average Disk Milliseconds Per Read</td>
<td>Resources</td>
</tr>
<tr>
<td>Average Disk Milliseconds Per Transfer</td>
<td>Resources</td>
</tr>
<tr>
<td>Average Disk Milliseconds Per Write</td>
<td>Resources</td>
</tr>
<tr>
<td>Blocked Sessions (Count)</td>
<td>Sessions</td>
</tr>
<tr>
<td>Blocking Session Wait Time (Seconds)</td>
<td>Sessions</td>
</tr>
<tr>
<td>Client Computers (Count)</td>
<td>Sessions</td>
</tr>
<tr>
<td>Databases Read/Write Errors Occurred</td>
<td>Databases</td>
</tr>
<tr>
<td>Disk Reads Per Second</td>
<td>Resources</td>
</tr>
<tr>
<td>Disk Transfers Per Second</td>
<td>Resources</td>
</tr>
<tr>
<td>Disk Writes Per Second</td>
<td>Resources</td>
</tr>
<tr>
<td>Host CPU Usage (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>Host Memory Usage (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>Oldest Open Transactions (Minutes)</td>
<td>Sessions</td>
</tr>
<tr>
<td>OS Average Disk Queue Length (Count)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Average Disk Queue Length Per Disk (Count)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Disk Time (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>OS Disk Time Per Disk (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>
<tr>
<td>Page Life Expectancy</td>
<td>Server</td>
</tr>
<tr>
<td>Session CPU Time (Seconds)</td>
<td>Sessions</td>
</tr>
<tr>
<td>SQL Server CPU Usage (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>SQL Server Memory Usage (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>SQL Server Response Time (Milliseconds)</td>
<td>Sessions</td>
</tr>
<tr>
<td>Tempdb Contention (ms)</td>
<td>Databases</td>
</tr>
<tr>
<td>User Connections (Percent)</td>
<td>Sessions</td>
</tr>
<tr>
<td>VM CPU Usage (Percent)</td>
<td>Resources</td>
</tr>
<tr>
<td>VM Memory Usage (Percent)</td>
<td>Resources</td>
</tr>
</tbody>
</table>

**Views that 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 > Dashboard
- Overview > Details
- Overview > Active Alerts
- Sessions > Summary
- Sessions > Details
- Sessions > Locks
- Sessions > Blocking
Pause alerts and decrease collection

Maintenance Mode allows you to temporarily stop alert generation and the collection of some performance metrics for the time period that your SQL Server instance is offline. Using Maintenance Mode allows you to decrease alert notification spam and the collection of unnecessary data.

It is possible to enable or disable maintenance mode on several SQL Server instances, see About Mass Maintenance Mode.

Uses for putting a SQL Server instance into Maintenance Mode

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 offline 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.

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, make sure you complete all maintenance and testing 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 Maintenance Mode 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 Server 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, and encryption options.

- **Baseline Configuration**
  - Allows you to set the date range and time used to calculate a performance baseline for the monitored SQL Server instance.

- **Query Monitor**
  - Allows you to enable or disable query monitoring, and set the options for the Query Monitor.

- **Activity Monitor**
  - Allows you to enable or disable monitoring of non-query activities, and set the options for the Activity Monitor.

- **Replication**
  - Lets you disable replication statistics collection to conserve system resources.
Table Statistics
Allows you to select the time and days SQLdm collects table statistics on your SQL Server instance. In addition, you can select the minimum size table needed for SQLdm to rebuild the index.

Custom Counters
Allows you to link custom counters to the monitored SQL Server instance.

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 is unavailable.

OS Metrics
Allows you to select how you want your operating system data collected, whether through OLE automation or direct WMI. You can also choose to not collect OS data.

Disk Drives
Allows SQLdm to alert on disk drives and database space calculations for the monitored SQL Server instance.

Cluster Settings
Lets SQLdm trigger an alert whenever a cluster is running on a non-preferred node within a clustered environment.

Wait Monitoring
Allows you to collect and view wait statistics in the hopes of diagnosing any bottleneck on your system.

Virtualization
Displays the name of the host server on which the selected monitored SQL Server instance is running.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

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.

Access the General tab
You can access the General tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting **Properties**.

Associate tags with an instance
The tag field allows you to select a tag to add your server to or add a new tag. Tags help you organize server instances into meaningful groups.

Select a diagnostic data collection interval
The data collection interval is the amount of time SQLdm waits between each collection of data on the selected SQL Server instance. You can set a different interval for each of the following functions:

- **Collect diagnostic data and raise alerts (max 30 minutes)**
  The interval between times that the SQLdm Collection Service collects diagnostic data and raises the associated alerts. Lower values cause SQLdm to raise alerts more quickly, but also cause more frequent refreshes, which increases your monitoring overhead.

- **Collect and alert on database metrics (max 24 hours)**
  The interval between times that the SQLdm Collection Service collects database space-related data and raises the associated alerts. Lower values cause SQLdm to raise alerts more quickly, but also cause more frequent refreshes, which increases your monitoring overhead. In environments with a large number of databases whose sizes do not change rapidly, setting the database data collection to a long interval can greatly reduce the monitoring footprint.

By default, SQLdm collects diagnostic data every six minutes and database data every 60 minutes. Consider the following factors before selecting your data collection interval:

Data collection purpose
If your goal with this data collection is to identify broad trends and alert on critical downtime, then a longer refresh interval is appropriate. If you are closely monitoring your SQL Server instance for minute-by-minute changes, use a shorter duration.

**Data collection frequency**

When diagnosing specific problems or when working with a problematic SQL Server instance, a short interval lets you capture enough data to diagnose the issue. In most situations, the default interval provides sufficient data for your diagnostic needs. Note that SQL Server instances used only occasionally may require less monitoring attention.

**Notification of existing or potential problems**

Remember that a lower collection interval results in SQLdm raising alerts more quickly, but also increases your monitoring overhead. This is true when any setting causes more frequent refreshes.

**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.

**Space needed for the SQLdm Repository**

A lower collection interval results in more frequent refreshes which are all stored in the SQLdm Repository. Make sure you have enough available space to accommodate these lower settings.

**Data spikes**

SQLdm averages most metrics over the timeframe between collections. More-frequent data collection causes increased movement in averages because of short-duration events. For example, a 15-second CPU spike has a greater effect on a one-minute refresh than on a six-minute refresh. You can use alert smoothing to reduce the impact of data spikes on a per-alert basis.

**Select a server availability verification interval**

The server availability verification interval is the amount of time SQLdm waits between verifying availability on the selected SQL Server instance. If the connection test collector does not complete within the time specified in the **Alert if the server is inaccessible (max 10 minutes)** field, the SQL Server instance is considered unresponsive. Setting this field to a very low value can result in false positive alerts.

**Collect extended session data**

The extended session collection data includes important session information, such as details, locks and blocks. If this information is important to you, make sure the **Collect extended session data, including session details, locks, and blocks** check box is selected.

**Limit the number of DBCC Inputbuffer executions**

You can limit the number of executions performed by the DBCC Inputbuffer, which retrieves the actual input command for the Session Details view, among others. Note that on busy servers, decreasing the **Limit executions of DBCC Inputbuffer to** value can reduce monitoring impact.

**Data collection credentials**

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.

**Select encryption options**

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 used to encrypt the data between the collection service and the monitored SQL Server instance. For this option to work correctly, configure the monitored SQL Server instance to support encryption.

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

This option is available only when the Encrypt Connection (SSL) is selected. This option allows you to skip the certificate validation 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 is rejected unless the Trust Server Certificate option is selected.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Configure server baseline options

The Baseline Configuration tab of the Monitored SQL Server Properties window allows you to configure your server baseline options. SQLdm calculates a baseline out of a pool of collected data based on the selected period and collection interval, and is represented by mean, standard deviation, minimum, maximum, and count statistics. One benefit of using this feature is that SQLdm uses the baseline to provide alert recommendations which you can use to set effective alert thresholds.

SQLdm allows you to select the date range used to calculate the performance baseline for one or more selected SQL Server instances. The day and time period fields allow you to select the days and times on which to base your alert baseline calculations. When your SQL Server instances are not busy, clear the days and times to remove them from the calculation and provide a more realistic and accurate baseline. The days and hours selected apply to both the Automatic and Custom baseline selections.

Access the Baseline Configuration tab

You can access the Baseline Configuration tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting Configure Baseline.

Select a baseline period

The baseline period you specify should represent a typical operation of the monitored SQL Server instances to which this baseline applies.

- **Automatic**
  Automatic calculates a dynamic baseline based on a moving window of the past seven days of collected data.

- **Custom**
  Custom allows you to set a specific time range that does not change.

Apply this baseline to more than one instance

You can apply a specific baseline to additional instances not included when you originally created the baseline configuration.

To add a baseline configuration to additional instances:

1. In the Servers pane, right-click the name of any instance that you want to have the new baseline configuration, and then select Configure Baseline.
2. Make any necessary changes to the baseline configuration, and then click Apply Baseline Configuration.
3. Select the appropriate instances from the Available Servers list, and then click Add > to move them to the Added Servers list.
4. Click OK. SQLdm applies this configuration to the other instances.

When verifying your baseline configuration changes for more than one instance in the Change Log view, note that instance property changes are logged only once. In comparison, baseline configuration changes are logged as many times as the number of instances affected by the change.

Metrics used in baseline calculations

SQLdm uses the following metrics when calculating a baseline for a monitored SQL Server instance:

- Available Memory
- Batches Per Second
- Blocked Sessions
- Buffer Cache Hit Ratio
- Buffer Cache Size
- Checkpoint Writes Per Second
- Client Computers
- Data File Count
- Data File Space Allocated
- Data File Space Used
- Data File Space Used (Percent)
- Database Count
- Disk Queue Length
- Disk Time (Percent)
- Distribution Latency
- Host CPU Usage
- Host CPU Usage MHz
- Host Disk Device Latency
- Host Disk Kernel Latency
- Host Disk Queue Latency
- Host Disk Read
- Host Disk Total Latency
- Host Disk Usage
- Host Disk Write
- Host Memory Swap In Rate
- Host Memory Swap Out Rate
- Host Memory Active
- Host Memory Consumed
- Host Memory Granted
- Host Memory Usage
- Host Net Received
- Host Net Transmitted
- Host Net Usage
- Full Scans Per Second
- Lazy Writer Writes Per Second
- Lock Waits Per Second
- Log File Count
- Log File Space Allocated
- Log File Space Used
- Log File Space Used (Percent)
- Log Flashes Per Second
- Logins Per Second
- Oldest Open Transaction (Minutes)
- Open Transactions
- OS CPU Privileged Activity
- OS CPU User Time
- OS Memory Used (Percent)
- OS Total Processor Activity
- Packet Errors Per Second
- Packets Received Per Second
- Packets Sent Per Second
- Page Life Expectancy
- Page Lookups Per Second
- Page Reads Per Second
- Page Splits Per Second
- Page Writes Per Second
- Pages Per Second
- Page Life Expectancy
- Procedure Cache Hit Ratio
- Procedure Cache Size
- Processor Queue Length
- Read Ahead Pages Per Second
- Replication Latency (Seconds)
- Replication Subscribed
- Response Time
- SQL Compilations Per Second
- SQL CPU Activity
- SQL Memory Allocated
- SQL Memory Used
- SQL Recompilations Per Second
- System Processes
- System Processes Consuming CPU
- Table Lock Escalations
- Tempdb GAM Wait Time Milliseconds
- Tempdb PFS Wait Time Milliseconds
- Tempdb SGAM Wait Time Milliseconds
- Tempdb Size
- Tempdb Size Percent
- Undistributed Transactions
- Unsubscribed Transactions
- Used Memory (Percent)
- User Connections (Percent)
- User Processes
- User Processes Consuming CPU
• Version Store Generation
• Version Store Size
• VM CPU Ready
• VM CPU Swap Wait
• VM CPU Usage
• VM CPU Usage MHz
• VM Disk Read
• VM Disk Usage
• VM Disk Write
• VM Memory Active
• VM Memory Ballooned
• VM Memory Consumed
• VM memory Granted
• VM Memory Swap In Wait
• VM Memory Swap Out Rate
• VM Memory Swapped
• VM Memory Usage
• VM Net Received
• VM Net Transmitted
• VM Net Usage
• Workfiles Created Per Second
• Worktables Created Per Second

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Apply baseline to multiple instances

SQLdm allows you to set a baseline configuration for a monitored SQL Server instance, and then apply that same baseline to other instances. Simply select the appropriate instances from the Available Servers list, and then click Add > to move them to the Added Servers list. Click OK, and then SQLdm applies this configuration to the other instances.

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. For instances running SQL Server 2008 or higher, query monitoring is enabled by default.

The Query Monitor window allows you to enable or disable query monitoring using the Enable the Query Monitor check box. You can also select the settings that are used:

- Query data collection
- Query Execution Plans collection
- Capturing poorly-performing queries

Query data collection

You can select to collect query data using Extended Events (collect query data using Extended Events radio button) or SQL Trace (Collect query data using SQL Trace radio button). For instances running SQL Server 2008 or higher, the Query Monitor uses by default Extended Events. First introduced in SQL Server 2008, Extended Events provide a new mechanism to capture information about events inside the Database Engine and diagnose performance problems. This functionality is highly efficient and lightweight. For more information about using Extended Events, see the Microsoft document, Extended Events.

To capture query data for monitored pre-SQL Server 2008 instances, enable the SQL Trace collection option. Note that enabling this option can degrade performance on your SQL Server.

Query Execution Plans collection

Query Execution Plans display how the SQL Server Database Engine navigates tables and uses indexes to access or process the data for a query or other DML Statement. To analyze a slow-running query, you can examine the query execution plan and pinpoint the root of the problem.

You can choose to collect or not query execution plans by clicking the Collect Query Plans (SQL Server 2008 and up only) check box in the Query Monitor window. For instances running SQL Server 2008 or higher, this option is enabled by default.
Capturing poorly-performing queries

The types of poorly-performing queries to capture include:

- SQL batches
- SQL Statements
- Stored procedures and triggers

You can also define specific poorly-performing thresholds such as duration, logical disk reads, CPU usage, and physical disk writes.

Access the Query Monitor tab

You can access the Query Monitor tab of the Monitored SQL Server Properties window by right-clicking the appropriate SQL Server instance, and then selecting Properties. Click Query Monitor when SQLdm displays the Monitored SQL Server Properties window.

Diagnosing specific performance issues

Select the events that help you best diagnose the specific issues occurring with query performance on your SQL Server instance. For example, you may want to monitor queries taking a long time to complete, requiring heavy CPU usage, and causing a large number of logical disk reads or physical disk writes, which indicates a memory issue.

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.

Using advanced Query Monitor configuration options

You can choose the applications, databases, and SQL text you want to include or exclude using the options available on the Advanced Query Monitor Configuration window. You can access this window by clicking the Advanced button on the Query Monitor tab.

Configure advanced query monitor options

The Advanced Query Monitor Configuration window allows you to supply filtering criteria such as including or excluding specific applications, databases, and SQL text from the Query Monitor Trace collection. Filtering allows you to improve the results you see in the Query Monitor view and reduce the impact of the Query Monitor trace on your server.

Access the Advanced Query Monitor Configuration window

You can access the Advanced Query Monitor Configuration window (click image to view full size) by right-clicking the appropriate monitored SQL Server instance, and then selecting Properties. Click Query Monitor when SQLdm displays the Monitored SQL Server Properties window, and then click Advanced.
**Using Query Monitor exclusion filtering**

Excluding data from your Query Monitor trace helps filter out information that is not important for the task at hand. SQLdm allows you to filter specific applications, databases, or SQL text and also exclude the SQLdm queries from the results. When adding multiple items to exclude, use a semi-colon (;) between items of the same type. You can also use a percent character (%) as a wildcard.

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 **Exclude** fields, using (;) to separate items and (%) as a wildcard.
4. Click **OK**.

**Using Query Monitor inclusion filtering**

SQLdm allow you to set Query Monitor inclusion filters. You can choose the applications, databases, and SQL text you want to include in the Query Monitor trace and SQLdm stores only queries that match these inclusion filtering criteria. When adding multiple items to include, use a semi-colon (;) between items of the same type. You can also use a percent character (%) as a wildcard.

1. On the Queries view for the server you want to include data in, press the **Configure Query Monitor** button.
2. Click **Advanced**.
3. Enter your filter information in the **Include** fields, using (;) to separate items and (%) as a wildcard.
4. Click **OK**.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
Capturing Autogrow events

Capturing autogrow information allows the associated autogrow alert to provide information about log and data growth on the Alerts view. To access this information, check Enable the Activity Monitor and Capture Autogrow in the Activity Monitor window.

Capturing Blocking events

Capturing blocking information allows the associated blocked alert to provide details on blocking and blocked sessions on the Alerts view. Blocking monitoring is supported on monitored instances running SQL Server 2005 or greater and is dependent on Activity Monitor. To access this information, check Enable the Activity Monitor and Capture Blocking (SQL 2005+) in the Activity Monitor window. Key details provided for blocking and blocked sessions include:

**Blocking Process Details:**
- Session ID
- Hostname
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions

**Blocked Process Details:**
- Session ID
- Hostname
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions
- Wait Resource

**Blocked Process Threshold**

SQL Server 2005+ includes a new SQL Server Profiler event, the Blocked Process report. This event helps identify blocking issues and provides relevant information to solve these issues. In SQLdm, users can take advantage of this feature by setting the Blocked Process Threshold value according to their needs. To access the Blocked Process Threshold option, enable Capture Blocking (SQL 2005+) in the Activity Monitor window. Note that when this option is enabled, the Blocked Process Threshold value is automatically set to 30 (seconds) but can be modified as desired.
Since collecting information about blocked processes in SQL server is resource-intensive, Idera recommends to set the blocked process threshold value to at least 5 (seconds) or your deadlock monitor runs constantly. For more information on this topic, see the Microsoft document Increase or Disable Blocked Process Threshold.

Modify the Activity Monitor Blocked Process threshold automatically changes the Blocked Process threshold value in your monitored SQL Server instance.

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.

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.

You can access the Replication tab by right-clicking the appropriate monitored SQL Server instance, and then selecting Properties. Click Replication when SQLdm displays the Monitored SQL Server Properties window.
Edit table statistics options

SQLdm lets you collect statistics and perform trend analysis over time. Table Statistics allow you to designate the days and times SQLdm collects table growth and reorganization statistics. Because statistics collection can negatively impact performance, SQLdm allows you to designate an individual Table Statistics collection 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:

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

Access the Table Statistics tab

You can access the Table Statistics tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting Properties. Click Table Statistics when SQLdm displays the Monitored SQL Server Properties window.

Schedule table statistics collection

Table Statistics collection is a resource-intensive process. Schedule your collection time for non-peak hours to reduce the possibility of your users experiencing decreased performance.

The SQLdm Collection Service may not include fragmentation data for a particular table for any of the following reasons:

- Table statistics collection has not occurred.
- The database hosting the table is 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 identifies and resolves SQL Server performance problems before they happen. Learn more > >
Schedule maintenance mode

Maintenance mode allows you to temporarily stop alert generation and performance metric collection for the time period that your SQL Server instance is offline. This pause allows you to perform necessary maintenance on a SQL Server without causing SQLdm to send numerous misleading alerts.

**Access the Maintenance Mode tab**

You can access the Maintenance Mode function by right-clicking the appropriate monitored SQL Server instance, and then selecting Maintenance Mode. This will prompt three command options: Enable, Disable and Schedule.

The **Enable** command puts the SQL Server instance into maintenance mode, stopping alert generation and collection of some performance metrics.

The **Disable** command allows SQLdm to resume performance statistics collection and alert generation for the selected SQL Server instance.

The **Schedule** command prompts the monitored SQL Server Properties dialog, where under the Maintenance Mode tab it is possible to select a specific maintenance mode setting.

**About mass maintenance mode**

Users looking to apply maintenance mode modifications to a large number of servers at once can run the Maintenance Mode command by right-clicking on the following: Server Tree, a specific tag, a particular server view and Tools.

**Available maintenance mode settings**
You can choose from the following options when setting a monitored SQL Server instance to maintenance mode:

- **Never**
  
  The server is never in maintenance mode and always generates alerts.

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

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

- **Recurring every month at the specified time**
  
  Set the server to enter maintenance mode at regular intervals, for the month, time, and duration you select.

- **Occurring once at the specified time**
  
  The server enters maintenance mode at a specified date and time, and leaves maintenance mode at a specified date and time.

---

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. [Learn more](#)

---

**Configure OS metrics monitoring**

SQLdm requires that you enable either OLE automation or direct Windows Management Interface (WMI) procedures for monitoring OS metrics. In addition, if your SQL Server instance includes enabled Lightweight Pooling, SQLdm cannot collect OS metrics information. If SQLdm cannot provide OS metrics, a message displays on the Dashboard and Services Summary views providing more information.
SQLdm collects metrics that track the performance of the computers that host your monitored SQL Server instances. These metrics include:

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

**Access the OS Metrics tab**

You can access the OS Metrics tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting Properties. Click OS Metrics when SQLdm displays the Monitored SQL Server Properties window.

**Disabling OS metric collection**

OS metrics collection is not required in SQLdm. However, if you disable this feature, SQLdm ceases to collect OS metrics and does not raise any previously-associated alerts. You can easily re-enable OS metric collection at any time if you decide that you want to continue collecting these metrics.

**Permissions necessary for direct WMI collection**

It is preferred that the WMI user is a local administrator on the monitored SQL Server instance. However, if you do not want to grant administrator access, use the following steps to configure remote WMI access in Microsoft Windows:

1. Add the user account to the Performance Log Users and Distributed COM users groups.
2. Grant privileges to WMI.

You also may need to add the WMI user account to the following policies:

- Act as part of the operating system
- Log on as a batch job
- Log on as a service
- Replace a process level token

For more information about using a direct WMI connection, see the Microsoft document, [Securing a Remote WMI Connection](#).

**Configure WMI to run out-of-process**

Edit your specific registry settings to allow WMI to run out-of-process.

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**
  Select the Collect Operating System data using OLE Automation option to enable OLE Automation procedures.

- **OLE Automation Procedures Unavailable**
  The stored procedures associated with OLE Automation are missing on the SQL Server instance. SQLdm cannot capture OS metrics until these stored procedures are available on the SQL Server instance.

- **WMI Service Not Running**
  The WMI Service on the monitored SQL Server instance may not be running. SQLdm cannot collect OS metrics until this service is started.

- **Lightweight Pooling Enabled**
  This is an advanced feature used in symmetric multiprocessing environments in SQL Server. SQLdm cannot collect OS metrics until this option is disabled on your monitored SQL Server instances.

**Configure WMI timeouts**

SQLdm allows you to configure the WMI timeout value. In some environments customers may experience WMI timeouts when their machine is too busy to respond in a timely manner. Although WMI timeouts should not be ignored and often point to an environmental issue, you can control this value when the behavior displayed is normal in your environment. The default WMI timeout value in SQLdm is of 90 seconds to reduce the...
amount of intermittent errors.

To configure the WMI timeout value:
1. Navigate to the SQLdm install directory
2. Open SQLdmCollectionService.exe.Config
3. Modify the <CollectionService> node under the <Services> node
4. Add wmiQueryTimeOut="300" as a parameter to this node
5. Save and close file
6. Restart the collection service

Sample configuration entry:

```xml
<Services>
  <CollectionService instanceName="Default" servicePort="5167"
    managementServiceAddress="Server" managementServicePort="5166"
    heartbeatIntervalSeconds="180" wmiQueryTimeOut="300"/>
</Services>
```

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
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 to use wait monitoring**

Wait Monitoring helps you diagnose congestion or blocks on your system. You should enable Wait Monitoring for the time period required to diagnose specific issues you experience. When your diagnosis is complete, disable Wait Monitoring.

**Access the Wait Monitoring tab**

You can access the Wait Monitoring tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting Properties. Click Wait Monitoring when SQLdm displays the Monitored SQL Server Properties window.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

Configure advanced query wait statistics options

The Advanced Query Wait Statistics Configuration window allows you to filter out applications, databases, SQL text from the Query Waits Statistics collection, and exclude specific SQLdm queries as well as limit the amount of rows to be collected at a time. Filtering allows you to improve the results you see in the Query Waits view and reduce the impact of the Query Wait trace on your server.

**Access the Advanced Query Wait Statistics Configuration window**

You can access the Advanced Query Wait Statistics Configuration window by right-clicking the appropriate monitored SQL Server instance, and then selecting Properties. Click Wait Monitoring when SQLdm displays the Monitored SQL Server Properties window, and then click Filter Options.

**Using Query Wait filtering**

Excluding data from your Query Waits Statistics collection helps filter out information that is not important for the task at hand. SQLdm allows you to filter specific applications, databases, SQL text and also exclude the SQLdm queries from the results or limit the amount of rows to be collected at a time. When including multiple items to exclude, use a semi-colon ( ; ) between items of the same type. You can also use a percent character ( %) as a wildcard.

To exclude data from your Query Wait Statistics collection while in the Queries view:

1. On the Queries Waits view for the server you want to exclude data from, press the Configure button.
2. Click Filter Options.
3. Enter your filter information in the provided fields, using ( ; ) to separate items and (%) as a wildcard.
4. Click OK.

**Configure virtualization**

The Virtualization tab of the Monitored SQL Server Properties window displays the name of the host server on which the selected monitored SQL Server instance is running. If this is not the correct host server, click VM Configuration to make the necessary changes to connect to the appropriate virtual machine.

**Access the Virtualization tab**

You can access the Virtualization tab of the Monitor SQL Server Properties window by right-clicking the appropriate monitored SQL Server instance, and then selecting Properties. Click Virtualization when SQLdm displays the Monitored SQL Server Properties window.

**Monitor sessions**

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.

**Access the Sessions tab**

SQLdm provides two paths to access the Sessions tab. The first access path is by clicking the appropriate monitored SQL Server instance, and then clicking the Sessions tab. The second access path is by expanding the Servers tree, and then clicking Sessions for the appropriate instance.

**Alerts associated with the Sessions tab**

The following alerts are associated with the Sessions tab:

- **Blocked Sessions (Count)**
  - Is the number of sessions blocked by other sessions holding requested locks.

- **Blocking Session Wait Time (Seconds)**
  - Is the amount of time (in seconds) that a SQL Server session is blocking other sessions.

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

- **Deadlock**
  - Is when two processes are waiting for a resource held by the other process.

- **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.

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

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

- **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.

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

---

*SQL Diagnostic Manager* identifies and resolves SQL Server performance problems before they happen. Learn more >>

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

**Get sessions performance summary**

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.

**Access the Summary view**

You can open the Summary view of the SQLdm Sessions tab by selecting the appropriate SQL Server instance, and then clicking Sessions > Summary.

**Response Time chart**

The Response Time chart displays the response time (in milliseconds) it takes a Select SQL statement to go to the SQL Server instance and
You can see the specific time and response time of any data point in the chart by hovering your mouse 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.

**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.

**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.

**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.

**Understanding your chart data and refresh interval**

SQLdm charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQLdm displays your chart data, see Charts.

**Get sessions performance details**

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, show your query history, trace the session, kill the session, print the associated statistics, or export statistics to Excel.

When you select a session, SQLdm displays the statistics for that session in the Details section at the bottom of the window. This section includes in-depth information regarding the session connection, usage, lock, and tempdb usage. If you want to copy the Last Command information to the Windows clipboard, right-click in that area, and then click Copy.

**Access the Details view**

You can open the Details view of the SQLdm Sessions tab by selecting the appropriate SQL Server instance, and then clicking Sessions > Details.

**Analyze locked sessions**

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.

Use 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.

**Access the Locks view**

You can open the Locks view of the SQLdm Sessions tab by selecting the appropriate SQL Server instance, and then clicking Sessions > Locks.

**Available SQL Server lock objects**

The Lock Statistics chart includes the following objects:

- **AllocUnit**
  - Represents a lock on an allocation unit.
- **Application**
  - Represents a lock on an application-specific resource.
- **Database**
  - Represents a lock on a database, including all objects in the database.
- **Extent**
  - Represents a lock on a contiguous group of eight pages.
- **File**
  - Represents a lock on a database file.
- **HoBT**
  - (Heap or BTree) Represents a lock on a heap of data pages, or on the BTree structure of an index.
- **Key**
  - Represents a lock on a row in an index.
- **Latch**
  - Represents a lock on a latch.
- **Metadata**
  - Represents a lock on a piece of catalog information, also called metadata.
- **Object**
  - Represents a lock on an item such as a table, stored procedure, or view, including all data and indexes. The object represents anything that has an entry in sys.all_objects.
- **Page**
  - Represents a lock on an 8 KB page in a database.
- **RID**
  - (Row ID) Represents a lock on a single row in a heap.
- **Table**
  - Represents a lock on a table.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
Analyze blocked sessions

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 then block other sessions.

You can use the History Browser to view past performance and go back to the point in time to view the sessions 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.

Deadlocks have no lead blockers and are circular in nature. For this reason, SQLdm does not display deadlocks in the Blocking Tree but does display instances of deadlocks in the Blocking Chart. In addition, SQLdm does not count self-blocking sessions as blocked or blocking sessions.

Access the Blocking view

You can open the Blocking view of the SQLdm Sessions tab by selecting the appropriate SQL Server instance, and then clicking Sessions > Blocking.

Trace 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

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.

Blocking Sessions Report

The Blocking Sessions Report (Block Reports) displays blocking and blocked sessions’ information. To access the Blocking Sessions Report in the Alert view, select the Show Block Details option in the right-click context menu. You can also access the same information on the Blocking View by selecting a specific SQL Server instance, pointing to Sessions, and then selecting the Blocking ribbon.

Key information provided for blocking and blocked sessions includes:
Blocking Process Details:

- Session ID
- Host Name
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions

Blocked Process Details:

- Session ID
- Host Name
- User Name
- Application
- Database
- Last Batch Started
- Transaction ID
- Open Transactions
- Wait Time (ms)
- Wait Resource

This feature allows you to export statistics in XML format.

To retrieve the blocking sessions report for SQLdm, select the "Non-Query activities>Capture Blocking (SQL 2005+)" check box in the Activity Monitor Tab.

Deadlock Sessions Report

The Deadlock Sessions Report (Deadlock Reports) displays deadlock sessions information. To access the Deadlock Sessions Report in the Alert view, select the Show Block Details option in the right-click context menu. You can also access the same information on the Blocking View by selecting a specific SQL Server instance, pointing to Sessions, and then selecting the Blocking ribbon.

Key information provided for deadlock sessions includes:

Deadlock Process Details:

- Session ID
- Status
- Host Name
- User Name
- Execution Context
- Application
- Database
- Last Batch Started
- Last Batch Completed
- Last Transaction Started
- Transaction ID
- Transaction Name
- Open Transactions
- Wait Time (ms)
- Wait Type
- Wait Resource

This feature allows you to export statistics in XDL format.
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. Use the Query Monitor when diagnosing poor query performance on your SQL Server instance. You should monitor queries if your SQL Server instance performance is poor, as a way to diagnose potential query issues.

Access the Queries tab

SQLdm provides two paths to access the Queries tab. The first access path is by clicking the appropriate monitored SQL Server instance, and then clicking the Queries tab. The second access path is by expanding the Servers tree, and then clicking Queries for the appropriate instance.

Use 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. Fields with SQL text hold 255 characters of text. To view the full SQL text, right-click the field and then select the appropriate option. 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 are different on different desktops and in different sessions on the same desktop.

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.

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

The Sigma (Σ) 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.

Understand 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 are automatically reduced to the signatures (parameters are stripped) and grouped accordingly.

You can also view signatures in the Advanced query signature view of the SQLdm web console.

Statement Mode

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

Use Filter Settings to customize your 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.
The Include incomplete queries filter in the Signature Mode and Statement Mode views is disabled by default. If you would like to include data with null values in these views, check the Include incomplete queries check box in the Filters section.

You can hide and unhide the Filter and List areas of the Query Monitor views in the ribbon.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

View the query monitor signature mode

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.

If you want to prevent certain queries from appearing on the Query Monitor views, click Configure Query Monitor > Advanced and enter the applications, databases, or SQL text you want to exclude from the Query Monitor collection. You can also configure these options directly in the SQLdm web console.

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.

If Query Monitor is disabled but Activity Monitor is enabled, SQL statements appearing in the Event Occurrences grid come from the Blocking Session collector. The collector only populates the following fields: Query, SQL text, Occurrences, Event Type, and Average Deadlocks.

Access the Signature Mode view

You can open the Signature Mode view of the SQLdm Queries tab by selecting the appropriate SQL Server instance, and then clicking Queries > Signature Mode.

You can also access the Query Signature Mode view through the Query tab of the SQLdm web console. For more information on the new advanced query views, see View your SQL Server queries information.

Sigma () character

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.

Understanding cell colors

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
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.

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

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.

Access the Statement Mode view

You can open the Statement Mode view of the SQLdm Queries tab by selecting the appropriate SQL Server instance, and then clicking Queries > Statement Mode.

Sigma () character

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.

Understanding cell colors

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >

View the query history

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.

Access the Query History view

You can open the Query History view of the SQLdm Queries tab by selecting the appropriate SQL Server instance, and then clicking Queries > Query History.

View Query History

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.

If Query Monitor is disabled but Activity Monitor is enabled, SQL statements appearing in the Event Occurrences grid come from the Blocking Session collector. The collector only populates the following fields: Occurrence, SQL text, Event Type, and Deadlocks.

### Sigma (σ) character

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.

### Understanding cell colors

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.

### View query waits

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.

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.

**Access the Query Waits view**

You can open the Query Waits view of the SQLdm Queries tab by selecting the appropriate SQL Server instance, and then clicking **Queries > Query Waits**.

### Monitor resources

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 result from poor database application design or point to other system processes using valuable resources needed by SQL Server.

**Access the Resources tab**

SQLdm provides two paths to access the Resources tab. The first access path is by clicking the appropriate monitored SQL Server instance, and then clicking the Resources tab. The second access path is by expanding the Servers tree, and then clicking Resources for the appropriate instance.

**Alerts associated with the Resources tab**

The following alerts are associated with the Resources tab:

- **Disk Reads / Seconds Per Disk**
The number of disk reads per second, per disk

**Disk Transfers / Seconds Per Disk**

The number of disk transfers per second, per disk.

**Disk Writes / Seconds Per Disk**

The number of disk writes per second, per disk.

**OS Average Disk Queue Length**

The average number of both read and write requests queued for all the disks 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 Memory Usage (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 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 Processor Queue Length**

Is the number of ready threads in the processor queue 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 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.

**Page Life Expectancy**

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

**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.

**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.

---

_sql_diagnostic_manager_

_sql DIAGNOSTIC MANAGER identifies and resolves SQL Server performance problems before they happen. Learn more >>_

---

Get the resource performance summary

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.

**Access the Resources Summary view**

You can open the Summary view of the SQLdm Resources tab by selecting the appropriate SQL Server instance, and then clicking Resources > Summary.
Use the CPU Usage chart

The CPU Usage chart helps you 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.

SQL Server is 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, the computer requires your attention. High usage may indicate the need to upgrade your CPU or add more processors.

Use the Call Rates chart

The Call Rates chart helps you determine why you have high CPU usage. The Call Rates chart displays the following metric statistics:

- **Batches**: Displays the rate at which SQL Server issues SQL Statement batches.
- **Compiles**: Displays the rate at which SQL Server compiles SQL Statements into the Procedure Cache.
- **Re-Compiles**: Displays the rate at which SQL Server recompiles stored SQL Statements into the Procedure Cache.

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. Consider running the Query Monitor as a way of diagnosing query performance issues. You can enable the Query Monitor on the Query Monitor tab of the Monitored SQL Server Properties window.

- **Transactions**: Displays the number of SQL Statement transactions affecting the Procedure Cache.

Use the Memory Usage chart

The Memory Usage chart allows you to analyze the way in which your SQL Server instance uses memory. 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.

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 you need more physical memory.

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.

An acceptable Buffer Cache and Procedure Cache percentage is 90 percent or higher. If this number is consistently lower, analyze your memory settings and add more memory if necessary.

Use the Disk Busy chart

Use the Disk Busy chart to analyze bottlenecks in your file system.

If your Disk Busy is consistently over 90 percent you may need to 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.

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 SQL Server Physical I/O chart displays the following reads and writes:

- **Checkpoint Writes**: Save time during recovery by creating a point at which all dirty pages are guaranteed written to disk.
Lazy Writer Writes
Create more available buffers for the system by clearing all unnecessary buffers.

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

Page Writes
Are disk writes where the requesting session must wait for certain operations to finish.

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.

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

Understanding your chart data and refresh interval
SQLdm charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQLdm displays your chart data, see Charts.

Get CPU performance details
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.

Access the CPU view
You can open the CPU view of the SQLdm Resources tab by selecting the appropriate SQL Server instance, and then clicking Resources > CPU.

CPU tab charts
The following charts are available in the CPU tab:

  **CPU Usage**
  Displays the percentage of a CPU consumed by your SQL Server, virtual machine, and host 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.

Understanding your chart data and refresh interval
SQLdm charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQLdm displays your chart data, see Charts.

Get memory performance details
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.

**Access the Memory view**

You can open the Memory view of the SQLdm Resources tab by selecting the appropriate SQL Server instance, and then clicking Resources > Memory.

**Memory tab charts**

The following charts are available on the Memory tab:

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

- **VM/Host Memory Usage**
  Displays the memory usage by your virtual machine and host as a percentage of total memory available.

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

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

- **VM Memory Areas**
  Displays the amount of memory allocated to and used by SQL Server to the total memory consumed on this virtual machine.

- **Host Memory Areas**
  Displays the amount of memory allocated to and used by SQL Server to the total memory consumed on this host server.

- **Cache Hit Ratios**
  Displays the hit ratios 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.

- **Page Life Expectancy (sec)**
  Provides the time in seconds that a page remains in the cache or buffer pool.

**Understanding your chart data and refresh interval**

SQLdm charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQLdm displays your chart data, see Charts.

---

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. [Learn more > >](https://www.idera.com/products/database-performance-monitoring-sql-dm)
You must have OS metrics enabled to show the information in the charts on the Disks tab.

**Access the Disk view**

You can open the Disk view of the SQLdm Resources tab by selecting the appropriate SQL Server instance, and then clicking Resources > Disk.

**Disk tab details**

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

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

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

- **Average Disk ms/Read Per Disk**
  Displays the average length of time (in milliseconds) for a disk to read during the sample interval.

- **Average Disk ms/Transfer Per Disk**
  Displays the average length of time (in milliseconds) for a disk transfer during the sample interval.

- **Average Disk ms/Write Per Disk**
  Displays the average length of time (in milliseconds) for a disk write during the sample interval.

- **VM Disk Usage**
  Displays the amount of data read from and written to disk in KB per second for this virtual machine during the sample interval. For additional information about how SQLdm works with virtual machines and collects metrics, see [How SQLdm works with a virtual environment](http://example.com/virtualenv).

- **Host Disk Usage**
  Displays the amount of data read from and written to disk in KB per second for the host server during the sample interval. For additional information about how SQLdm works with virtual machines and collects metrics, see [How SQLdm works with a virtual environment](http://example.com/virtualenv).

**Understanding your chart data and refresh interval**
SQLdm charts provide data whether collected real time or as part of a scheduled collection event. For additional information about how SQLdm displays your chart data, see Charts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

View file activity

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 on which the files reside. 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 Server Properties window.

Activity lights

Read (green) 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.

Access the File Activity view

You can open the File Activity view of the SQLdm Resources tab by selecting the appropriate SQL Server instance, and then clicking Resources > File Activity.

Database Files + Remaining Disk Files

Database Files + Remaining Disk Files chart depicts the activity of all the non-SQL database files and files not included in the File Activity window.

Use the filter to display vital database activity

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.

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
Manage your procedure cache

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 is used since inception. The References column displays the number of other cache objects referencing the associated cache object, with “1” as the base.

Access the Procedure Cache view

You can open the Procedure Cache view of the SQLdm Resources tab by selecting the appropriate SQL Server instance, and then clicking Resources > Procedure Cache.

Procedure Cache tab charts

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.
Available wait categories

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Monitor databases

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.

The following concepts are important to understand when using the Databases 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 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.

Access the Databases tab

SQLdm provides two paths to access the Databases tab. The first access path is by clicking the appropriate monitored SQL Server instance, and then clicking the Databases tab. The second access path is by expanding the Servers tree, and then clicking Databases for the appropriate instance.

Alerts associated with the Databases tab

The following alerts are associated with the Databases tab:

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.

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.

**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.

**Table Fragmentation (Percent)**

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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

### 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 using click+Shift and click+Ctrl to compare several databases.

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

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.

**Access the Database Summary view**

You can open the Summary view of the SQLdm Databases tab by selecting the appropriate SQL Server instance, and then clicking Databases > Summary.

**Database Summary tab charts**

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

**Capacity Usage**

You can choose to view how data or the log is used on your database. The drop-down list allows you to choose to view the data or log 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.

---

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Monitor AlwaysOn Availability Groups

The AlwaysOn Availability Groups view allows you to monitor availability groups, availability databases, and availability replicas.

If you have configured the AlwaysOn Availability Groups feature on your SQL 2012 Server, SQLdm automatically recognizes the availability databases and starts to monitor them.

AlwaysOn Availability Groups are part of an integrated solution, introduced in SQL Server 2012 with the goal of achieving the highest level of data availability and disaster recovery for organizations. Availability Groups grant DBAs the ability to automatically or manually failover a group of databases as a single unit with support for up to four secondary replicas. For additional information on availability groups, see the Microsoft document, Overview of AlwaysOn Availability Groups (SQL Server).

**Access the AlwaysOn Availability Groups view**
You can open the AlwaysOn Availability Groups view by clicking the appropriate monitored SQL Server instance, and then clicking Databases > Availability Group.

**Database AlwaysOn Availability Groups statistics**

In this view, you can see the health of all your availability groups, availability databases and monitored availability replicas. The database table provides the following information for each availability group (as seen in the view by default):

- **Database Name**
  Displays the name of the availability database.

- **Group Name**
  Displays the name of the availability group for which the connected server instance hosts a replica.

- **Replica Name**
  Displays the name of the server instance that hosts the availability replica.

- **Failover Mode**
  Displays the failover mode for which the availability replica is configured. The possible failover modes are automatic or manual.

- **Availability Mode**
  Describes if data synchronization will use either the synchronous-commit or asynchronous-commit availability mode.

- **Replica Role**
  Indicates the current role of the availability replica. Possible values are primary or secondary.

- **Synchronization Health**
  Displays database synchronization state in the availability group. Possible values include: not healthy, partially healthy, and healthy.

- **Redo Queue Size (KB)**
  Displays the amount of log records from log files in KB that need redoing in the secondary replica to complete synchronization.

- **Redo Rate (KB/s)**
  Rate in KB per second at which log records are redone in the secondary database to complete synchronization.

- **Log Send Queue Size (KB)**
  Amount of log records in KB needed to ship to the secondary replica to complete synchronization.

- **Log Send Rate (KB/s)**
  Indicates the rate in KB per second at which log records are being sent to the secondary replica to complete synchronization.

- **Listener DNS Name**
  The network name of the availability group listener.

- **Listener IP Address**
  The IP address reserved for the availability group listener. If the nodes of your Windows cluster are on different subnets you will need an IP address for each of those subnets.

- **Failover Readiness**
  Indicates whether the secondary database is synchronized with the corresponding primary database. Possible values include: True (the database is marked as synchronized and is ready for a failover) and False (the database is not synchronized and not ready for a failover).

- **Synchronization Database Status**
  Indicates whether the availability database is currently synchronized with the primary replica. Possible synchronization states are: not synchronizing, synchronizing, synchronized, reverting, and initializing.

- **Database Status**
  Description of the database state of the availability replica. Possible values include: online, restoring, recovering, recovery_pending, suspect, emergency, and offline.

- **Suspended Status**
An AlwaysOn availability database is suspended. Possible values are true or false.

**Last Hardened Time**
Indicates the time when the log-block identifier was received for the last hardened LSN on the secondary replica.

**Operational Status**
Indicates the current operational state of the secondary replica. Possible state values include: pending failover, pending, online, offline, failed, failed no quorum, and NULL.

**Connection Status**
Indicates whether a secondary replica is currently connected to the primary replica. Possible values include: connected and disconnected.

**Synchronization Performance (s)**
Indicates the time in seconds it takes to synchronize between the primary and secondary replicas.

**Estimated Data Loss (s)**
Indicates the time difference in seconds of the last transaction log record in the primary replica and secondary replica. If the primary replica fails, all transaction log records within the time window will be lost.

**Estimated Recovery Time (s)**
Indicates the time in seconds it takes to redo the catch-up time. The catch-up time is the time it will take for the secondary replica to catch up with the primary replica.

Additional columns available through the Column Chooser:

**Database ID**
Identifier of the database, unique within an instance of SQL Server.

**FileStream Send Rate (KB/s)**
Indicates the rate of the FileStream in KB per second at which transactions are being sent to the secondary replicas.

**Group ID**
Unique identifier (GUID) of the availability group.

**Last Connect Error #**
Number of the last connection error.

**Last Connect Error Description**
Description of the last connection error.

**Last Connect Error Time**
Timestamp of the last connection error.

**Listener Port**
The TPC port used by the availability group listener.

**Replica ID**
Identifier of the availability replica within the availability group.

You can hide and unhide additional columns in the Availability Groups table by right clicking on any header in the grid and selecting a column from the Column Chooser dialog.

**Availability Groups charts**
The AlwaysOn Availability Groups view includes charts that display the queue size and transfer rates of different availability groups.

**Queue Size chart**
Provides users with graphical details about queues in the AlwaysOn Availability Groups feature. This chart is a stacked bar chart for the "Log Send Queue Size" and "Redo Queue Size."
Transfer Rates chart

Provides users with graphical details about data transfer rates for redo and logs. This chart is a line chart for the "Log Send Rate" and "Redo Rate."

Availability Groups alerts

To view a list of all available alerts, see Metric alerts.

Understand the view colors

The AlwaysOn Availability Groups view uses a color-based alert system. The following colors are associated with a status and action within SQLdm:

<table>
<thead>
<tr>
<th>Color</th>
<th>Alert Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>OK</td>
<td>Acceptable threshold where SQLdm does not generate an alert.</td>
</tr>
<tr>
<td>Blue</td>
<td>Informational</td>
<td>Informational threshold where SQLdm generates an informational alert.</td>
</tr>
<tr>
<td>Yellow</td>
<td>Warning</td>
<td>Warning threshold where SQLdm generates a warning alert.</td>
</tr>
<tr>
<td>Red</td>
<td>Critical</td>
<td>Critical threshold where SQLdm generates a critical alert.</td>
</tr>
</tbody>
</table>

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Get the tempdb status summary

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.

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.

Access the Tempdb Summary view

You can open the Tempdb Summary view of the SQLdm Databases tab by selecting the appropriate SQL Server instance, and then clicking Data bases > Tempdb Summary.

Tempdb Summary tab charts

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 collects 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:
Monitor your tempdb database

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.

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.

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)
- Shared global allocation map (SGAM)

Latch contention of this sort is usually an indication that you should create more tempdb data files. In some situations, using Trace Flag 1118 may also alleviate tempdb contention.

Available alerts

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)

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
You can use the Database Configuration view to analyze configuration differences between databases or databases with incorrect configuration settings.

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

**Access the Configuration view**

You can open the Configuration view of the SQLdm Databases tab by selecting the appropriate SQL Server instance, and then clicking Databases > Configuration.

**Null values in the Collation and Version columns**

If the database is not started and AUTO_CLOSE is set to ON, these columns are set to null until the database is started.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

### 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.

**Database alerts and statuses**

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>The database is in standby mode.</td>
</tr>
<tr>
<td>OK</td>
<td>Read Only</td>
<td>The database is in read-only mode.</td>
</tr>
<tr>
<td>OK</td>
<td>Single User</td>
<td>The database is in single-user mode.</td>
</tr>
<tr>
<td>OK</td>
<td>Restore Mirror</td>
<td>Restoring the database mirror.</td>
</tr>
<tr>
<td>OK</td>
<td>Undetermined</td>
<td>No information is available one this database.</td>
</tr>
<tr>
<td>OK</td>
<td>Cleanly Shut Down</td>
<td>The database is shut down and needs no recovery.</td>
</tr>
<tr>
<td>OK</td>
<td>DBO Use Only</td>
<td>The database is in a dbo-use mode (restricted_user).</td>
</tr>
<tr>
<td>Warning</td>
<td>Restoring</td>
<td>Restoring the database.</td>
</tr>
<tr>
<td>Warning</td>
<td>Recovering</td>
<td>Recovering the database.</td>
</tr>
<tr>
<td>Warning</td>
<td>Recovery Pending</td>
<td>The database recovery is not yet begun.</td>
</tr>
<tr>
<td>Warning</td>
<td>Offline</td>
<td>The database is offline.</td>
</tr>
<tr>
<td>Critical</td>
<td>Suspect</td>
<td>The database integrity is suspect.</td>
</tr>
<tr>
<td>Critical</td>
<td>Emergency Mode</td>
<td>The database is in emergency mode.</td>
</tr>
<tr>
<td>Critical</td>
<td>Inaccessible</td>
<td>The SQLdm Monitoring Service is unable to open the database. A serious issue such as a disk failure can cause an inaccessible database.</td>
</tr>
</tbody>
</table>
Change the default alert status level

Alerting is based on the most critical status associated with a database at the time of collection.

- 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.

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

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.

- Database growth statistics are collected every scheduled refresh.

The Files view is an important tool that displays 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 of SQL Server performance bottlenecks.

For Example: If the % 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.

Access the Files view

You can open the Files view of the SQLdm Databases tab by selecting the appropriate SQL Server instance, and then clicking Databases > Files.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Get file statistics

Select any database in the list to display the associated file statistics in the Data Files tab of the Files view. 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

You can select multiple databases as well as multiple files to compare file statistics.

Access the Data Files tab

You can open the Data Files tab of the SQLdm Databases Files view by selecting the appropriate SQL Server instance, and then clicking Databases > Files. The Data Files tab is the default view.
Get file group information

The File Groups tab of the Files view 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

Select multiple databases and multiple file groups to compare how they are used. This can help you quickly spot issues with your file usage.

Access the File Groups tab

You can open the File Groups tab of the SQLdm Databases Files view by selecting the appropriate SQL Server instance, clicking Databases > Files, and then clicking the File Groups tab.

View transaction log information

The Transaction Logs tab of the Files view 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

Log files in high transaction rate environments can occupy multiple disks. The Transaction Logs tab allows you to easily monitor these configurations.

Access the Transaction Logs tab

You can open the Transaction Logs tab of the SQLdm Databases Files view by selecting the appropriate SQL Server instance, clicking Databases > Files, and then clicking the Transaction Logs tab.

Get disk information

The Disks tab of the Files view provides a graphical representation of your current disk usage. Select the drop-down list above the chart to choose from the following chart types:

- Megabytes
- Percent

You must have OS metrics enabled to show the information in the charts on the Disks tab.

Access the Disks tab

You can open the Disks tab of the SQLdm Databases Files view by selecting the appropriate SQL Server instance, clicking Databases > Files, and then clicking the Disks tab.
Analyze backups and restores

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.

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.

Access the Backups & Restores view

You can open the Backups & Restores view of the SQLdm Databases tab by selecting the appropriate SQL Server instance, and then clicking Databases > Backups & Restores.

Analyze tables and indexes

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.

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.

Access the Tables & Indexes view

You can open the Tables & Indexes view of the SQLdm Databases tab by selecting the appropriate SQL Server instance, and then clicking Databases > Tables & Indexes.

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 also lists the data distribution data values and occurrences of each value.
Table Size Details chart

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.

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.

Indexes tab

The Indexes tab displays all indexes defined for the selected table and an overview of each index.

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 the Microsoft document, Tables and Indexes.

⚠️ The Collection Service may not gather fragmentation data for a particular table because:

- Table statistics collection has not occurred
- The database hosting the table was 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

Update table statistics

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.

Rebuild indexes

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.

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.

If you are rebuilding a very large index, the SQLdm Console checks 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.

If you have enabled database mirroring on your SQL Server, the SQLdm Database Mirror Monitoring view is enabled by Default.

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.

Access the Mirroring view

You can open the Mirroring view of the SQLdm Databases tab by selecting the appropriate SQL Server instance, and then clicking Databases > Mirroring.

Database mirroring statistics

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.
- **Partner Instance**
  - The name of the SQL Server instance that is the mirror partner.
- **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 how the user established which SQL Server instance hosts the principal and which SQL Server instance hosts the mirrored database.
- **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).

Available details for a specific session

Select a session, and then click the Status tab to view the following details. You can also view historical information for a selected session using the History tab. This information includes the history of the metrics over time for both the local and remote partner.

- **Unsent log (in kilobytes)**
  - Size of the log waiting in the send queue on the principal, displayed in kilobytes.
- **Oldest unsent transaction (dd.hh:mm)**
  - Date and time of the oldest unsent transaction in the send queue on the principal.
- **Time to send log (estimated)**
  - Estimated number of minutes required by the principal to send the log waiting in the send queue to the mirror based on the current send rate. Note that this rate is affected by the rate of incoming transactions.

**Send rate (in kilobytes per second)**
Rate at which the principal is sending transactions to the mirror, displayed in kilobytes per second.

Rate of new transactions (in kilobytes per second)
Rate at which transactions are added to the unsent queue on the principal.

Unrestored log (in kilobytes)
Size of the log waiting in the redo queue on the mirror, in kilobytes.

Time to restore log (estimated)
Estimated number of minutes required to apply the log in the redo queue to the mirror database.

Current restore rate (in kilobytes per second)
Rate at which transactions in the redo queue are applied to the mirror database, in kilobytes per second.

Mirror commit overhead
Size of the log waiting in the redo queue on the mirror, in kilobytes. This delay is due to mirror operations and is the first indication that the traffic exceeds the capabilities of the hardware.

Time to send and restore all current log (estimated)
Estimated number of minutes required to process all outstanding transactions on both the principal and mirror.

Operating mode
Displays one of the following operating modes for the current database mirroring session: High availability with automatic fail-over (has a witness, synchronous), High availability without automatic fail-over (no witness, synchronous), or High performance (no witness, asynchronous).

Witness Address
Network address of the witness SQL Server instance.

Mirror Address
Network address of the mirror SQL Server instance.

Principal Address
Network address of the principal SQL Server instance.

Available actions in the Database Mirror Monitoring view
Each row of the mirrored databases table provides a menu of actions you can perform:

Failover
Select Fail over to partner 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 failover is 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 or Resume Monitoring
Use Suspend/resume session 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 "Failed over" or "Normal"
Select Mark session as "failed over" or Mark session as "normal" operational status to define the status of a mirror relationship. Once you have set a preferred operational state, SQLdm alerts you if the configuration differs from your selection.

Clear the role preference for this mirroring session
Use Clear preferred role for this session to clear the preferred role for the session and stop receiving alerts for differences between the current configuration and the Normal status.

Available alerts for mirrored databases
You can configure the following alerts 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 occurred on a mirrored database on the monitored server.

**Mirroring Oldest Unsent Transactions**
- 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 occurred in terms of the preferred operational status.

**Mirror Status**
- Status of the mirrored database such as synchronized, suspended or disconnected.

**Mirroring Unrestored Log**
- 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 are not yet applied.

**Mirroring Unsent Log**
- Size of the unsent log in the send queue on the principal in kilobytes.

**Mirroring Witness Connection**
- If a witness is configured, this alert displays whether the witness is connected.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

### 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 (Note that 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

The Start and Stop service actions are not supported for virtual SQL Server instances or 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.

### Access the Services tab

SQLdm provides two paths to access the Services tab. The first access path is by clicking the appropriate monitored SQL Server instance, and then clicking the Services tab. The second access path is by expanding the Servers tree, and then clicking the Services for the appropriate instance.

### Alerts associated with the Services tab

The following alerts are associated with the Services tab:

**Cluster Active Node**
- The active node of the cluster is a non-preferred node. You can configure the preferred node under Server Properties > Cluster Settings.

**Cluster Failover**
- There is a change in the active node. This alert remains active for the amount of time specified in the advanced settings for that alert.

**Distribution Latency (Seconds)**
Time in seconds that a replication transaction is holding at the Publisher and waiting for a Distributor.

**DTC Status**
Displays the status of the Distributed Transaction Coordinator service.

**Full-Text Search Status**
Displays the status of the Full-Text Search service.

**Last Full-Text Catalog Update (Hours)**
The number of hours beyond which the Full-Text Catalog statistics data is considered outdated.

**Non-Distributed Transactions (Count)**
The number of transactions not written to the Distribution database.

**SQL Server Agent Job Completion**
Provides the status of the Agent Job Completion metric. Values for this alert include Succeeded, Retry, Canceled, Failed, and Unknown.

**SQL Server Agent Job Failure**
Indicates that a scheduled job processed by the SQL Server Agent service ended abnormally.

**SQL Server Agent Long Running Job (Minutes)**
Provides the maximum number of minutes used to complete a scheduled SQL Server Agent job.

**SQL Server Agent Long Running Job (Percentage)**
Indicates the percentage of time a scheduled job is using over what it normally uses to complete the job.

**SQL Server Agent Status**
Displays the status of the SQL Server Agent service.

**SQL Server Status**
Displays the status of the SQL Server service.

**Unsubscribed Transactions (Count)**
The number of non-subscribed transactions written to the Distribution database.

**Unsubscribed Transactions (Seconds)**
The number of seconds a transaction written to the database waits before subscription.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

Get an overview of your services performance

The Services Summary tab includes the status of your Services. You can start and stop your Distributed Transaction Coordinator (DTC), or Full-Text Search by selecting the appropriate button.

SQL Server 2008 and above no longer have a separate full-text search service. If the monitored instance is SQL Server 2008 or above, the full text search does not show on the grid or graph.

The Service Availability chart displays the DTC, Full-Text Search, SQL Server, and SQL Server Agent status.

An Unavailable status in the chart is caused by a pending status change such as starting, stopping, pausing, and continuing.

Access the Services Summary view

You can open the Summary view of the SQLdm Services tab by selecting the appropriate SQL Server instance, and then clicking Services > Summary.
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.

Access the SQL Agent Jobs view

You can open the SQL Agent Jobs view of the SQLdm Services tab by selecting the appropriate SQL Server instance, and then clicking Services > SQL Agent Jobs.

Last Run Outcome colors

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.

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

Use the Full-Text Search view 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 populates 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.

Access the Full-Text Search view

You can open the Full-Text Search view of the SQLdm Services tab by selecting the appropriate SQL Server instance, and then clicking Services > Full-Text Search.

Optimize or Rebuild my Full-Text Catalog
Optimize

When you select the Optimize button, SQL Server optimizes the space utilization of the catalog and improves 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.

Analyse replication information

The Replication view allows you to analyze data replication details for the selected SQL Server instance. Included are the topology and key metrics that detail the current state of active subscribed and non-subscribed transactions on all SQL Server instances related to replication on the monitored SQL Server. For more information about replication, see the Microsoft document, SQL Server Replication.

Collecting replication statistics can have a performance impact on the monitored SQL Server instance.

Using the Replication view

The following elements in the Replication view help you make sure that proper replication is occurring and that your data is distributed to the correct subscribers.

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, we recommend 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 the participant 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.

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.

Access the Replication view

You can open the Replication view of the SQLdm Services tab by selecting the appropriate SQL Server instance, and then clicking Services > Replication.

Monitor logs

The Logs tab allows you view SQL Server related logs. Use these logs to diagnose SQL Server problems where a history of events is necessary to form context. The current log is selected by default but 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.

Access the Logs tab

SQLdm provides two paths to access the Logs tab. The first access path is by clicking the appropriate monitored SQL Server instance, and then clicking the Logs tab. The second access path is by expanding the Servers tree, and then clicking Logs for the appropriate instance.
Available actions

Actions

Click Cycle Server Log to archive the information in the most recent log and start a new log.

Click Configure to determine the number of SQL Server logs retained by SQL Server.

Details section

The Details section provides more information for the selected event.

Filter

The Filter options allow you to return specific data. Note that these selections work only once you enable log alerts.

Logs tab

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.

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, check the error log.

For Example: if 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.

Search

The Search feature allows you to find information based on text available in the message, message number, or source.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Filter SQL Server events

The Filter Settings window allows you to select:

- The time range to filter events, such as Date Range - Begin and Date Range - End
- 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.

Access the Filter Settings window

You can open the Filter Settings window from the SQLdm Logs tab by selecting the appropriate SQL Server instance, and then clicking Logs, and then clicking Filter.

Settings Server Log filters

To Filter the Server Log:

1. Click the Filter button on the Logs tab.
2. Verify that SQLdm displays the correct Source.
3. Make the necessary changes in the Message Severity area.
4. Type a Date Range - Begin and Date Range - End in the Period section of the Filter Settings window.
5. Click OK.

Search for SQL Server events

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Cycle the SQL Server log

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Configure 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.

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

View the Timeline for this instance

The Timeline view provides a chronological chart of events that affect your monitored SQL Server instance. The view details the point in time for this monitored SQL Server instance when an alert occurred, including custom counter alerts.

SQLdm displays as many alerts as can fit in the window and uses color codes based on the alert type as shown in the key below the menu. While a single Timeline point can represent more than one alert, the point itself appears in the color of the most critical alert for that time.

Use the filters to pinpoint the information you want to view. If the displayed time frame is not exactly what you want to view, select another period from the Filter drop-down list. Notice that the measurement at the bottom of the window changes from hours to days if you switch from a daily option to one over a period of days. Select Pick a Day to select a specific day using the calendar.

You can also use the time frame slider at the bottom of the view to zoom in or out by dragging the start and end bars to the desired position. Click the left or right arrows to move your set range to another period in the timeline. SQLdm retains the range size, but changes the results based on the dates within the range.

SQLdm displays alerts on the Timeline in your local time even if the monitored SQL Server instance where the event occurred is located in a different time zone.

Access the Timeline view

You can open the Timeline view of the SQLdm Overview tab by selecting the appropriate SQL Server instance, and then clicking Overview > Timeline.

Access more alert detail from the Timeline
View the total number of alerts at a specific time

When you hover your cursor over a point in the timeline, SQLdm displays the number of alerts for the time displayed in the timestamp.

Zoom in for more detail

Your alert configuration may cause the Timeline view to appear cluttered in the default settings. Occasionally, a timeline point appears without an attached alert flag to let you know about the associated alert. If you experience these issues, use the time frame slider at the bottom of the view to zoom in for a clearer picture of a set of alerts.

View alerts

While the Timeline provides your alerts in chronological format, you may want to view more detail about a single alert while using this view. To access the Alerts view from the Timeline, simply right-click the alert and select View Alerts. SQLdm displays the view you receive when you click the Alerts pane for a monitored SQL Server instance.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Alert on SQL Server Metrics

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 an email notification
- Pop up an alert message in your Windows taskbar
- Write an event to the Windows Event log
- Generate an event on the Timeline
- 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
- Create a new To Do item

In response to a set threshold entering the error level, SQLdm can perform one or more of the following actions:

- Send an email notification
- Write an event to the Windows Event log
- Generate an event on the Timeline
- 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
- Create a new To Do item

Once you correct the situation triggering the alert, SQLdm alerts you again if the situation recurs.

Database- or disk-level alerts

SQLdm includes some alerts that allow you to set them at a database or disk level, meaning that you can set different alert thresholds for each database or disk within a monitored SQL Server instance.

Alert templates

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 Templates and complete the required fields to configure an alert template.

Informational alerts

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.
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.

SQLdm Mobile

SQLdm Mobile includes the Idera Newsfeed as well as several dashboard views that define your server health and expose critical alerts. You can access the SQLdm Mobile Web application from most smart phones and mobile devices.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Configure alerts

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 then select Configure 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 either on the Configuration tab of the Alert Configuration window or, for database- and disk-level alerts, on the Database Threshold Configuration window available by selecting the database or disk, and then clicking Edit.

You can also base your alerts on the past performance of the metrics collected by SQLdm. This is a powerful and effective way to make sure that the alerts you receive are outside of your typical metric ranges.

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. If the metric features a per-database or per-disk alert, on the Configuration tab of the Alert Configuration window, click Add. Use the drop-down list to select the database or disk to which you want to apply these settings. If the metric applies at the instance level, continue with the next step.
5. Check the boxes next to Informational, Warning, and Critical to include alerts for these states.
6. Change the alert thresholds by moving the arrows to the appropriate levels or by double-clicking the value and typing a new threshold level.
7. If the metric features a per-database or per-disk alert, click Advanced to apply any advanced settings, such as alert suppression or autogrow settings. Click OK. If the metric applies at the instance level, continue with the next step.
8. Select the Comments tab and enter information you want displayed in the alert message for this metric.
9. Click Apply.
10. If you want to replicate these edits to other SQL Server instance SQLdm is monitoring, tags, or templates, click Yes.
11. Click OK to accept your changes.

Note that when you modify an alertable metric in the Alert Configuration window, the Create Template button is disabled. Once you apply the modification, this command button is enabled again.

Alert refresh

A full refresh of alerts excluding table fragmentation occurs in the following circumstances even if longer collection intervals are defined:

- when the collection service starts
- when the collection service receives a new workload, such as redirecting the Management Service to a new Repository
- when you add a SQL Server for collection
- when a SQL Server resumes from Maintenance Mode
- when a user selects Refresh Alerts

This refresh does not cause collection of non-alertable data, nor cause collection of anything that is disabled.

Table Fragmentation alert thresholds

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, you may experience 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.

Apply different alert thresholds to multiple databases or disks

Some of the available alerts allow you to apply them independently to multiple databases or disks on your monitored SQL Server instance. When you click the alert to modify the thresholds, notice that a list of your databases or disks appears with the associated thresholds as they are currently set. Click Edit to make your changes. The Database Threshold Configuration dialog lets you specify the database, and then select the thresholds for that database only. The Disk Threshold Configuration dialog provides the same functionality, only disks replace the databases.

Alert templates

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 Templates and complete the required fields to configure an alert template.

Informational alerts

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.

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
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.

**Access the Alert Templates window**

You can open the Alert Templates window from the SQLdm Console menu bar by selecting **Tools > Alert Configuration Templates**.

**Create a new alert template**

Create a new alert template or edit the existing Default Template to suit your needs when alerting on your monitored SQL Server instances.

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 to use this new template as 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.

**Change an existing template**

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**.

**Apply template changes to a tag or server**

Once you create a new template or make changes to an existing template, you should apply the updates to your tags or servers. Select the template from the list, click **Apply To**, select a tag or servers 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 tag or server, these changes are not automatically applied when you change the template. You must use the **Apply To** button and select the appropriate tag or servers.

**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 **Set Default**, and then click **OK**. If there is only one template in your list, SQLdm uses this template as the default.

**Apply alert templates to individual databases or disks within an instance**

SQLdm includes the ability to apply some alert thresholds on a per-database or per-disk basis within an instance. When you create an alert template and include one of these alerts, you cannot select which database or disk to apply the alert to because that is unknown in a template. For that reason, these alerts apply to all databases and disks within the instance to which it is applied.

**Configure alert template settings**

The Alert Template Configuration window allows you to set up the alerts templates for multiple or all SQL Server instances.

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:
### Alert Type | Color | Meaning
--- | --- | ---
OK | Green | Acceptable threshold where SQLdm does not generate an alert.
Informational | Blue | Informational threshold where SQLdm generates an informational alert.
Warning | Yellow | Warning threshold where SQLdm generates a warning alert.
Critical | Red | Critical threshold where SQLdm generates a critical alert.

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 either on the Configuration tab of the Alert Configuration window or, for database- and disk-level alerts, on the Database Threshold Configuration window available by selecting the database or disk, and then clicking Edit.

You can also base your alerts on the past performance of the metrics collected by SQLdm. This is a powerful and effective way to make sure that the alerts you receive are outside of your typical metric ranges.

For each alert metric, you can enter the specific criteria by double-clicking the values on the Configuration tab.

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.

**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 SQLdm to replicate your edits to every monitored SQL Server instance,** click Yes. Click OK to accept your changes.

**Use alert templates in individual databases or disks within an instance**

SQLdm includes the ability to apply some alert thresholds on a per-database or per-disk basis within an instance. When you create an alert template and include one of these alerts, you cannot select which database or disk to apply the alert to because that is unknown in a template. For that reason, these alerts apply to all databases and disks within the instance to which it is applied. For additional information, see Configure database thresholds or Configure disk thresholds.

**About informational alerts**

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 Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

**Configure database thresholds**

The Database Threshold Configuration dialog allows you to set alert thresholds for selected databases on your SQL Server instance. Not all alerts are available for database-specific application. The following alerts include database-specific settings:

- Database Full (Percent)
- Database Full (Size)
Configure disk thresholds

The Disk Threshold Configuration dialog allows you to set alert thresholds for selected disks on your SQL Server instance. Not all alerts are available for applying disk-specific thresholds. The following alerts include disk-specific settings:

- OS Disk Free Space (Size)
- OS Disk Full (Percent)

Use the Advanced button to set how and when alerts are raised by limiting the number of alerts and notifications generated.

To add a disk for specific alert configuration:

1. Right-click a SQL Server instance in the Servers tree.
2. Select Configure Alerts.
3. Select the disk-specific metric you want to edit from the list in the Alert Configuration window.
4. On the Configuration tab, click Add.
5. Select the disk from the available drop-down list.
6. Make the appropriate changes to your alert configuration for this disk, and then click OK. SQLdm now displays that disk in the list on the Configuration tab.
7. Click OK to accept your changes.

To edit the alert threshold configuration for a disk:

1. Right-click a SQL Server instance in the Servers tree.
2. Select Configure Alerts.
3. Select the disk-specific metric you want to edit from the list in the Alert Configuration window.
4. On the Configuration tab, click the disk you want to configure, and then click Edit.
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, and then click OK.
6. Click Apply.
7. If you want SQLdm to replicate your edits to every monitored SQL Server instance, click Yes, select the instances you want to include, and then click OK.
8. Click OK to accept your changes.

To delete the alert threshold configuration for a disk:

1. Right-click a SQL Server instance in the Servers tree.
2. Select Configure Alerts.
3. Select the metric from which you want to remove the specific database from the list in the Alert Configuration window.
4. On the Configuration tab, click the database you want to delete, and then click Delete.
5. Click Yes in the confirmation window.
6. Click Apply.
7. If you want to delete the same database from any other SQL Server instance SQLdm is monitoring, click Yes, select the instances you want to include, and then click OK.
8. Click OK to accept your changes.
threshold level, and then click OK.
6. Click Apply.
7. If you want SQLdm to replicate your edits to every monitored SQL Server instance, click Yes, select the instances you want to include, and then click OK.
8. Click OK to accept your changes.

To delete the alert threshold configuration for a disk:
1. Right-click a SQL Server instance in the Servers tree.
2. Select Configure Alerts.
3. Select the metric from which you want to remove the specific disk from the list in the Alert Configuration window.
4. On the Configuration tab, click the disk you want to delete, and then click Delete.
5. Click Yes in the confirmation window.
6. Click Apply.
7. If you want to delete the same disk from any other SQL Server instance SQLdm is monitoring, click Yes, select the instances you want to include, and then click OK.
8. Click OK to accept your changes.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

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.

**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
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)

To 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.

To 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 Edit Configuration.
4. If you clicked View/Edit, click Edit Configuration.
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 SQLdm to replicate your edits to every monitored SQL Server instance, 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.

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 prompts 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:
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 alerts 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 Free Space (Size)
- OS Disk Full (Percent)
- OS Disk Time Per Disk (Percent)

To 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. Select the appropriate disk, click Edit, and then click Advanced.
5. Click the Alert Filters tab.
6. Click Exclude.
7. Select one or more disk drives from the list, and then click Exclude.
8. Click OK.
9. To accept these changes, click OK.

To 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 Edit Configuration.
4. If you clicked View/Edit, click Edit Configuration.
5. Select the metric for which you want to configure an alert.
6. Select the appropriate disk, click Edit, and then click Advanced.
7. Click the Alert Filters tab.
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 SQLdm to replicate your edits to every monitored SQL Server instance, click Yes.
11. To accept these changes, click OK.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

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. You can also limit the size of your logs to decrease your response time and avoid using additional space on your server.

To 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.

To 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 Edit Configuration.
4. If you clicked View/Edit, click Edit Configuration.
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 SQLdm to replicate your edits to every monitored SQL Server instance, click Yes.
11. To accept these changes, click OK.

Limit the size of a log that SQLdm reads

In an attempt to improve response time and avoid using unnecessary space on your server, you can designate the maximum size of your SQL Server Agent log and your SQL Server Error log that SQLdm reads. When a log reaches the size limit set in the Alert Configuration, it triggers an alert. SQLdm does not display in the Logs view any data from error or agent logs in excess of the size limit. A quick solution to decrease the size of your file is to click Cycle Server Log to archive the information in the most recent log and start a new log. You can also cycle the log using SSMS.

Before cycling your log, it is important to note that if you have a limit set, then the log file deleted is the oldest log file stored. If you want to keep that log file, copy the file elsewhere before cycling the log.

SQL Server 2000 users also must restart the SQL Server Agent Service after cycling the SQL Server Agent log. The only way to cycle the agent log on SQL Server is to cycle the agent service. This process results in a new log.

To adjust the log size limit that SQLdm reads:

1. View the Alert Configuration for your monitored SQL Server instance.
2. Select either the SQL Server Agent Log or the SQL Server Error Log alert, and then click Advanced.
3. In the Alert Filters tab, adjust the Size (MB) field based on what you want for your log size limit.
4. Click OK, and then click Apply.
5. Click OK in the confirmation message if you want to apply your change to any other SQL Server instances, or click No.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
environment. Each time an auto-growth event occurs, SQL Server holds up database processing while looking for disk space for the additional data. Usually this new space is not adjacent to the existing space and causes physical disk fragmentation. As more and more auto-growth events occur, the more the disk becomes fragmented, leading to longer periods while SQL Server reads the databases.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Configure alert filters - database exclusions

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 alerts that allow for database exclusions include the following:

- Data File Autogrow
- Database Full (Percent)
- Database Full (Size)
- Database Status
- Log File Autogrow
- Log Full (Percent)
- Log Full (Size)
- Table Fragmentation (Percent)

Access the Alert Filters tab of the Alert Configuration window

1. Right-click the SQL Server instance you want to manage and then select Configure Alerts.
2. Select the alert you want to configure.
3. If the alert is database-specific, select the appropriate database, click Edit, and then click Advanced. If the alert is not database-specific, click Advanced.
4. Click the Alert Filters tab.

Exclude a database from an alert

You can select one or more databases to exclude from alerting on the selected metric.

1. Access the Alert Filters tab using the previous steps.
2. Click Exclude.
3. Select one or more databases from the list, and then click Exclude.
4. Click OK.
5. 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.

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.

Configure session exclusions

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 even a 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.

Configure custom counter suppression

The Advanced Alert Configuration window allows you to select whether you want an alert created when SQLdm cannot collect a custom counter. Preventing an alert when you cannot collect the custom counter eliminates any potential noise the alert could generate.

If a custom counter collection fails, it is included in a critical summary alert that lists all the custom counters that it could not collect.

Add or edit job filter

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.

Access the add or edit job filters

To access the Add Job Filter or Edit Job Filter window:
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 Advanced, and select Job Filters.
5. Click Add or Edit in either the Include or Exclude list.

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.

![Use the percent symbol (%) as a Wildcard.]

3. Click OK.

![The Browse button is only available when adding or editing job step filters for a single SQL Server instance.]

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Alert value configuration

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, you receive a Critical Level alert if the Database is ever offline. The alerts are available as shown in the following image.

<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>

Access the Value Configuration window

SQLdm displays the Value Configuration window when you attempt to edit values for certain metrics.

**To access the Value Configuration window:**

1. Right-click a monitored SQL Server instance, and then select Configure Alerts.
2. In the Alert configuration window, 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
   - Host Power State
   - VM Power State
3. Click Edit.

About informational alerts

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 Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
Base alerts on past performance

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 collection begins, SQLdm gives you recommendations for tweaking your alert thresholds for the following metrics:

- Blocked Sessions (Count)
- Client Computers (Count)
- Custom Counters
- OS Average Disk Queue Length (Count)
- OS Disk Time (Percent)
- OS Memory Usage (Percent)
- OS Paging (Per Second)
- OS Privileged Time (Percent)
- OS Processor Queue Length (Count)
- OS Processor Time (Percent)
- OS User Time (Percent)
- SQL Server CPU Usage (Percent)
- SQL Server Response Time (Milliseconds)

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. About 24 hours after collections begin, SQLdm displays a bar in green with black diagonal lines for related alerts on the Alert Configuration window, as shown in the following image. This bar represents you recommended reference range.

3. As your performance baseline period ends, text appears at the top of your Alert Configuration window regarding available recommendations. 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

View suggested alert thresholds

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Idera Website  Products  Purchase  Support  Community  About Us  Resources  Legal
Set recommended alert options

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 identifies and resolves SQL Server performance problems before they happen.

Configure alerts for important cluster events

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

The preferred node. 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.

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.

Organize alerts

The Alerts view allows you to view all alerts for all of the servers in your organization. You can organize the alerts in the list using either the Current View or the Filter.

Access the Alerts view

You can open the Alerts view from the SQLdm Console by clicking Alerts in the Navigation pane.

Organize your alerts by the Current View

You can organize your alerts by selecting one of the following views from the Current View section of the Navigation Pane:

- **Active**
  Organizes all your alerts that are currently active.

- **Severity**
  Organizes all your alerts by their severity.

- **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.

Organize your alerts using the Filter
SQLdm allows you to organize your alerts using the available filter options. Click the Show Filter Options/Hide Filter Options toggle in the Navigation Pane to open the Alerts filter. Filter options such as the specific view, SQL Server instance, metric, severity, and time range associated with the alert help pinpoint to data you want to view.

You can also filter alerts based on the assigned tags.

Configure how SQLdm responds to alerts
The Alert Actions and Responses window allows you add, modify, and configure alert responses. You can:

- **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.

When you first add your SQL Server instance to SQLdm, you should begin setting up your alert thresholds, and then set up your alert action rules and providers to specify who receives related notifications.

Access the Alert Actions and Responses window
You can open the Alert Actions and Responses window from the SQLdm Console menu by selecting Tools > Alert Actions and Responses.

Available Alert Responses tab actions
You can enable or disable an action provider by checking/clearing 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. The copy function can save time when you want to change just one setting on a particular rule.

Remove
Click Remove to delete the selected alert response.

Available Action Providers tab actions

You can enable or disable an action provider by checking/clearing the box in the Enabled column.

Add

Click Add to create a duplicate action to add multiple instances to a rule. For example, 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.

Remove

Click Remove to delete an SMTP or SNMP action provider.

---

Configure automated responses to alerts

Use the Alert Response window to create or modify your new alert response rule.

Access the Alert Response window

1. Select Tools > Alert Actions and Responses.
2. Click Add on the Alert Responses tab.

Add a new rule

After you create a new rule, you can then create a ToDo 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. This version of SQLdm 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.

Available conditions

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 contains a 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 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.

Available action providers

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 triggers the Query Monitor to start so that you can collect more information.

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.
5. Enter the full path of the directory in which you want the program to run.
6. Click Test to make sure 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.

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.

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, this version of SQLdm 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.

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Add action provider

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.

---

**Alert Communications wizard**

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.

*Access 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.

**Choose provider type**

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.

*Access the Choose Provider Type 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.

*Configure your email settings*

1. Click Alerts in the Navigation pane, and then click Alert Actions and Responses.
2. Click the Action Providers tab.
3. Click Add.
4. On the Welcome page of the Alert Communications wizard, click Next.
5. Select the Simple Mail Transfer Protocol (SMTP) provider type, type a unique name, and then click Next.
6. Type the SMTP address, port number, and the number of seconds to wait before a timeout occurs.
7. If the Server requires authentication, check Server requires authentication, and then type the appropriate login information for the email server.
8. Type the Name and E-mail address you want to appear in the From field in alert notifications.
9. To verify the connection to the SMTP server, click Test. Type the email address to send the test email, and then click OK.
10. Verify that the correct recipient received the test email message, and then click OK. If the test is unsuccessful, review your settings.
11. Click Finish.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Configure network management settings

The Alert Communications wizard allows you to set up Simple Network Management Protocol (SNMP) for sending out alert events to your Network Management system on the SNMP Configuration window.

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.

SQLdm supports SNMPv1 and SNMPv2

Access the Alert Communications wizard SNMP Configuration 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 Next.

Set up your SNMP communication

1. Enter the SNMP address, port number, and the community name that you want to use.
2. Click OK.

It is best to specify a computer name for the address, especially in a DHCP environment. If the manager has a static address, enter the IPv4 formatted address instead.

3. Click Test to verify the connection to the SNMP enabled management console.
4. Verify that the test event arrived and click OK if the test is successful.
5. Click Finish.

Understanding the Idera MIB

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.

SNMP traps enable an agent to spontaneously inform the management station of significant events by way of an unsolicited SNMP message. Traps are suitable when there is a large number of devices and each device has a large number of objects.

Note that in order for a management system to interpret a trap sent to it by an agent, it needs to have the MIB for that trap loaded.
The MIB file (Idera.MIB) defines the following data:

<table>
<thead>
<tr>
<th>Name</th>
<th>OID</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alert Summary</td>
<td>.1.3.6.1.4.1.24117.1.2.1</td>
<td>DisplayString</td>
<td>Summary text of the alert</td>
</tr>
<tr>
<td>Alert Text</td>
<td>.1.3.6.1.4.1.24117.1.2.2</td>
<td>DisplayString</td>
<td>Full text description of the alert</td>
</tr>
<tr>
<td>Metric</td>
<td>.1.3.6.1.4.1.24117.1.2.3</td>
<td>DisplayString</td>
<td>SQLdm metric that triggered the alert</td>
</tr>
<tr>
<td>Severity</td>
<td>.1.3.6.1.4.1.24117.1.2.4</td>
<td>DisplayString</td>
<td>Severity of the alert</td>
</tr>
<tr>
<td>Instance</td>
<td>.1.3.6.1.4.1.24117.1.2.5</td>
<td>DisplayString</td>
<td>The SQL server instance that triggered the alert</td>
</tr>
<tr>
<td>Database</td>
<td>.1.3.6.1.4.1.24117.1.2.6</td>
<td>DisplayString</td>
<td>If applicable to the alert, the database that triggered the alert</td>
</tr>
<tr>
<td>Table</td>
<td>.1.3.6.1.4.1.24117.1.2.7</td>
<td>DisplayString</td>
<td>If applicable to the alert, the table that triggered the alert</td>
</tr>
<tr>
<td>Timestamp</td>
<td>.1.3.6.1.4.1.24117.1.2.8</td>
<td>DisplayString</td>
<td>The date and time that the alert was detected</td>
</tr>
<tr>
<td>Value</td>
<td>.1.3.6.1.4.1.24117.1.2.9</td>
<td>DisplayString</td>
<td>The value detected for the specific metric</td>
</tr>
<tr>
<td>Description</td>
<td>.1.3.6.1.4.1.24117.1.2.10</td>
<td>DisplayString</td>
<td>Description of metric</td>
</tr>
<tr>
<td>Comments</td>
<td>.1.3.6.1.4.1.24117.1.2.11</td>
<td>DisplayString</td>
<td>User specified comments pertaining to the metric</td>
</tr>
</tbody>
</table>

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

**Report on SQL Server Performance**

The Reports view lets 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.

- **Access the Reports view**
  
  SQLdm provides two paths to access the Reports view. The first access path is by clicking Reports in the Navigation pane. The second access path is by clicking Go > Reports. The second path is the only option if you hide your Navigation pane in the SQLdm Console.

- **Customize 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.

  Click Show Filters 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.

- **Deploy reports to Microsoft Reporting Server**
  
  You can use the SQLdm Reports Deployment wizard to deploy specified reports to the Microsoft Reporting Server.

---

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
Report filters

Every report in SQLdm contains filters to allow you to select the information that you want included in your reports. If you want to return to the default filters after making some choices, click Reset Filters at the top of the report. Use the Show Filters/Hide Filters toggle depending on your preference.

The following list includes all of the filters available, though each report contains different options.

- **Case Insensitive**
  
  Check whether you want SQLdm to ignore query case for the report results.

- **Chart Type**
  
  Select the type of chart you want to convey your report results.

- **Compare Server**
  
  *Optional*. Select a SQL Server instance and period of time to compare statistics with your main selected Server and Period.

- **Compare with same server**
  
  Check whether you want to compare the same monitored SQL Server instance using a different metric or period.

- **Database, Database Name**
  
  Select or type the name of one or more databases on which you want to report.

- **Forecast Type, Forecast Units**
  
  Select the 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**
  
  Select the drive within the monitored SQL Server instance on which you want to report.

- **End Hour**
  
  Select or type the end time of the range from which you want to report trends. SQLdm includes trends through to the most recent collection time.

- **Include Batches/SQL Statements/Stored Procedures/System Databases**
  
  Check whether you want to include batches, SQL statements, stored procedures, or system databases in your report results.

- **Metric**
  
  Select the metric you want to include in your report.

- **Min. Duration**
  
  Select or type the minimum duration of the queries to display in your report.

- **Min. CPU/Executions/Reads/Writes**
  
  Select or type the minimum number or size of CPU, executions, reads, or writes for the queries you want to display in your report.

- **Min. Bytes Read/Bytes Written/Data Size/Growth/Transactions**
  
  Select or type the minimum number or size of bytes read or written per second, data size, growth, or transactions for the databases you want to display in your report.

- **Number of Applications**
  
  Select or type the number of database applications that you want included in your report.

- **Number of Databases/Servers**
  
  Select or type the number of databases or monitored SQL Server instances you want included in your report.

- **Number of Queries**
  
  Select or type the number of queries that you want returned in the report results.

- **Action**
  
  Select an action type on which you want to report.
Repository User
Select a repository user by whom you want to filter your report.

Workstation
Select a workstation on which you want to report.

Workstation User
Select a workstation user account used to access the workstation on which you want to report.

Period
Select the time range on which you want to report. Choose Custom Range to select the sample months, days, hours, or minutes on which you want to report. The sample choices depend on the type of report selected.

Range
Displays the period of time selected for your report.

Sample
Select the period of time between data points on the graphs that appear in your report. This selection also represents the most current information as of that particular point.

Search Item and Search Value
Select up to four different items and their values to search for in your server inventory, and then type the value you want to search to find. Use a percent (%) character as a wildcard when specifying a Search Value.

Server
Select the monitored SQL Server instance or instances on which you want to report.

Show Problems Only
Check whether you want to include only problems in your report results.

Signature Mode
Check whether you want to group similar queries in your report results.

Start Time, Start Hour
Select or type the start time for the range from which you want to report trends. SQLdm includes trends through to the most recent collection time.

Tag
Select a sample group for the report based on the tags that you assigned to those monitored SQL Server instances.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more >>]

---

### Monitor reports

Monitor reports give you overview information for your monitored SQL Server instances and the virtual machines on which the monitored SQL Server instances are running. In addition, oversee overall activity in your SQLdm environment through the Activity Monitor Reports.

**Access the Reports view**

SQLdm provides two paths to access the Reports view. The first access path is by clicking Reports in the Navigation pane. The second access path is by clicking Go > Reports. The second path is the only option if you hide your Navigation pane in the SQLdm Console.

**Server Monitor Reports**

- **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.

**Availability Group Topology**

Use the **Availability Group Topology** report to view the current topology of an AlwaysOn Availability Groups configuration.

**Virtualization Monitor Reports**

**Virtualization Summary**

Use the **Virtualization Summary** report to view a summary of your virtualized environment.

**Virtualization Statistics**

Use the **Virtualization Statistics** report to view an analysis of your virtualization performance trends for a virtualized machine.

**Activity Monitor Reports**

**Change Log Summary**

Use the **Change Log Summary** report to view a list of all actions and configuration changes performed in your SQLdm environment over a specified period of time.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more > >](#)

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

---

**Server monitor reports**

Server monitor reports provide overview information for your monitored SQL Server instances. Click the links below for more information.

**Server Monitor Reports**

**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.

**Availability Group Topology**

Use the **Availability Group Topology** report to view the current topology of an AlwaysOn Availability Groups configuration.
Enterprise Summary

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 to 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 SQLdm calculates 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

Server Summary

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 to run this report

You should run the Server Summary report to view the detailed performance and health information of a single SQL Server.

How SQLdm calculates 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

Active Alerts

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 to 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 SQLdm calculates metric values on this report

The values on the Active Alerts report correlate with the related SQL Server metric.
Mirroring Summary
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 to run this report
You should run the Mirroring Summary report to view a quick overview of the health of your mirrored databases.

How SQLdm calculates metric values on this report
The values on the Mirroring Summary report correlate with the related SQL Server metric.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Metric Thresholds
The Metric Thresholds report provides a list of metric thresholds for a monitored SQL Server instance. To successfully monitor OS metrics, enable WMI or OLE automation on the monitored instance.

In the Metric Thresholds report results, notice that the Threshold Instance column displays the name of the database or disk affected by the metric in that row.

When to 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.

How SQLdm calculates metric values on this report
The values on the Metric Thresholds report correlate with the related SQL Server metric.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Availability Group Topology
The Availability Group Topology report allows users to view the current topology of an AlwaysOn availability group's configuration. The report displays all of the availability groups on the server, replicas that participate in the group, and the databases within the group.

You can filter results by availability group only. The report lists detailed information for availability replicas and corresponding databases.
Availability replica table:
Provides information on availability replica roles assigned, failover and availability modes selected, and the connection mode type selected in secondaries.
Availability databases table:
Provides information on the database failover readiness and synchronization state of the availability databases contained within the selected availability replica.

When to run this report
You should run the Availability Group Topology report to keep track of modifications made to your current AlwaysOn availability groups and avoid any potential setbacks to your high-availability and disaster recovery strategies.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Virtualization monitor reports
Virtualization monitor reports provide overview information for the virtual machines on which your monitored SQL Server instances are running. Reports are compatible for Hyper-V or VMWare VMs. Click the links below for more information.

Virtualization Monitor Reports

Virtualization Summary
Use the Virtualization Summary report to view a summary of your virtualized environment.
Virtualization Statistics

Use the Virtualization Statistics report to view an analysis of your virtualization performance trends for a virtualized machine.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

Virtualization Summary

The Virtualization Summary report provides an overview of the health and performance of the virtualized environment for the selected SQL Server instance. This report includes information about the VM as well as configuration of the host server. Click the appropriate instance and SQLdm displays the Virtualization Summary report.

When to run this report

You can run the Virtualization Summary report to view a quick overview of the health and performance of the virtualized environment for your selected monitored instance. Run this report when you want a quick glance at the VM information for this instance.

How SQLdm calculates metric values on this report

Most values on the Virtualization 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 does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055

Activity monitor reports

Activity monitor reports provide overview information of all actions and configuration changes performed in SQLdm. Click the link below for more information.

Activity Monitor Reports

Change Log Summary

Use the Change Log Summary report to view a summary of key actions performed in your SQLdm environment.
Change Log Summary

The Change Log Summary report lets you view a list of all actions and configuration changes performed in your SQLdm environment over a specified period of time and through various filter options. The report contains detailed information, such as repository user, action type and time of action among other key data. For example, users are able to view all changes performed by a specific repository user.

**When to run this report**

You should run the Change Log Summary report to audit overall activity in your SQLdm environment. This report allows you to analyze key changes that might hinder your performance or represent a risk.

Analysis reports

Analysis reports provide information that allows you to analyze the current effectiveness of your SQL Servers. Click the links below for more information.

**Access the Reports view**

SQLdm provides two paths to access the Reports view. The first access path is by clicking Reports in the Navigation pane. The second access path is by clicking Go > Reports. The second path is the only option if you hide your Navigation pane in the SQLdm Console.

**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.

- **Baseline Statistics**
  Use the Baseline Statistics report to review and compare metric and baseline values across two SQL Server instances.

**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 Application report to find database applications that consume system resources.
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.

**Tempdb Statistics**

Use the Tempdb Statistics report to analyze performance statistics for your tempdb database.

**Transaction Log Statistics**

Use the Transaction Log Statistics report to analyze the use of your log files.

**Availability Group Statistics**

Use the Availability Group Statistics report to view the historical health of your availability groups, availability replicas, and availability databases.

**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.

*SQL Diagnostic Manager* identifies and resolves SQL Server performance problems before they happen. Learn more > >

**Server analysis reports**

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.

**Baseline Statistics**

Use the Baseline Statistics report to review and compare metric and baseline values across two SQL Server instances.

**Query Wait Statistics**

Use the Query Wait Statistics report to analyze the different query waits in a SQL Server.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Top Servers

The Top Servers report identifies your worst performing SQL Server instances based on the number of active alerts for the current monitored instances, response time, CPU usage, memory usage, and disk usage.

You can further filter your monitored instances by wait time with waits that exceed a defined **Wait Threshold** in milliseconds.

Use this report to compile a list of instances that need immediate attention.

When to run this report

Run the Top Servers report daily to keep ahead of critical issues that can cause an unavailable instance.

How SQLdm calculates metric values on this report

Most values on the Top Servers 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](https://idera.secure.force.com/) 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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Server Statistics

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 to 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 SQLdm calculates metric values on this report

Most values on the Server 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](https://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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Server Inventory

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 and whether you have a clustered instance.

You can use % as a wildcard in the Search Value field.

When to run this report

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 are deployed.

Deleted SQL Server instances

When you choose to delete a SQL Server instance, SQLdm displays the confirmation message, "Would you like to retain the data collected for the selected instance?" When the user clicks Yes, SQLdm retains data for the instance and includes it in some reports, such as the Server Inventory report.

How SQLdm calculates metric values on this report

Most values on the Server Inventory report correlate with the related SQL Server 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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Query Overview

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 to 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 SQLdm calculates metric values on this report

The values on the Query Overview report correlate with the related SQL Server 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

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Top Queries

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 viewing results, note that you can click the SQL statement to view the Query Statistics report, which provides details on an individual SQL statement or signature as collected by the Query Monitor.
When to 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 SQLdm calculates 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

Query Statistics

The Query Statistics report lets you view details for an individual SQL statement or signature as collected by the Query Monitor. This report requires that you enabled Query Monitoring in the SQLdm Console and have collected sufficient query data for the specified time period.

When to run this report

You should run the Query Statistics report to analyze an individual SQL statement or signature for a monitored instance.

How SQLdm calculates metric values on this report

Most values on the Query 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 does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #00000055
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #00002237

Alert History

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 to 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 SQLdm calculates metric values on this report

The values on the Alert History report correlate with the related SQL Server 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

Baseline Statistics
The Baseline Statistics report lets you analyze and compare baselines within a single SQL Server instance and across two instances. When viewing baseline statistics for a monitored SQL Server instance, you can compare the baseline metric values at two different times or two different metrics at the same time. Include another instance and you can compare baselines values occurring at the same time or different times.

**When to run this report**

You should run the Baseline Statistics report when you want to view trends in the average value of a metric for a SQL Server instance and how this value changes over time to help you in capacity planning. This report also provides insight into the utilization of your different instances by comparing the baselines over time.

**Report selections change based on the selected Period**

The Baseline Statistics report Period indicates the date range for which you want to see results. If you select **Today** or **Custom Range**, SQLdm displays additional fields that allow you to filter results by the start and end hours.

- **If you select Custom Range as the Period, and the start and end dates are both for the same day, SQLdm calculates the selection the same as if you select Today as the Period.**

**How SQLdm calculates metric values on this report**

The baseline values are the most recent calculations for the selected period. Note that the Metric Value is an average of all data for the selected metric. For example, if you select **Hours** from the drop-down list, the Baselines Statistics report grid includes 24 rows for a single day, one row for each hour from 00:00 to 23:00. Each row displays a metric value which is the average of the selected metric for that hour, such as 05:00 to 05:59:59, and the baseline values which are calculated for the entire 24-hour day. As a result, SQLdm displays 24 rows of different metric values but equal baseline values.

You can access our [Customer Service Portal](https://idera.secure.force.com/) for more information about the algorithm used to calculate a specific metric.

- How does SQL diagnostic manager gather statistics from monitored SQL Servers? - Solution #0000055

---

**Query Wait Statistics**

The Query Wait Statistics report lets you analyze different wait type categories on your SQL Server instance. By analyzing these waits you can better determine where your biggest bottlenecks are occurring and what changes could have the greatest performance impact. You can select from various wait type categories such as Backup, Excluded, I/O, Lock, Memory, Non-I/O Page Latch, Non-Page Latch, Transaction Log, and Other.

This report requires that you enabled Query Monitoring in the SQLdm Console and have collected sufficient query data for the specified time period.

**When to run this report**

You should run the Query Wait Statistics report on a routine basis, such as every week, to proactively identify potential performance issues that may be caused by queries.

**How SQLdm calculates metric values on this report**

Most values on the Query Wait 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](https://idera.secure.force.com/) 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 #0000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #0000301
- What does the 'Memory Usage' metric in the Console and Reports represent? - Solution #00002237

---

**Database analysis reports**

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.

**Tempdb Statistics**
Use the Tempdb Statistics report to analyze performance statistics for your tempdb database.

**Transaction Log Statistics**
Use the Transaction Log Statistics report to analyze the use of your log files.

**Availability Group Statistics**
Use the Availability Group Statistics report to view the historical health of availability groups, availability replicas, and availability databases.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

---

**Top Databases**

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.

You can further filter your monitored databases by wait time with waits that exceed a defined Wait Threshold in milliseconds. Use this report to compile a list of databases that used most often or have the heaviest loads.

You can use % as a wildcard in the Search Value field.

**When to 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 SQLdm calculates metric values on this report**

Most values on the Top Databases 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 (https://idera.secure.force.com/) 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 Statistics

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 to run this report

You should run the Database Statistics report when attempting to understand performance differences or troubleshooting database issues.

How SQLdm calculates metric values on this report

Most values on the Database 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

Top Database Applications

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.

When to 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 SQLdm calculates metric values on this report

Most values on the Top Database Applications 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

Mirroring History

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 to 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.

**Top Tables by Growth**

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 to 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 a list of tables that are the top contributors and decide whether you want to move them to another database.

**How SQLdm calculates 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 #0000301
- What does the ‘Memory Usage’ metric in the Console and Reports represent? - Solution #00002237

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >
When to run this report

You should run the Tempdb Statistics report to view different space utilization and data throughput statistics currently occurring on your monitored SQL Server instances. This report allows you to obtain statistics through different metrics or chart types. Below a list of all possible metrics:

- Data File Size MB
- Data File Growth
- Reads Per Second
- Writes Per Second
- Transactions Per Second
- User Objects MB
- Internal Objects MB
- Version Store MB
- Mixed Extents MB

How SQLdm calculates metric values on this report

Most values on the Tempdb Statistics report are averages of data from the system DMV `sys.dm_db_file_space_usage` on the monitored SQL Server.

You can access our Customer Service Portal (www.idera.com/support) for more information about the algorithm used to calculate a specific metric.

How SQL Diagnostic Manager gathers statistics from monitored SQL Servers

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen.

Transaction Log Statistics

The Transaction Log Statistics report provides data about the database transaction log size and activity on a per-database basis. You can display your results in hours or days, depending on the sample range selected. Note that SQLdm displays the Cache Reads and Flushes chart types as data per second.

When to run this report

You should run the Transaction Log Statistics report when attempting to troubleshoot performance problems related to transaction logs and to discover what triggered the Long Running Transaction alert, Log Full (Percent) or Log Full (Size) alert, or Log File Autogrow alert.

How SQLdm calculates metric values on this report

Most values on the Transaction Log 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.

Availability Group Statistics

The Availability Group Statistics report allows users to view the health of their availability groups, availability replicas, and availability databases over a specific time range.

You can filter results by tag, availability group, period, sample, server, and chart type. The report compiles crucial health metrics that include the Redo Rate (KB/sec), Redo Queue Size (KB), Log Send Rate (KB/sec), and Log Send Queue Size (KB).

The Availability Group Statistics report also provides DBAs with a graphical representation of a selected metric and a statistics table with weighted averages for relevant metrics.

For individual details on availability databases per replica, click on the expandable command button.
When to run this report

You should run the Availability Group Statistics report when you want to analyze the historical health of your availability groups and corresponding components. This report helps you to ensure the highest level of availability by identifying and tracking critical health metrics.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Resource analysis reports

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.

Session Statistics

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. Check Show Tabular Data to display results in a single tabular or grid format.

Check Show Baseline to include baseline data points in the Session Statistics report. For additional information about baseline configuration, see Configure server baseline options.

When to 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 SQLdm calculates 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
**CPU Statistics**

The CPU Statistics report lets you view trends for CPU usage on a monitored SQL Server instance over a specified period of time. Check Show Tabular Data to display results in a single tabular or grid format.

Check Show Baseline to include baseline data points in the CPU Statistics report. For additional information about baseline configuration, see Configure server baseline options.

**When to run this report**

You should run the CPU Statistics report to analyze contributors to CPU usage for a monitored instance.

**How SQLdm calculates 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 #0000055
- How does SQL diagnostic manager calculate the reorganization percentage for database tables? - Solution #0000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #0000237

**Disk Details**

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, make sure that WMI/OLE automation is enabled on the monitored instance. Check Show Tabular Data to display results in a single tabular or grid format.

**When to 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 SQLdm calculates 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 #0000055
- How to customize existing reports used with Reporting Services? - Solution #0000307

**Disk Statistics**

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, make sure that you enable WMI or OLE automation on the monitored instance. Check Show Tabular Data to display results in a single tabular or grid format.

Check Show Baseline to include baseline data points in the Disk Statistics report. For additional information about baseline configuration, see Configure server baseline options.
When to 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 SQLdm calculates 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 #0000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #0002237

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Replication Statistics

The Replication Statistics report lets you view trends for the replicated transactions on a monitored SQL Server instance over a specified period of time. Check Show Tabular Data to display results in a single tabular or grid format.

Check Show Baseline to include baseline data points in the Replication Statistics report. For additional information about baseline configuration, see Configure server baseline options.

When to run this report

You should run the Replication Statistics report to analyze the efficiency of replication between databases.

How SQLdm calculates 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 #0000301
- What does the 'Memory Usage' metric in the console and reports represent? - Solution #0002237

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

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, make sure that you enable WMI or OLE automation on the monitored instance. Check Show Tabular Data to display results in a single tabular or grid format.

Check Show Baseline to include baseline data points in the Memory Statistics report. For additional information about baseline configuration, see Configure server baseline options.

When to 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 SQLdm calculates 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
Plan reports

Planning reports allow you to plan for future needs based on current trends.

Access the Reports view

SQLdm provides two paths to access the Reports view. The first access path is by clicking Reports in the Navigation pane. The second access path is by clicking Go > Reports. The second path is the only option if you hide your Navigation pane in the SQLdm Console.

Available reports

- **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.

**Disk Space Usage Forecast**

The Disk Space Usage Forecast report lets you predict how much disk space you need 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.

*The Disk Space Usage Forecast report does not display the maximum capacity of the disk. You can use this report to estimate how much space you need, but it does not indicate when existing space runs out.*

*When to run this report*

Run the Disk Space Usage Forecast report to help anticipate your needs and plan for future acquisition of disk space.

**Available forecast types**

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 gives a more accurate forecast of future growth.

**How SQLdm calculates 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.
SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >

**Database Growth Forecast**

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 to run this report*

You should run the Database Growth Forecast report to help anticipate your needs and plan for future database expansion.

*Available forecast types*

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 gives a more accurate forecast of future growth.

*How SQLdm calculates 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

---

**Table Growth Forecast**

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 to run this report*

You should run the Table Growth Forecast report to help anticipate your needs and plan for future creation of additional tables.

*Available forecast types*

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 gives a more accurate forecast of future growth.

*How SQLdm calculates 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

Deploy SQLdm reports to Reporting Services

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.

Access the Reports Deployment wizard

SQLdm provides two paths to access the Reports Deployment wizard where you can select reports to deploy to a Microsoft Reporting Services Server. The first path is by clicking 'Deploy them now' at the bottom of the Reports > Getting Started view. The second path is by first selecting a specific report in the Reports > Getting Started view, and then clicking the Deploy Report option from the Navigation Pane.

In order to use this feature, you must have access to a server with Microsoft Reporting Services. For more information about Reporting Services, see the Microsoft document, Reporting Services Reports.

In the case of interlinked reports such as the Availability Group Statistics and Topology reports, users need to make sure to first deploy both reports from SQLdm or Microsoft Reporting Services displays an error message.

Select reports to deploy

SQLdm includes built-in reports that are specially designed to generate commonly-requested metrics reports using the SQL Server data collected in the repository.

Select reports

In the Reports Deployment wizard 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:

<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>
<tr>
<td>Metric Thresholds</td>
<td>View all metric thresholds for a server.</td>
</tr>
<tr>
<td>Availability Group Topology</td>
<td>View the current topology of your availability groups configuration.</td>
</tr>
</tbody>
</table>
### Virtualization Monitoring Reports

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtualization Summary</td>
<td>View a summary of your virtualized environment.</td>
</tr>
<tr>
<td>Virtualization Statistics</td>
<td>Analyze the performance trends of your virtual machines.</td>
</tr>
</tbody>
</table>

### Activity Monitoring Reports

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Log Summary</td>
<td>View a summary of all key actions performed in your SQLdm configuration.</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>
<tr>
<td>Baseline Statistics</td>
<td>Analyze and compare performance baselines across two SQL Server instances.</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>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>
<tr>
<td>Tempdb Statistics</td>
<td>Analyze the performance trends of your tempdb database.</td>
</tr>
<tr>
<td>Transaction Log Statistics</td>
<td>Analyze the database transaction log.</td>
</tr>
<tr>
<td>Availability Group Statistics</td>
<td>View the historical health of your availability groups, availability replicas, and availability databases. Plus view additional information through the Availability Group Database Statistics sub-report.</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>Disk Details</td>
<td>Track key disk metrics for server.</td>
</tr>
<tr>
<td>Disk Statistics</td>
<td>Track key disk performance metrics.</td>
</tr>
</tbody>
</table>
Specify a reporting services server and data source

In order to successfully deploy the reports, you must specify a Microsoft Reporting Services Server and a data source for the report.

Type the name of the report server or select it from the drop-down list, and then click Next.

Configure report email notifications

Enter the email address where you would like SQLdm to send the report notification, and then click Next. For more information on configuring report servers for email delivery, see the Microsoft document, How to: Configure a Report Server for E-mail Delivery (Reporting Services Configuration).

View the report deployment summary

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.

Custom reports

The Custom Reports view allows you to create reports on any counter collected by SQLdm and on any of your custom counters. You can create custom reports to view graphs of the counters followed by a grid containing all of the counters graphed over the entered time frame.

Access Custom Reports

SQLdm provides two paths to access the Reports view where you can access your custom reports. The first access path is by clicking Reports in the Navigation pane. The second access path is by clicking Go > Reports. The second path is the only option if you hide your Navigation pane in the SQLdm Console.

Once in the Reports view, click Custom in the Navigation pane.

Filter your 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.

Deploy reports to Microsoft Reporting Server

You can use the SQLdm Reports Deployment wizard to deploy specified reports to your Microsoft Reporting Server.

Edit or Delete an existing report
You can edit or delete an existing custom report by selecting the report name from the list displayed in the Reports pane, and then click Edit or Delete. To edit your report, use the Custom Report wizard.

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 edit an existing report.

To create a new report, type a name for the report, and then click Next.

To edit an existing report, select the custom report from the displayed list, and then click Next.

Select counter type and counters

The Select Counters page allows you to choose the counters you want to include in your report. You can select up to 10 counters from all of the counters that SQL diagnostic manager currently collects.
To select the counters you want to include in your 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. You can select counters from the following groups:
   - Operating System Counters.
   - SQL Server Counters.
   - Virtualization Counters for both Hyper-V and VMware server virtualization platforms.
   - Custom Counters.

2. If you want SQLdm to display tabular data, check the Show tabular data in report box.
3. Click Next.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Select counter order

Update the counter order for this report to specify where you want SQLdm to display the counters included in your custom report and how to aggregate metrics.

SQLdm offers the following options to change the aggregation for each metric included in your report.

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.

**To rearrange or change the metric aggregation for your counters:**

1. **If you want to rearrange a counter in your report,** click the up or down arrow next to the counter you want to move.
2. **If you want to change the aggregation for a metric in your report,** select the appropriate Aggregation method.
3. Click Next.
4. Complete the Custom Report wizard by clicking **Finish**.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. [Learn more](#) > >

### Manage SQLdm

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. [Learn more](#) > >

### 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.

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.

#### 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 always have Administrator powers.

#### Available actions 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**

  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.
You can toggle non-system, SQLdm login permissions by checking and clearing 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.

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more > >](#)

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

---

**Add users to SQLdm**

The Add Permission wizard allows you to 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.

*Logins belonging to the sysadmin role on the SQLdm Repository have SQLdm Administrator privileges and you cannot edit or delete these accounts.*

Each login account can have multiple permissions associated with it. For example, if you want User A to have the ability to view all your SQL Server instances but modify only one of them, you can create two permissions. Include View permissions in the first login to allow the login to view all your SQL Server instances, and enable Modify permissions for the specified SQL Server instance in the second login.

*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 tags or 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.

---

**Specify user**

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 already exist, one is created on the SQLdm Repository Server.

*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.
<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>Configure default alerts</td>
</tr>
<tr>
<td></td>
<td>Edit notification rules or providers</td>
</tr>
<tr>
<td></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></td>
<td>Enter or exit Maintenance Mode</td>
</tr>
<tr>
<td></td>
<td>Configure alerts, but can view alert settings</td>
</tr>
<tr>
<td></td>
<td>Edit SQL Server properties, but can view properties</td>
</tr>
<tr>
<td>Today Page &gt; Active Alerts</td>
<td>Configure alerts, but can view alerts</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></td>
<td>Edit properties, but can view properties</td>
</tr>
<tr>
<td>Server Overview &gt; Dashboard</td>
<td>Enter or exit Maintenance Mode</td>
</tr>
<tr>
<td></td>
<td>Configure alerts, but can view alert settings</td>
</tr>
<tr>
<td>Server Overview &gt; Details</td>
<td>Configure alerts, but can view alert settings</td>
</tr>
<tr>
<td>Server Overview &gt; Configuration</td>
<td>Edit value</td>
</tr>
<tr>
<td>Session &gt; Details</td>
<td>Trace or kill a session</td>
</tr>
<tr>
<td>Session &gt; Locks</td>
<td>Trace or kill a session</td>
</tr>
<tr>
<td>Session &gt; Blocking</td>
<td>Trace or kill a session</td>
</tr>
<tr>
<td>Queries/Non-query activities</td>
<td>Configure Query/Activity Monitor properties, but can view properties</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>Configure default alerts</td>
</tr>
<tr>
<td></td>
<td>Edit notification rules and providers</td>
</tr>
<tr>
<td>Alerts &gt; Grid</td>
<td>Configure alerts, but can view alerts</td>
</tr>
<tr>
<td></td>
<td>Clear alerts</td>
</tr>
<tr>
<td></td>
<td>Clear alerts of this type for this instance</td>
</tr>
<tr>
<td>Administration pane</td>
<td>Access or view Administration</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.
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.

You cannot edit or delete logins belonging to the sysadmin role on the SQLdm Repository as they have SQLdm Administrator privileges.

Assign instances to user

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.

View the Add Permission summary

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.

Edit user permissions

The Edit Permissions window allows you to change the permissions for the specified login.

Logins belonging to the sysadmin role on the SQLdm Repository have SQLdm Administrator privileges and you cannot edit or delete the login.

To 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

Edit assigned instances

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >

<table>
<thead>
<tr>
<th>Idera Website</th>
<th>Products</th>
<th>Purchase</th>
<th>Support</th>
<th>Community</th>
<th>About Us</th>
<th>Resources</th>
<th>Legal</th>
</tr>
</thead>
</table>

**Use the Change Log to review changes in your SQLdm configuration**

Granted the number and different types of users in SQL Server production environments, keeping track of changes to your SQLdm configuration is critical. Through the Change Log you can view a list of key actions performed in your SQLdm environment, such as when a server is added for monitoring, an alert configuration or grooming configuration is changed. The Change Log view gives thorough and instant information on modifications.

The Change Log view is available by default in SQLdm version 8.0.

**Access the Change Log view**

To open the Change Log view, click **Administration > Change Log** from the navigation pane.

**Change Log Window Properties**

This window allows you to view details about an individual event in the Change Log. You can view the following information:

- Date and time of action
- Type of action
- Object on which the action occurred
- Workstation and workstation user
- Repository user who executed the action
- Description of action

**Available actions**

**Scroll through activities**

To scroll from one action to the next, use the up and down arrows.

**Copy, export and print action details**

To copy action details to another application, export contents to Microsoft Excel or even print, right-click on selected row. This action prompts the aforementioned commands.

**Getting more information**

You might be interested in obtaining further details about the actions performed in SQLdm and their possible impacts. The details pane contains in depth action information and automatically displays when a row in the table is highlighted. The split bar is adjustable to your needs.

**Filters**

Allows you to filter the listed activities by time span, for example, last 30 days or even arrange a custom range.

**Enable Groups**

Allows you to group actions by a specific header column, such as the workstation from which an action was executed or the date an action occurred. To enable groups, click **Group By Box**. You can as well expand all groups or collapse all groups.

**Available columns**
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.

About 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. You can use a console application, Windows application, or a Web application as the hosting 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.

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 is registered:

`get-psprovider`

To receive general help:

`get-help`

To display a list of cmdlets for the SQLdm snapin:
Applications 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.

PowerShell Cmdlets for SQLdm

The following cmdlets help you automate 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>Allows you to specify server names containing special characters, such as</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 that you want 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.

Use the trial license

SQLdm includes a limited-time trial account with every installation. Note that Idera provides separate trial and production installers. As the end of the trial period nears, SQLdm warns you each time you start the console that your trial period is ending. At the end of this trial period, if the license is not upgraded, SQLdm stops functioning.

SQLdm limits the number of monitored SQL Server instances to 15 when using a trial license.

Upgrade your trial license to a full 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.

Once you receive a new license, open the License Keys window by clicking Help > Manage Licenses. Type your license key in the available space, and then click Enter. Your new license lists the number of additional SQL Server instances that you can monitor by SQLdm. Each new license you add to SQLdm adds to the total number of SQL Server instances that you can monitor.

To get a new license, contact your Idera Sales representative.

Manage your SQL Server instances

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
- Access the VM Configuration window to associate your monitored virtual SQL Server instances to a host machine
- Apply an existing alert template to a server or group of servers
- Test the connection status of your SQL Server instances

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more »>

### Edit your Management Service configuration

The SQLdm Management Service Configuration wizard allows you to change how the SQLdm Management Service interacts with the SQLdm Repository.

You can edit the SQLdm Management Service settings whenever you change the location of your SQLdm Repository or if you want to change the account or authentication method the SQLdm Management Service uses to connect to the SQLdm Repository.

**Access the Management Service Configuration wizard**

You can access the SQLdm Management Service Configuration wizard 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 that you want the SQLdm Management Service to use to write configuration and collected statistics to the SQLdm Repository.
5. Click Test to verify the connection.
6. Click Next.
7. Verify the changes are correct.
8. Click Finish to apply the changes.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more »>

### Select a SQL Server instance for the Management Service

The SQLdm Management Service Configuration wizard allows you to select the SQL Server instance for the SQLdm Management Service to connect with the SQLdm Repository. Select the appropriate SQL Server instance from the lists of available local and network instances, and then click OK.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more »>

### Select a database for the Management Service

The SQLdm Management Service Configuration wizard allows you to select the database for the SQLdm Management Service to connect with the SQLdm Repository. Select the appropriate database from the list of available databases, and then click OK.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more »>

### Monitored metrics

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 a number of metrics for your use, such as:

- Batches Received (Per Second)
- Blocked Sessions (Count)
Batches Received (Per Second)

The Batches Received (Per Second) metric counts the total number of batch requests that the SQL Server computer receives 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 must 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.

**Reduce the number and impact of T-SQL batches**

In this situation, you should reduce the number of T-SQL batches and the impact of each 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 reduces the size and complexity of the T-SQL batch, which means less network traffic and less CPU time consumed producing a complex execution plan.

• Once you move most of the T-SQL code into stored procedures, consider grouping as many of those calling T-SQL commands into a single network batch, if 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 reduces network queues dramatically as better use is made of the default 4 KB network packet size.

**Blocked Sessions (Count)**

The Blocked Sessions (Count) metric counts the number of sessions on the SQL Server instance blocked by other sessions holding requested locks.

By default, a blocked session waits forever for a lock. You can change this behavior by adding `SET LOCKTIMEOUT 60000` to the top of the first T-SQL batch after you make a connection.

**Buffer Cache Hit Ratio (Percent)**

The Buffer Cache Hit Ratio (Percent) metric calculates the percentage of data pages SQL Server found in memory without reading the data 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.

**Reduce the percentage of memory used by SQL Server**

Once SQL Server is up for at least 30 minutes, use the following solutions if values lower than 85% regularly occur:

- Add more physical memory (RAM) to the computer.
- If your site makes use of extended stored procedures that are infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- Allowing SQL Server to consume more of the available memory, making sure that OS Paging does not increase.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.
- If the computer is running multiple instances of SQL Server, then consider placing each instance on a separate physical computer.

**Client Computers**

The Client Computers 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.
SQL Compilations (Per Second)

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 recompilation overhead than necessary. Too many compilations can impact a server’s performance.

---

Data Used (Percent)

The Data Used (Percent) 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 slowly increases over time in a linear fashion with the population of any available history tables.

Reduce the percentage of data used

If your site experiences a sudden sharp increase in this value of over 10% in an hour, investigate the issue to fully understand the underlying causes which often lead to poor performance. Possible causes include:

- A large number of rows were inserted into a table. If so, then make sure that the table is not fragmented due to this operation. To verify, run `DBCC SHOWCONTIG`.
- An index with very poor cardinality (uniqueness) was added to a large table. Indexes with poor cardinality often exceed the size of the underlying table itself and in almost all cases are not used by any SQL Server query.
- A large number of rows were incorrectly changed by performing an insert of a new row followed by a delete of the old row. Although the net effect of the change is essentially the same number of rows; the database would have grown significantly in size and require you to issue `DBCC SHRINKDATABASE`. 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 the tables are fragmented more quickly, databases increase in size, and the performance of the transactions is degraded due to the large number of page splits the table had to undergo.

---

Database Status

The Database Status metric indicates the operational status of the database, such as Normal, Offline, or Suspect. You can select the database status for which you want to receive alerts.

---

Disk Queue Length

The Disk Queue Length 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.

Reasons why this metric has no value

SQLdm is unable to collect the corresponding counter due to one of the following reasons:

- It usually takes two refreshes to begin populating the Disk Queue metric, which occurs within a few seconds.
OS Metrics collection is disabled or not working properly. You must enable OS Metrics collection for SQLdm to collect hardware performance metrics.

Lightweight Pooling is enabled. SQLdm cannot collect OS metrics if Lightweight Pooling is enabled on the SQL Server instance.

The counter does not exist on this SQL Server instance.

Disk Time (Percent)

The Disk Time (Percent) 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_PercentIdleTime is measured to get Disk Idle Time percent.

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

The I/O Waits 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.

Lock Waits

The Lock Waits 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)

The Log Flushes (Per Second) metric reads LOG_FLUSHES/SEC from sysperfinfo. Each time data changes, the changes write to the log cache, which resides in memory. Under certain conditions, this log cache is flushed to disk. These situations include:

- A commit or roll back of an explicit or implicit transaction.
- A CHECKPOINT statement execution.
- SQL Server issues an automatic checkpoint.
- The Log Cache is full and needs emptying.
Reduce log flushes

On an OLTP system that runs relatively small transactions (on average, less than 40 rows 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 you exceed this percentage, then one of the following abnormal conditions exists:

- T-SQL batches sent for execution contain a number of individual data manipulation statements (inserts, deletes, or updates) not enclosed within a transaction.
- Large batch jobs which update significant amounts of data are run during the day.
- Deadlocks or Lock Timeouts are causing a number of transactions to roll back or execute a ROLLBACK TRANSACTION statement.
- Your Recovery Interval parameter is too low. Raise this parameter to 10-15 minutes.
- An explicit CHECKPOINT statement is executed within some transactions.
- If you need to delete the entire contents of a table, use the TRUNCATE TABLE statement instead of DELETE as it produces less log flushes.

If you implement the previously-mentioned measures and the Log Flushes still greatly exceed 140% of SQL Batches, then consider:

- Switching your RAID array from a RAID 5 to a RAID 10 solution. Each write IO results in two writes using RAID 10 vs. 4 for RAID 5 (100% more efficient with writes).
- As long as the RAID controller has some form of battery backup, then switch its cache mode from Write-through to Write-back, as this setting increases the system’s ability to handle write IOs by an order of magnitude.
- Adding more cache memory to the RAID controller.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Memory Page Faults (Per Second)

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.

Reduce memory page faults

Use the following solutions to reduce the page fault rate if it rises above 100 for a period of time longer than a momentary spike:

- 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Memory Paging

The Memory Paging 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.

Reasons why this metric has no value

SQLdm is unable to collect the corresponding counter due to one of the following reasons:

- It usually takes two refreshes to begin populating the Memory Paging metric, which occurs within a few seconds.
- OS Metrics collection is disabled or not working properly. You must enable OS Metrics collection for SQLdm to collect hardware performance metrics.
- Lightweight Pooling is enabled. SQLdm cannot collect OS metrics if Lightweight Pooling is enabled on the SQL Server instance.
- The counter does not exist on this SQL Server instance.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
Memory Waits

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 SQL Server Waits and Queues view, and review the Microsoft Best Practice Article SQL Server Waits and Queues.

Packets Received (Per Second)

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.

Packets Sent and Received

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.

Page Reads (Per Second)
This metric counts the number of physical reads the disk sub-system performs to bring SQL Server database pages into the data (buffer) cache. For the first few minutes after you start or restart SQL Server, almost any request for data is retrieved from disk. As the data cache fills, these requests are reduced. This value is measured by reading 'PAGE READS/SEC' from sysperfinfo.

Currently, even the fastest disks can support a maximum sustained rate of around only 85 random IOs per second. To determine if your IO subsystem is "flooded," perform the following calculation:

On RAID 10: \[
\frac{((\text{Page reads} + \text{read ahead pages} + (\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 2))}{\text{number of disk spindles in RAID array}} \text{ must be } < 65.
\]

Reduce page reads

If the page reads rate exceeds 65 for more than five seconds every 20 minutes, then consider these solutions:

- 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 is 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) releases that memory for use by both the procedure cache as well as the Data Pages which 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_HEL PEXTENDEDPROC.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- If you are running large, batch-style reporting queries on the OLTP server, then large numbers of data pages are flushed from cache causing much additional IO overhead. In this situation, set up a separate SQL Server Reports computer to service all but the smallest reports. You can feed this Reports Server 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Page Writes (Per Second)

This metric counts the number of physical database writes the disk sub-system performs. This value is measured by reading PAGE READS/SEC from sysperfinfo.

Currently, even the fastest disks can support a maximum sustained rate of around only 85 random IOs per second. To determine if your IO subsystem is "flooded," perform the following calculation:

- On RAID 5: \[
\frac{((\text{Page reads} + \text{read ahead pages} + (\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 4))}{\text{number of disk spindles in RAID array}} \text{ must be } < 65.
\]
- On RAID 10: \[
\frac{((\text{Page reads} + \text{read ahead pages} + (\text{page writes} + \text{checkpoint pages} + \text{lazy writes} + \text{log flushes}) \times 2))}{\text{number of disk spindles in RAID array}} \text{ must be } < 65.
\]

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 DBCC DLLNAME (FREE). Once an extended stored procedure is 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) releases that memory for use by both the procedure cache as well as the Data Pages which 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_HEL PEXTENDEDPROC.
- Allow SQL Server to consume more of the available memory (making sure that OS Paging does not increase).
- If you are running large, batch-style reporting queries on the OLTP server, then large numbers of data pages are flushed from cache causing much additional IO overhead. In this situation, set up a separate SQL Server Reports computer to service all but the smallest reports. You can feed this Reports Server 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
Percent of SQL Server Data Space Used

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Physical Page Reads and Writes

This metric counts the number of physical page reads and writes to disk from the SQL Server instance.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Processor Queue Length

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Re-compiles (Per Second)

The Re-compiles (Per Second) 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. Causes of re-compiles include various events including schema changes, statistics updates, and explicit calls to WITH RECOMPILE. Excessive re-compilation impacts you server performance.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

SQL Server Batches Received (Per Second)

The SQL Server Batches Received (Per Second) metric counts the number of T-SQL batches that SQL Server receives 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 must:

1. Be transmitted to the server using the network.
2. Be parsed by SQL Server to check syntax.
3. Include a compiled execution plan.
4. Include an executed execution plan.
5. Transmit the results back to the client computer or Web.

You must minimize the number and reduce the impact of these T-SQL batches 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 action reduces the size and complexity of the T-SQL batch, resulting in less network traffic and CPU time consumed producing a complex execution plan.
- Grouping as many calling T-SQL commands into a single network batch after you move most of the T-SQL code into stored procedures.
- 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 action dramatically reduces network queues as you make better use of the default network packet size of 4 KB.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

SQL Server CPU Usage (Percent)

The SQL Server CPU Usage (Percent) 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.

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. You can remove most of these reasons by careful coding.
- 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.

For example: SELECT * FROM Northwind.dbo Employees is a fully-qualified object reference whereas SELECT * FROM Employees is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects “as is,” whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, 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. Both of these situations consume a significant amount of CPU time.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
Log Space Used (Percent)

The Log Space Used (Percent) metric calculates the added size all log files used as a percent of summed allocated size of all data files.

A transaction log continues to grow until its inactive portion is truncated. Transaction log truncation occurs because either:

- the BACKUP LOG statement successfully concluded, or
- if the database is using the Simple Recovery Model, then a CHECKPOINT T-SQL statement executes or SQL Server performed an internal automatic CHECKPOINT. SQL Server performs this action 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. No matter which method is used to truncate the log, if there is a perpetually open transaction then the log continues to grow in size until it consumes all available disk space.

Reduce the percentage of the log used

If your transaction log size continues to grow despite issuing BACKUP LOG or CHECKPOINT statements, check for a rogue transaction. Begin your research by sorting the Last Batch column on the SQL Processes grid into ascending mode such that the processes containing the oldest batches appear first. Scroll down the list until you find the first user process with a non-zero number of transactions in its Trans column, and then kill this process. Once this is done, perform a CHECKPOINT or BACKUP LOG to reduce the Log File Used Size.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

SQL Server Memory Usage

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

SQL Server Response Time (ms)

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Total Memory Usage

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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Transaction Log Waits
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.

**Transactions (Per Second)**

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.

**Wait type category - Backup**

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

**Wait type category - I/O Page Latch**

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

**Wait type category - Lock**

The Lock wait type category includes all the waits that are caused when a task is waiting to acquire a lock, including the following:
SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Wait type category - Memory

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

### Wait type category - Non-I/O Page Latch

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

### Wait type category - Non-Page Latch

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
SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Wait type category - Other

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_SPECPROC_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_FORCEOwn_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_COLLCREATE
- HTTP_ENUMERATION
- HTTP_START
- IMP_IMPORT_MUTEX
- IMPROV_IOIWAIT
- 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_RESERVED_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
- QNMANAGER_ACQUIRE
- QPJOB_KILL
- QPJOB_WAITFOR_ABORT
- QRY_MEMGRANT_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_ERRHDL_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_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
**Wait type category - Transaction Log**

The Transaction Log wait type category includes all the waits that are associated with transaction log bottlenecks, including the following:

- LOGBUFFER
- LOGMGR
- WRITELOG

**Metric alerts**

SQLdm offers numerous alerts to successfully monitor and warn you about your SQL Server instances and availability groups. 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:

- Availability Group Estimated Data Loss (Seconds)
- Availability Group Estimated Recovery time (Seconds)
- Availability Group Log Send Queue Size (KB)
- Availability Group Preferred Replica
- Availability Group Redo Queue Size (KB)
- Availability Group Redo Rate (KB/sec)
- Availability Group Role Change
- Availability Group Synchronization Health
- Availability Group Synchronization Performance (Seconds)
- Average Disk Milliseconds Per Read
- Average Disk Milliseconds Per Transfer
- Average Disk Milliseconds Per Write
- Blocked Sessions
- Blocked Sessions Wait Time (Seconds)
- Client Computers
- CLR Enabled
- Cluster Active Node
- Cluster Failover
- Data File Autogrow
- Database Full (Percent)
- Database Full (Size)
- Database Read/Write Error Occurred
- Database Status
- Deadlock
- Disk Reads Per Second
- Disk Transfers Per Second
- Disk Writes Per Second
- Distribution Latency (Seconds)
- DTC Status
- Full-Text Search Status
- Host CPU Usage (Percent)
- Host Memory Swap Detected
- Host Memory Usage (Percent)
- Host Power State
- Index Row Hits (Percent)
- Last Full-Text Catalog Update (Hours)
- Log File Autogrow
- Log Full (Percent)
- Log Full (Size)
- Longest Running Version Store Transaction (Minutes)
- Maintenance Mode Enabled
- Mirror Commit Overhead (Milliseconds)
- Mirrored Server Role Change
- Mirroring Oldest Unsent Transactions
- Mirroring Preferred Configuration
- Mirroring Status
- Mirroring Unrestored Log (KB)
- Mirroring Unsents Log (KB)
- Mirroring Witness Connection
- Non-Distributed Transactions (Count)
- Oldest Open Transaction (Minutes)
- OLE Automation Disabled
- OS Average Disk Queue Length (Count)
- OS Average Disk Queue Length Per Disk (Count)
- OS Disk Free Space (Size)
- OS Disk Full (Percent)
- OS Disk Time (Percent)
- OS Disk Time Per Disk (Percent)
- OS Metrics Collection Status
- OS Paging (Per Second)
- OS Privileged Time (Percent)
- OS Processor Queue Length (Count)
- OS Processor Time (Percent)
- 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 Job Completion
- SQL Server Agent Job Failure
- SQL Server Agent Log
- 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 (Percent)
- SQL Server Data Used (Percent)
- SQL Server Error Log
- SQL Server Log Space Used (Percent)
- SQL Server Memory Usage
- SQL Server Response Time (Milliseconds)
- SQL Server Status
- SQLdm Service Status
- SQLdm Repository Grooming Timed Out
- Table Fragmentation (Percent)
- Tempdb Contention (ms)
- Unsubscribed Transactions (Count)
- Unsubscribed Transactions (Seconds)
- User Connections (Percent)
- Version Store Generation Ratio
- Version Store Size (MB)
- VM CPU Ready Wait Time (ms)
- VM CPU Usage (Percent)
- VM Host Server Change
- VM Memory Swap Delay Detected
- VM Memory Usage (Percent)
- VM Power State
- VM Reclaimed/Ballooned Memory (KB)
- VM Resource Configuration Change
- VM Resource Limits Detected
- WMI Service Unavailable

---

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. [Learn more >>](#)
The Availability Group Estimated Data Loss (Seconds) alert tracks the time period in seconds that data is not synchronized to the secondary replica. This value reflects the time difference of the last commit between the primary and secondary replicas. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see Configure alerts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Availability Group Estimated Recovery time (Seconds) alert

The Availability Group Estimated Recovery time (Seconds) alert provides the required time in seconds that the secondary replica needs to catch up with the primary replica. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see Configure alerts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Availability Group Log Send Queue Size (KB) alert

The Availability Group Log Send Queue Size (KB) alert indicates the amount of log records in KB needed to ship to the secondary replica to complete synchronization. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see Configure alerts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Availability Group Redo Queue Size (KB) alert

The Availability Group Redo Queue Size (KB) alert indicates the amount of log records from log files in KB that need redoing in the secondary replica to complete synchronization. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see Configure alerts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Availability Group Redo Rate (KB/sec) alert

The Availability Group Redo Rate (KB/sec) alert provides the rate in KB per second at which log records are redone in the secondary database to complete synchronization. Alert includes affected databases.

For more information on configuring alerts and setting thresholds, see Configure alerts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Availability Group Role Change alert

The Availability Group Role Change alert indicates when the primary role changes to a secondary replica. Alert includes affected databases for the entire availability group.
Availability Group Synchronization Health alert

The Availability Group Synchronization Health alert provides the database synchronization state in the availability group. Possible values include: not healthy, partially healthy, and healthy.

For more information on configuring alerts and setting thresholds, see Configure alerts.

Availability Group Synchronization Performance (Seconds) alert

The Availability Group Synchronization Performance (Seconds) alert indicates the time required in seconds to complete synchronization between the primary and secondary replicas.

For more information on configuring alerts and setting thresholds, see Configure alerts.

Average Disk Milliseconds Per Read alert

The Average Disk Milliseconds Per Read alert provides the average time (in milliseconds) that a read operation uses data from the disk.

Average Disk Milliseconds Per Transfer alert

The Average Disk Milliseconds Per Transfer alert provides the average time (in milliseconds) used by a disk transfer operation.

Average Disk Milliseconds Per Write alert

The Average Disk Milliseconds Per Write alert provides the average time (in milliseconds) used by an operation writing data to the disk.
Blocked Sessions (Count) alert

The Blocked Sessions (Count) alert provides the number of sessions on the SQL Server instance blocked by other sessions holding requested locks. For this alert to display, enable monitoring of non-query activities with capturing of blocking events.

By default, blocked sessions wait for a lock wait forever, which is not optimum behavior. You can change this default behavior by making a connection, and then adding `SET LOCKTIMEOUT 60000` to the top of the first T-SQL batch.

Reduce the number of blocked sessions

If you experience more than one blocked session on an average day, your site may experience one of the following issues:

- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement with no corresponding `COMMIT TRANSACTION` statement. You must correct the T-SQL.
- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement but where the `COMMIT TRANSACTION` statement is in a following T-SQL batch that is only executed once the end-user confirms the transaction. You must correct the T-SQL.
- The site includes some long-running transactions processing at peak times.

When your transactions run longer than they should and clash, consider drastically reducing the time that a lock is held 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 waits for the preceding one 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.

Blocking Session Wait Time (Seconds) alert

The Blocking Session Wait Time (Seconds) alert provides the time in seconds that a SQL Server session blocks other sessions.

By default, locked sessions waiting for a lock wait forever, which is not optimum behavior from an end-user point of view. You can change this default behavior by adding `SET LOCKTIMEOUT 60000` to the top of the first T-SQL batch after making a connection.

The Active Alerts view provides additional information on the Blocking Session Wait Time (Seconds) alert. To access this information, refer to the right-click context menu after having organized your alert view by metric. This menu grants access to real time alert information, historical view, block details, and online help specific to the alert.

To access similar information on the Blocking View, select a specific SQL Server instance, point to Sessions, and then select the Blocking ribbon.

Blocking Sessions Report

The Blocking Sessions Report displays blocking and blocked sessions' information. To access the Blocking Sessions Report in the Alert view, select the Show Block Details option in the right-click context menu.

Key information provided for blocking and blocked sessions includes:
Reduce blocked sessions wait time

If you experience more than one blocked session on an average day, your site may experience one of the following issues:

- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement with no corresponding `COMMIT TRANSACTION` statement. You must correct the T-SQL.
- T-SQL batches submitted containing a `BEGIN TRANSACTION` statement but where the `COMMIT TRANSACTION` statement is in a following T-SQL batch that is only executed once the end-user confirms the transaction. You must correct the T-SQL.
- The site includes some long-running transactions processing at peak times.

Where your transactions run longer than they should and clash, consider drastically reducing the time that a lock is held by:

- Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary `SELECT` statements.
- Grouping all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible within a transaction with as few `SELECTS` as possible separating them.
- Committing the transaction as soon as possible after the final `DML` statement.
- Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
- Avoiding the use of server-side cursors during a transaction as they slow execution considerably.
- Minimizing or eliminating the number of SQL re-compilations made to the object if a stored procedure and/or trigger is invoked inside a transaction. See the SQL RE-compilations counter for steps to dramatically reduce recompiles.
- Increasing the speed of transaction throughput such that it becomes less likely that one transaction waits for the preceding transaction. You can improve transaction throughput speed 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. Note that each write IO results in two writes using RAID 10 vs. four for RAID 5 (100% more efficient with writes).
  - Switching the RAID controller cache mode from Write-through to Write-back so long as the RAID controller has some form of battery backup. This change 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 the SMP computer.
  - Upgrading the CPU, memory, and motherboard with faster models.
  - Minimizing the number of Context Switches by turning on Use NT Fibers in SQL Server.
  - Switching on the Boost SQL Server Priority.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. [Learn more](#)
The Client Computers (Count) alert provides the number of unique client computers connected to the SQL Server instance. SQLdm measures this value by getting the count of processes from distinct computers in the `sysprocesses` table.

**CLR Enabled alert**

The CLR Enabled alert indicates that the Common Language Runtime (CLR) configuration option is enabled for the SQL Server instance. You can run managed code in the .NET Framework. Use the Server Configuration view to reconfigure this option on SQL Server 2005 instances.

**Cluster Active Node alert**

The Cluster Active Node alert indicates that the active node of the cluster is non-preferred. The active cluster node changes when a failover event occurs. If the original preferred node is offline for an extended period of time, you can change the preferred node setting 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 began.

**Cluster Failover alert**

The Cluster Failover alert indicates a change in the active node. This alert remains 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.

**Prevent cluster failover alerts when you 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.

**Data File Autogrow alert**

The Data File Autogrow alert indicates that a data file autogrow occurred on the specified database. For this alert to display, enable monitoring of non-query activities with capturing of autogrow events. Note that you can avoid unnecessarily long reads and poor response time from SQL Server by minimizing the number of autogrow events occurring in your environment.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

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”.
Database Full (Percent) alert

The Database Full (Percent) alert provides 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, SQLdm auto-growth settings and available disk space. If this value exceeds 80%, consider replacing disks with bigger disks.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see Set general server options.

Calculate the database full percentage using the autogrow settings

When setting the thresholds for your Database Full (Percent) alert, click Advanced > Autogrow Settings and SQLdm provides a choice for you to calculate your displayed percentage. Select Yes, alert on the current used size divided by the maximum possible size to take possible autogrowth into consideration when calculating the database full percentage. Select No, alert on the current used size divided by the current file size to calculate the database full percentage based only on the current size of the database disk file.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Database Full (Size) alert

The Database Full (Size) alert provides the amount of disk space, in megabytes, currently used by the database as the sum of data, text, and indexes. This value represents the actual amount of used data within the database file and does not represent the size of the database file itself.

SQLdm raises this alert based on the database metrics collection interval specified in Server Properties. For more information, see Set general server options.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Database Read/Write Error Occurred alert

The Database Read/Write Error Occurred 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 is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Database Status alert

The Database Status alert indicates that the Database Status metric reached an alert status. This status results 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

This alert is raised based on the diagnostic data and database metrics collection intervals specified in Server Properties. For more information,
Reduce the number of database status issues

Navigate to the Summary 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 is highlighted either in 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.

Deadlock alert

The Deadlock alert indicates that a deadlock occurred on the monitored server. For this alert to display, enable monitoring of non-query activities with capturing of deadlock events.

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.

Resolve deadlocks

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 information helps with diagnosing the problem application and includes the option to export deadlock data to view in SQL Server Management Studio or SQL Server Profiler.

Disk Reads Per Second alert

The Disk Reads Per Second alert provides the rate of read operations on the disk. A high number of disk reads may indicate that SQL server is executing poorly performing queries. Note that this alert is disk specific.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. Note that you need to enable this alert first in the Alert Configuration window to rise.

Disk Transfers Per Second alert

Comprised of both Disk Reads/sec and Disk Writes/sec, the Disk Transfers Per Second alert provides the rate of transfer operations on the disk and helps to determine if a drive does not have enough supporting disks. Note that this alert is disk specific.

A large increase in disk transfers per second may signal a change in workload or a developing I/O or memory pressure issue.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. Note that you need to enable this alert first in the Alert Configuration window to rise.
Disk Writes Per Second alert

The Disk Writes Per Second alert provides the rate of write operations on the disk. Note that this alert is disk specific.

A high number of disk writes per second may indicate that the SQL Server is running low on allocated memory due to large amounts of data being written. It may also point to a situation of memory pressure that occurs when data is not able to remain in cache and has to be flushed to disk frequently.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. Note that you need to enable this alert first in the Alert Configuration window to rise.

Distribution Latency (Seconds) alert

The Distribution Latency (Seconds) alert tracks the amount of time, in seconds, that a replication transaction is at the Publisher waiting for a Distributor to receive it, and has exceeded the warning threshold.

Causes of high distribution latency

The most common cause of high distribution latency is when a transaction has a large amount of commands. Once the transaction is complete, distribution latency decreases.

The Replication view on the Services tab displays a graph with the non-distributed count and the non-distributed time. Graphs that trend upward may indicate a problem that needs resolution.

Reduce distribution latency

You can reduce distribution latency by:

- Reducing traffic on the publisher
- Limiting non-replication related work by the publisher
- Increasing the network bandwidth available to the publisher

DTC Status alert

The DTC Status alert provides the status of the Distributed Transaction Coordinator service.

Host CPU Usage (Percent) alert

The Host CPU Usage (Percent) alert indicates an unusually high amount of CPU usage by the VMWare or Hyper-V host server hosting the virtual
machine on which the monitored SQL Server instance resides. The percentage of host server processor usage is listed under the control. A high host server percentage could indicate a large number of active client sessions. This alert is enabled by default and is available only on instances hosted on virtual machines enabled for VM monitoring.

**Reduce your host CPU usage**

Consistently high host 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 you can remove most of these reasons by careful coding.
- 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.

For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, 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. Both of these situations consume a significant amount of CPU time.

**SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >**

**Host Memory Swap Detected alert**

The Host Memory Swap Detected alert indicates that memory is swapped to disk for the host server hosting the virtual machine on which the monitored SQL Server resides. This may be an indication of memory pressure on the host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLdm VM Configuration window.

**SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >**

**Host Memory Usage (Percent) alert**

The Host Memory Usage (Percent) alert provides the percent of available memory used by the host server hosting the virtual machine on which the monitored SQL Server resides. This alert is enabled by default and is available only on instances hosted on virtual machines enabled for VM monitoring.

**Reduce your host memory usage**

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 infrequently called, then after calling them, issue a `DBCC DDLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- 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. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
  - Wherever possible, using the (NOLOCK) optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` on select statements 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, using the `TABLOCKX` hint table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
  - When deleting all rows in any table, using the `TRUNCATE TABLE` statement instead of the `DELETE` statement as fewer locks and other system resources are consumed in the process.
  - Reducing the time that a lock is held by:
    - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary `SELECT` statements.
    - Grouping all `UPDATES`, `DELETES`, and `INSERTS` as closely as possible within a transaction with as few `SELECTS` as possible separating them.
    - Committing the transaction as soon as possible after the final DML statement.
    - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction
- Allowing 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.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.

**Host Power State alert**

The Host Power State alert provides the power state of the host as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

**Full-Text Search Status alert**

The Full-Text Search Status alert provides the status of the Full-Text Search service.

**Index Row Hits (Percent) alert**

The Index Row Hits (Percent) alert provides the percentage of the density of an index as used by the SQL Query Optimizer to determine index effectiveness. A high density indicates the existence of many duplicates and an inefficient index. A low density indicates high selectivity and an effective index.

This alert is raised based on the database metrics collection interval specified in Server Properties.

This alert is for visual indicators in the Desktop Client only. SQLdm does not send notification when this alert is triggered.

**Last Full-Text Catalog Update (Hours) alert**

The Last Full-Text Catalog Update (Hours) alert provides the time, in hours, since the last Full-Text Catalog update was performed and beyond which the statistics data is considered outdated.

Since full-text indexes are contained in full-text catalogs, when you create new content or change existing content in a catalog, you must regenerate the full text index. This, along with inefficient indexes can impact your monitored SQL Server performance.

SQLdm allows users to optimize full-text catalogs, for details see Monitor Your Full-Text Searches.

Note that this alert is raised based on the diagnostic data collection interval specified in Server Properties.

This alert is for visual indicators in the Desktop Client only. SQLdm does not send notification when this alert is triggered.
Log File Autogrow alert

The Log File Autogrow alert indicates that a log file autogrow occurred on the specified database. For this alert to display, enable monitoring of non-query activities with capturing of autogrow events. Note that you can avoid unnecessarily long reads and poor response time from SQL Server by minimizing the number of Autogrow events occurring in your environment.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

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”.

Log Full (Percent) alert

The Log Full (Percent) alert provides 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 is raised based on the database metrics collection interval specified in Server Properties. For more information, see Set general server options.

Calculate the log full percentage using the autogrow settings

When setting the thresholds for your Log Full (Percent) alert, click Advanced > Autogrow Settings and SQLdm provides a choice for you to calculate your displayed percentage. Select Yes, alert on the current used size divided by the maximum possible size to take possible autogrowth into consideration when calculating the log full percentage. Select No, alert on the current used size divided by the current file size to calculate the log full percentage based only on the current size of the database disk file.

Log Full (Size) alert

The Log Full (Size) alert provides the amount of disk space, in megabytes, currently used by the transaction log for the database. This value represents the actual amount of data within the log and does not represent the size of the database log file itself.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see Set general server options.

Longest Running Version Store Transaction (Minutes) alert
The Longest Running Version Store Transaction (Minutes) alert provides the time in minutes that the longest version store transaction is open and preventing version store cleanup. This alert occurs only on instances running SQL Server 2005 or above.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Maintenance Mode Enabled alert

The Maintenance Mode Enabled 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. You cannot configure this alert.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Mirror Commit Overhead (Milliseconds) alert

The Mirror Commit Overhead (Milliseconds) alert indicates that the commit overhead (in MS) of the mirrored database 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.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

For more information about mirror commit overhead, see the Microsoft document, Use Warning Thresholds and Alerts on Mirroring Performance Metrics (SQL Server).

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Mirrored Server Role Change alert

The Mirrored Server Role Change alert indicates when the role of a mirrored database changes.

When you configure this alert, you can set the time at which the alert resets. Reset this alert when the role of the mirrored database has no impact.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Mirroring Oldest Unsent Transactions (Minutes) alert

The Mirroring Oldest Unsent Transactions (Minutes) alert provides the age of unsent transactions queued on the processor and 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

For more information about oldest unsent transactions, see the Microsoft document, [Use Warning Thresholds and Alerts on Mirroring Performance Metrics (SQL Server)](#).

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Mirroring Preferred Configuration alert

The Mirroring Preferred Configuration alert indicates that a database mirror partner took on a non-preferred role. This event means the roles in the mirroring relationship do not conform to the roles specified as Normal. This alert does not reset over time and remains until you restore the roles or you change the Normal state. You can restore the roles or modify the Normal definition in the **Mirroring** view of the Databases tab.

SQLdm triggers this alert based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Mirroring Status alert

The Mirroring Status alert provides the status of the mirrored database based on the current databases status other than OK. By default, the values Synchronizing and Synchronized equal an OK state. A Suspended database triggers a Warning alert, and a Disconnected or Pending Failover database status triggers the Critical alert.

You can change the status for the mirrored databases in the **Mirroring** view of the **Databases** tab.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Mirroring Unrestored Log (KB) alert

The Mirroring Unrestored Log (KB) alert indicates that the amount of unrestored logs (in KB) queued on the mirrored database 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.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see [Set general server options](#).

For more information about unrestored logs, see the Microsoft document, [Use Warning Thresholds and Alerts on Mirroring Performance Metrics (SQL Server)](#).

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more > >

### Mirroring Unsent Log (KB) alert

The Mirroring Unsent Log (KB) alert indicates that the amount of unsent logs (in KB) queued on the mirrored database 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.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

For more information about unsent logs, see the Microsoft document, Use Warning Thresholds and Alerts on Mirroring Performance Metrics (SQL Server).

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Mirroring Witness Connection alert

The Mirroring Witness Connection alert indicates that the connection to the mirror is lost.

This alert is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Non-Distributed Transactions (Count) alert

The Non-Distributed Transactions (Count) alert provides the number of replication transactions published by the Publisher but not received by the Distributor.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

Oldest Open Transaction (Minutes) alert

The Oldest Open Transaction (Minutes) alert provides the time (in minutes) that the longest running transaction in the database has used since issuing the BEGIN TRANSACTION statement.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

OLE Automation Disabled alert

The OLE Automation Disabled alert indicates that the OLE automation configuration option is disabled in SQL Server 2005 or the procedures are dropped in SQL Server 2000 for the SQL Server instance. SQLdm does not collect data such as OS metrics and service details using OLE automation. Use the OS Metrics tab of the server Properties dialog to reconfigure this option. For additional information about OS metrics collection, see Configure OS metrics monitoring.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

OS Average Disk Queue Length (Count) alert
The OS Average Disk Queue Length (Count) alert provides the average number of both read and write requests queued for all the disks on the SQL Server computer.

**Reduce queue length**

If this queue often remains above six 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 your 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.

**OS Average Disk Queue Length Per Disk (Count) alert**

The OS Average Disk Queue Length Per Disk (Count) alert provides the average number of both read and write requests queued for each disk on the SQL Server computer.

**Reduce queue length**

If this queue often remains above six 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 your 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.

**OS Disk Free Space (Size) alert**

The OS Disk Free Space (Size) alert provides the amount of available disk space, in megabytes, on the SQL Server computer. This alert triggers separately for each disk on a server.

**OS Disk Full (Percent) alert**

The OS Disk Full (Percent) alert provides the percentage of space used on a logical disk on the SQL Server computer. This alert is triggered separately for each disk on a server.
OS Disk Time (Percent) alert

The OS Disk Time (Percent) alert provides the percentage of elapsed time that all disks are busy servicing read and write requests on the SQL Server computer.

Reduce OS processor time

If this value regularly exceeds 75%, you should take action 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 you can remove most of these reasons by careful coding. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- 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. For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, 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. Both of these situations consume a significant amount of CPU time.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

OS Disk Time Per Disk (Percent) alert

The OS Disk Time Per Disk (Percent) alert provides the percentage of elapsed time that an individual disk is busy servicing read and write requests on the SQL Server computer. This alert triggers separately for each disk on a server.

Reduce OS processor time

If this value regularly exceeds 75%, you should take action 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 you can remove most of these reasons by careful coding. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- 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. For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, 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. Both of these situations consume a significant amount of CPU time.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

OS Memory Usage (Percent) alert

The OS Memory Usage (Percent) alert provides the percentage of physical and virtual memory. Consistently high Total Memory Usage could indicate either:

- A memory leak in a program in use.
- A need to increase the amount of total memory.

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.

Reduce OS memory usage
A high percentage of Total Memory Usage indicates a memory bottleneck. To investigate this issue, open the Memory view on the Resources tab to view how memory is 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.

**OS Metrics Collection Status alert**

The OS Metrics Collection Status alert provides the status of SQL diagnostic manager OS Metrics Collection.

**OS Paging (Per Second) alert**

The OS Paging (Per Second) alert indicates that a memory threshold meets or exceeds the setting, 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.

**Reduce memory paging**

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.

**OS Privileged Time (Percent) alert**

The OS Privileged Time (Percent) alert provides the percentage of CPU time spent by all processes executing in privileged mode on the SQL Server computer.

**OS Processor Queue Length (Count) alert**

The OS Processor Queue Length (Count) alert indicates that the number of threads waiting on the processor reached an alert threshold. A consistently high processor queue length can indicate processor congestion. If your system has multiple processors, you may need to increase your alert thresholds.

**Reduce your OS processor queue length**

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 you can remove most of these reasons 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
OS Processor Time (Percent) alert

The OS Processor Time (Percent) alert provides the percentage of CPU time used by all processes on the computer hosting your SQL Server instance. The WMI object `Win32_PerfRawData_PerfOS_Processor` property `PercentProcessorTime` is measured to get percent processor time.

Reduce OS processor time

If this value regularly exceeds 75%, you should take action 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 you can remove most of these reasons by careful coding. See the SQL Re-compilations counter for ways to dramatically reduce recompiles.
- 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. For example, `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects “as is,” whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, 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. Both of these situations consume a significant amount of CPU time.
- Stop unnecessary programs (such as the Print Spooler or activities such as the database server acting as either a BDC or PDC) from running.

Reasons why the CPU Usage metric has no value

SQLdm is unable to collect the corresponding counter due to one of the following reasons:

- It usually takes two refreshes to begin populating the CPU Usage metric, which occurs within a few seconds.
- OS Metrics collection is disabled or not working properly. You must enable OS Metrics collection for SQLdm to collect hardware performance metrics.
- Lightweight Pooling is enabled. SQLdm cannot collect OS metrics if Lightweight Pooling is enabled on the SQL Server instance.
- Alerts are disabled. SQLdm displays an alert status when at least one of the three alerts is enabled.
- The counter does not exist on this SQL Server instance.

OS User Time (Percent) alert

The OS User Time (Percent) alert provides the percentage of CPU time spent by all processes executing in user mode on the SQL Server computer.

Page Life Expectancy alert

The Page Life Expectancy alert provides 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.

Diagnose problems with your SQL Server instance

Microsoft recommends a minimum target for page life expectancy of 300 seconds. This means that any given page in memory is kept in the buffer for five 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.

**Procedure Cache Hit Ratio (Percent) alert**

The Procedure Cache Hit Ratio (Percent) alert provides the ratio of procedure cache hits to procedure cache lookups. This value indicates the reuse of an execution plan from memory as opposed to a compilation from disk.

**Manage your Procedure Cache Hit Ratio**

If you regularly encounter values below 80% once SQL Server is 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. For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, 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. Both of these situations consume a significant amount of CPU time.
- 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 is 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` releases that memory for use by both the procedure cache as well as the Data Pages which has 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_HEL PEYXTENDEDPROC`.
- 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 (Count) alert**

The Query Monitor Events (Count) alert provides 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.

**Session CPU Time (Seconds) alert**

The Session CPU Time (Seconds) alert provides the CPU Time (in seconds) used by the SQL Server session since the last scheduled refresh.

**Session Tempdb Space Usage (MB) alert**

The Session Tempdb Space Usage (MB) alert provides 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 is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

**SQL Server Agent Job Completion alert**

The SQL Server Agent Job Completion alert provides the status of the Agent Job Completion metric. Values for this alert include Succeeded, Retry, Canceled, Failed, and Unknown.

**SQL Server Agent Job Failure alert**

The SQL Server Agent Job Failure alert indicates that the scheduled SQL Server Agent job ended with a Failed status.

**SQL Server Agent Log alert**

The SQL Server Agent Log alert indicates that specific text appeared in the SQL Server Agent error log. You can customize which text triggers this alert by using the Alert Filters tab on the Advanced Alert Configuration window. This tab also allows you to customize the severity of the alert based on the frequency of occurrence.

The default settings are:

- 2 - Warning
- 1 - Critical

**Use a size limit on your SQL Server Agent log**

SQLdm allows you to limit the size of your SQL Server Agent log to help improve response time and avoid using unnecessary space on your server. For additional information and steps on setting this limit, see Configure text and expression alerts.

**SQL Server Agent Long Running Job (Minutes) alert**

The SQL Server Agent Long Running Job (Minutes) alert provides the maximum number of minutes used to complete a scheduled SQL Server Agent job.

**SQL Server Agent Long Running Job (Percent) alert**
The SQL Server Agent Long Running Job (Percent) alert provides the percentage of time that a scheduled SQL Server Agent job uses over the average successful completion time for this job in the current job history.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

### SQL Server Agent Status alert

The SQL Server Agent Status alert provides the status of the SQL Server Agent Service.

### SQL Server Agent XPs Disabled alert

The SQL Server Agent XPs Disabled alert indicates that the Agent XPs configuration option is disabled in SQL Server 2005 or later versions. The alert may also signal that procedures are dropped in SQL Server 2000 for the SQL Server instance. Note that SQLdm does not collect SQL Agent job details.

Agent XPs are disabled when SQL Server Agent is stopped. You can re-enable Agent XPs through the Server Configuration view and allow viewing of agent job histories even when the agent is not running.

### SQL Server CPU Usage (Percent) alert

The SQL Server CPU Usage (Percent) alert indicates an unusually high amount of CPU usage by SQL Server. 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.

**Reduce your SQL Server CPU usage**

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 you can remove most of these reasons by careful coding.
- 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.

For example: `SELECT * FROM Northwind.dbo Employees` is a fully-qualified object reference whereas `SELECT * FROM Employees` is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects “as is,” whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive COMPIL 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. Both of these situations consume a significant amount of CPU time.

### SQL Server Data Used (Percent) alert

The SQL Server Data Used (Percent) alert indicates that you have a high percentage of allocated database space in use.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see Set general server options.
Reduce your SQL Server database space used

- You may need to increase the amount of space allocated to databases, add disks to the computer hosting the instance, or move databases to another host computer.
- If you have autogrowth turned off, this alert indicates that you need to either allocate more space for databases or add more disk space.
- If you have autogrowth turned on and this alert is triggered often, you may need to adjust your autogrowth settings to a higher percentage to improve performance. Constant growth adjustments can degrade performance.
- For SQL Server 2000 users, SQLdm may incorrectly report the file size. This issue occurs when certain columns in the sysindexes table 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:

  DBCC UPDATEUSAGE ('database_name' ) GO

  We recommend that you execute this command during non-peak hours as it is server intensive.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

### SQL Server Error Log alert

The SQL Server Error Log alert indicates that specific text appeared in the SQL Server log. You can customize which text triggers 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

Use a size limit on your SQL Server Error log

SQLdm allows you to limit the size of your SQL Server Error log to help improve response time and avoid using unnecessary space on your server. For additional information and steps on setting this limit, see Configure text and expression alerts.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>

### SQL Server Log Used (Percent) alert

The SQL Server Log Used (Percent) alert provides the percentage of allocated disk space used by logs on the SQL Server instance reached an alert threshold. Once the log space is completely filled, all updates to the event log fail.

This alert is raised based on the database metrics collection interval specified in Server Properties. For more information, see Set general server options.

Reduce your SQL Server log space used

- 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 autogrowth is 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 autogrowth is turned on and you are seeing this alert often, you may need to adjust your autogrowth settings to a higher percentage to improve degraded performance resulting from constant growth adjustments.
- In certain cases, SQLdm may incorrectly report the file size. This issue 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 issue, use the following T-SQL syntax to correct this issue:

  DBCC UPDATEUSAGE ('database_name' ) GO

  We recommend that you execute this command during non-peak hours as it is server intensive.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
The SQL Server Memory Usage (Percent) alert provides the total server memory (Total Server Memory per sysperfinfo) as a percent of total physical memory from WMI.

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 infrequently called, then after calling them, issue a DBCC DLLNAME (FREE). Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A DBCC dllname (FREE) releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.
- 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. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
  - Wherever possible, using the (NOLOCK) optimizer hint or SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED on select statements 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, using the TABLOCKX table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
  - When deleting all rows in any table, using the TRUNCATE TABLE statement instead of the DELETE statement as fewer locks and other system resources are consumed in the process.
  - Reducing the time that a lock is held by:
    - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary SELECT statements.
    - Grouping all UPDATES, DELETES, and INSERTS as closely as possible within a transaction with as few SELECTS as possible separating them.
    - Committing the transaction as soon as possible after the final DML statement.
    - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
    - Allowing 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.
- Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.

The SQL Server Response Time (Milliseconds) alert provides 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.

Reduce your SQL Server response time

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.
SQL Server Status alert

The SQL Server Status alert provides the status of the SQL Server service.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

SQLdm Service Status alert

The SQLdm Service Status alert provides the status of the SQLdm Manager Service. The possible statuses include Running, Stopped, or Unknown. You cannot configure this alert.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Table Fragmentation (Percent) alert

The Table Fragmentation (Percent) alert provides the percentage of fragmented pages in a table. It is the scan density from 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 is raised based on the diagnostic data collection interval specified in Server Properties. For more information, see Set general server options.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Tempdb Contention (ms) alert

The Tempdb Contention (ms) alert provides the latch wait time (in milliseconds) for tempdb allocation maps. The three tracked allocation page types are:

- Page free space (PFS)
- Global allocation map (GAM)
- Shared global allocation map (SGAM)

Latch contention of this sort is usually an indication that you need to create more tempdb data files. In some situation, using Trace Flag 1118 may also alleviate tempdb contention. This alert occurs only on instances running SQL Server 2005 or above.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Unsubscribed Transactions (Count) alert

The Unsubscribed Transactions (Count) alert provides the number of replication transactions received by the Distributor but not received by a Subscriber.
Unsubscribed Transactions (Seconds) alert

The Unsubscribed Transactions (Seconds) alert provides the time (in seconds) that a replication transaction is holding at the Distributor and waiting for a Subscriber to receive it.

User Connections (Percent) alert

The User Connections (Percent) alert provides the percentage of currently-used user connections allowed by the SQL Server instance. A high percentage indicates a load on the SQL Server. Note that the higher this value, the more SQL Server reserves for connections.

Version Store Generation Ratio alert

The Version Store Generation Ratio alert provides 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.

Version Store Size (MB) alert

The Version Store Size (MB) alert provides 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.

VM CPU Ready Wait Time (ms) alert

The VM CPU Ready Wait Time (MS) alert provides the time (in milliseconds) that a virtual CPU is waiting for a physical CPU to become available for processing for the virtual machine on which the monitored SQL Server resides. A spike in this value may indicate CPU pressure on the host server. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.
VM CPU Usage (Percent) alert

The VM CPU Usage (Percent) alert provides the percentage of available virtual CPU time used by the virtual machine hosting the monitored SQL Server. This may differ from the processor utilization percentage reported by the guest OS. The percentage of processor usage is listed under the control. A high percentage could indicate a large number of active client sessions. This alert is disabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

Reduce the percentage of available virtual CPU time used by your VM

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 you can remove most of these reasons by careful coding.
- 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.

For example: SELECT * FROM Northwind.dbo Employees is a fully-qualified object reference whereas SELECT * FROM Employees is a poorly-qualified object. You can reuse the execution plans of fully-qualified objects "as is," whereas plans where you either cannot reuse the not fully-qualified objects or, if they are reused, then they are subject to a highly restrictive COMPIL 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. Both of these situations consume a significant amount of CPU time.

VM Host Server Change alert

The VM Host Server Change alert indicates that the virtual machine hosting the monitored SQL Server instance moved to a different host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLdm VM Configuration window.

VM Memory Swap Delay Detected alert

The VM Memory Swap Delay Detected alert indicates that the memory being swapped to disk is causing a performance delay on the virtual machine on which the monitored SQL Server instance is running. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

VM Memory Usage (Percent) alert

The VM Memory Usage (Percent) alert provides the percentage of available memory used by the virtual machine hosting the monitored SQL Server. This result may vary from the memory usage percentage reported by the guest operating system. This alert is disabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

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 infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.

- 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. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
  - Wherever possible, using the `(NOLOCK)` optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` on select statements 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, using the `TABLOCKX` table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
  - When deleting all rows in any table, using the `TRUNCATE TABLE` statement instead of the `DELETE` statement as fewer locks and other system resources are consumed in the process.
  - Reducing the time that a lock is held by:
    - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary SELECT statements.
    - Grouping all UPDATES, DELETES, and INSERTS as closely as possible within a transaction with as few SELECTS as possible separating them.
    - Committing the transaction as soon as possible after the final DML statement.
    - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
    - Allowing 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.
  - Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.

---

**VM Power State alert**

The VM Power State alert provides the power state of the VM as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

**VM Reclaimed/Ballooned Memory (KB) alert**

The VM Reclaimed/Ballooned Memory (KB) alert provides the amount of memory that is reclaimed by the host server and is no longer available to the virtual machine on which the SQL Server instance is running. This may be a sign of memory pressure on the host. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.

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 infrequently called, then after calling them, issue a `DBCC DLLNAME (FREE)`. Once an extended stored procedure is called, it remains in memory until SQL Server is shut down even if it is never again called, which then wastes available memory. A `DBCC dllname (FREE)` releases that memory for use by both the procedure cache and Data Pages, which has a significant positive effect on both the Procedure Cache Hit Ratio and the Buffer Cache Hit Ratio. Execute `sp_helpextendedproc` to view the extended stored procedures currently loaded in memory by SQL Server.

- 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. To maintain proper system performance and throughput, keep the number of locks to a minimum by:
  - Wherever possible, using the `(NOLOCK)` optimizer hint or `SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED` on...
select statements 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, using the `TABLOCKX` table hint. This table hint prevents SQL Server from initially taking exclusive row locks, granting many of these locks, and then escalating them to an exclusive table lock.
- When deleting all rows in any table, using the TRUNCATE TABLE statement instead of the DELETE statement as fewer locks and other system resources are consumed in the process.
- Reducing the time that a lock is held by:
  - Performing as much work as possible before the transaction performs its first update, delete, or insert. For example, add any necessary SELECT statements.
  - Grouping all UPDATES, DELETES, and INSERTS as closely as possible within a transaction with as few SELECTS as possible separating them.
  - Committing the transaction as soon as possible after the final DML statement.
  - Avoiding any stops for user input once the transaction begins. Be sure to gather all user inputs before the transaction starts.
  - Allowing 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.
  - Limit SQL Server computers to performing only SQL Server work. Stop any unnecessary programs, such as allowing the computer to act as either a primary or backup domain controller.

---

**VM Resource Configuration Change alert**

The VM Resource Configuration Change alert indicates that a resource configuration change occurred for the virtual machine hosting the monitored SQL Server instance. This alert is enabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

**VM Resource Limits Detected alert**

The VM Resource Limits Detected alert indicates that CPU or memory resource limits are set for the virtual machine hosting the monitored SQL Server instance. Resource limits are set by the VM administrator on the total amount of resources that a VM is allowed to consume. This alert is disabled by default and is available only on instances hosted on virtual machines that are enabled for VM monitoring.

**WMI Service Unavailable alert**

The WMI Service Unavailable alert indicates that the WMI service is unavailable on the SQL Server computer. SQL diagnostic manager does not collect OS metrics.

**Availability Group Preferred Replica alert**

The Availability Group Preferred Replica alert triggers when the preferred replica defined for a database changes to a different replica in a set period of time.
For more information on configuring alerts and setting thresholds, see Configure alerts.

**SQL Diagnostic Manager** identifies and resolves SQL Server performance problems before they happen. Learn more >>

SQLdm Repository Timed Out alert

The SQLdm Repository Timed Out alert indicates that grooming timed out on your SQL Server Repository. SQL diagnostic manager no longer deletes stored metrics in the SQLdm Repository at the set schedule.

To review your grooming settings access **Tools > Grooming Options**.

**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
- understand default alert IDs
- troubleshoot WMI connectivity issues
- integrate SQLdm with SCOM

**Migrate or Recover Your Installation**

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.

**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
System diagnostics

The System Diagnostics window allows you to test the connections between the SQLdm Services and other SQLdm components, as well as collect service and desktop client logs. This connection test is designed to assist you when the location of SQLdm components changes 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 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

**One-click System Diagnostics button:**
You can run system diagnostics plus collect service and desktop client logs by clicking on the Run and Gather Logs button. This feature also saves the status of each collected log in a text file.

In a SQLdm distributed environment, the Run and Gather Logs command, gathers only local logs. To collect all product logs, run this option from the machine where your SQLdm services are installed.

**Access the System Diagnostics window:**

To open the System Diagnostics window, select Help > System diagnostics from the Toolbar menu.

In a console-only installation the Service Configuration Tool button is not available since the SQLdm Management Service Configuration Wizard is installed in the machine where all the SQLdm services reside.

**Choose to migrate or recover SQLdm**

When choosing whether a migration or disaster recovery strategy is best for your environment, consider if you 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 is advantageous during disaster recovery, especially if the only available computer is in another domain. Verify that a two-way trust exists between the target domains to support full communication between the Repository and consoles.

<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, first back up and then restore the Repository database.

**Create your migration or recovery plan**

Once you decide whether a SQLdm Migration or Recovery is best for your environment, create a plan so that when the time comes, you are prepared and get SQL diagnostic manager up and running as quickly as possible.

**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.

**Creating a Recovery Plan**

A disaster recovery plan details the steps to remedy unexpected outages to make sure 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.

**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 you want to host the Repository and Services, make sure this instance meets or exceeds the product requirements as well as these specific requirements.

- You can dedicate the target instance to hosting the SQL diagnostic manager only
- The target instance runs 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 changed the default recovery model to full, make sure your backup includes all 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 changed the default recovery model to full, make sure your backup includes all transaction logs.

Review disaster recovery guidelines

Make sure 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Migrate your SQLdm installation

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 order for a successful migration.

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 requires a new license key. You can easily request a new license key by sending an email to: licensing@idera.com.

To complete the request, make sure you have 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 Start > Control Panel > Uninstall a Program 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 installs 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 SQL diagnostic manager on the Quick Start window.
4. On the Welcome window of the setup program, 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 folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
7. Select your setup type and then click Next.
8. Browse and select the SQL Server instance your SQLdm Repository was restored to and click Next.
9. Enter your service account credentials and click Next.
10. 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.
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 Quick Start on the window.
4. On the Welcome window of the setup program, 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 folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
7. Select the Custom Install setup type, and then click Next.
8. Select SQLdm Services (also select the Desktop Client if you want to run it on the same computer), and then click Next.
9. Specify which SQL Server instance is hosting the Repository database, verify that the displayed database name is correct, and then click Next.
10. Specify the account credentials that you want the SQLdm services to use to connect to the SQLdm Repository and the monitored SQL Server instances, and then click Next. This account must belong to the sysadmin role on the SQLdm Repository database and each instance you plan to monitor.
11. Click Install to begin your installation.
12. Click Finish to exit the setup program.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Migrate the SQLdm Repository

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 order for a successful migration.

Get a New SQLdm License Key

Because you are migrating the Repository database to a different SQL Server instance, SQLdm requires 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 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 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 > 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more

---

**Migrate the SQLdm services**

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 order for 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 installs 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. On the Welcome window of the setup program, click Next.
4. Review and accept the license agreement by clicking I accept the terms in the license agreement, and then click Next.
5. Accept the default folder for your SQLdm installation, or click Change to specify a different folder, and then click Next.
6. Select your setup type and then click Next.
7. Browse and select the SQL Server instance on the original computer that contains your SQLdm Repository, and click Next.
8. Enter your service account credentials and click Next.
9. 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 SQLdm in a clustered environment

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, Windows Server 2008, and Windows Server 2012 environments.

- Installing in a Windows Server 2003 Clustered Environment
- Installing in a Windows Server 2008 Clustered Environment
- Installing in a Windows Server 2012 Clustered Environment
- Configure alerts for important cluster events

Windows Server 2003 clustered environment

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

- Create a separate cluster resource group to host SQLdm services. A separate group allows the SQLdm services to fail without causing other resources in the group to failover.
- Before you create this separate cluster resource group, gather a dedicated IP, NetBIOS name, and a dedicated disk for the cluster resource group.
- SQLdm Collection, Management, and Predictive Analytics services can run as generic services on your newly-created cluster resource group.
- The system may request the network Name and network IP as part of cluster configuration.

Install SQLdm Services on Cluster Nodes

You must install the SQLdm services on each cluster node for the services to work correctly when a failure occurs on the primary cluster node hosting the SQLdm services.

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 SQL diagnostic manager on the Quick Start window.
4. On the Welcome window of the setup program, 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.

You cannot perform the installations concurrently, as the installers collide when checking the repository. You must perform the installations 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. 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:
• 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.

3. 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.

4. In the Cluster Administrator tool, add the 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.

5. Select a Disk that is in a Cluster Resource Group and create a directory on that disk used to hold local data for the services.

6. On the primary node, create the default registry key and add the full data path value under
HKEY_LOCAL_MACHINE\SOFTWARE\Idera\SQLdm\Default.

7. Create the DataPath value.

8. Select a Cluster Resource Group in which you want to define the SQLdm resources.

9. 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.

10. 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

11. 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

12. 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

13. 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

14. 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.
Complete the Cluster Configuration in SQLdm

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.

Windows Server 2008 clustered environment

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

- Create a separate cluster resource group to host SQLdm services. A separate group allows the SQLdm services to fail without causing other resources in the group to failover.
- Before you create this separate cluster resource group, gather a dedicated IP, NetBIOS name, and a dedicated disk for the cluster resource group.
- SQLdm Collection, Management, and Predictive Analytics services can run as generic services on your newly-created cluster resource group.
- The system may request the network Name and network IP as part of cluster configuration.

Install SQLdm Services on Cluster Nodes

You must install the SQLdm services on each cluster node for the services to work correctly when a failure occurs on the primary cluster node hosting the SQLdm services.

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 SQL diagnostic manager on the Quick Start window.
4. On the Welcome window of the setup program, 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.
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 you want to use 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, right-click on 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 name of service and disk, and then click Apply
   - On the Registry Replication tab, if blank, add SOFTWARE\Idera\SQLdm\Default\DataPath
6. In Services and Applications, right-click the newly created service and select Add a Resource and Generic Service.
7. Select the SQLdm Collection Service.
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 General tab, select Use Network Name for computer name
   - On the Dependencies tab, add the name of service and disk, and then click Apply
   - On the Registry Replication tab, add SOFTWARE\Idera\SQLdm\Default\DataPath
10. In Services and Applications, right-click the newly created service 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 name of service and disk, and then click Apply
   - 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诊断管理器

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 Diagnosis Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Windows Server 2012 clustered environment

The following instructions guide you through the installation of SQLdm in a Windows Server 2012 based clustered environment.

Things to consider before you install SQLdm in a clustered environment

- Create a separate cluster resource group to host SQLdm services. A separate group allows the SQLdm services to fail without causing other resources in the group to failover.
- Before you create this separate cluster resource group, gather a dedicated IP, NetBIOS name, and a dedicated disk for the cluster resource group.
- SQLdm Collection, Management, and Predictive Analytics services can run as generic services on your newly-created cluster resource group.
- The system may request the network Name and network IP as part of cluster configuration.

Install SQLdm Services on Cluster Nodes

You must install the SQLdm services on each cluster node for the services to work correctly when a failure occurs on the primary cluster node hosting the SQLdm services.

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 SQL diagnostic manager on the Quick Start window.
4. On the Welcome window of the setup program, 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.
Configure the Cluster Resource Group

Cluster Resource Group settings must be modified to allow SQLdm to recognize the cluster nodes. Note that in certain versions of 2012, the Registry Replication tab is missing. A PowerShell command, provided in the instructions below constitutes a workaround for this issue.

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 you want to use 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 Roles> Configure Role...
   - In the Select Role tab, select Generic Service
   - In the Select Service tab, select the SQLdm Management Service
   - Enter the network name and IP address for the SQLdm services
   - Select the clustered storage that hosts the path defined in step 1
   - Provide the Registry Replication defined in step 2
   - Verify the settings and click Next to complete the wizard
5. In Roles, click the newly created service name and stop Role
6. In Roles, select the newly created service name and select the Resources tab found in the bottom
7. Right-click on the SQLdm Management Service and select Properties
   - On the Dependencies tab, add the name of service and disk, and then click Apply
   - On the General tab, select Use Network Name for computer name, and click OK
   - To configure the Registry Replication entry, open PowerShell with administrator rights and execute the following command:
     
     ```powershell
     Add-ClusterCheckpoint -ResourceName "SQLdm Management Service (Default)"
     -RegistryCheckpoint "SOFTWARE\Idera\SQLdm\Default\DataPath"
     ```
8. In Roles, right-click the newly created service and select Add a Resource and then select Generic Service
9. Select the SQLdm Collection Service.
10. In Roles, select Next and finish out the wizard.
11. Right-click on the SQLdm Collection Service and select Properties.
    - On the Dependencies tab, add the name of service and disk, and then click Apply
    - On the General tab, select Use Network Name for computer name, and click OK
    - To configure the Registry Replication entry, open PowerShell with administrator rights and execute the following command:
      
      ```powershell
      Add-ClusterCheckpoint -ResourceName "SQLdm Collection Service (Default)"
      -RegistryCheckpoint "SOFTWARE\Idera\SQLdm\Default\DataPath"
      ```
12. In Roles, right-click the newly created service and select Add Resource and then select Generic Service
13. Select the Predictive Analytics service from the list of resources
14. Select Next and then Finish to exit the wizard
15. Right-click on the SQLdm Predictive Analytics Service and select Properties
    - On the Dependencies tab, add the name of service and disk, and then click Apply
    - On the General tab, select Use Network Name for computer name, and click OK
    - To configure the Registry Replication entry, open PowerShell with administrator rights and execute the following command:
      
      ```powershell
      Add-ClusterCheckpoint -ResourceName "SQLdm Predictive Analytics Service (Default)"
      -RegistryCheckpoint "SOFTWARE\Idera\SQLdm\Default\DataPath"
      ```
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.
19. 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.
20. 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 identifies and resolves SQL Server performance problems before they happen. Learn more > >

Deploy Idera Newsfeed in a clustered environment

You can deploy the SQLdm Mobile & Newsfeed Service to a clustered environment for continuous remote alerting. See the following topics for installation and configuration instructions for Windows Server 2003, Windows Server 2008, and Windows Server 2012 environments.

- Installing in a Windows Server 2003 Clustered Environment
- Installing in a Windows Server 2008 Clustered Environment
- Installing in a Windows Server 2012 Clustered Environment

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

Deploy the Idera Newsfeed to a Windows 2003 cluster

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

- 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 fail over. 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 resource cluster group makes sure that all four services fail over together, and does not require additional resources.

- The SQLdm Mobile & Newsfeed Service runs 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 need to provide credentials for the service account and the name of a SQL Server instance that can host the SQLdm Mobile Repository database. Make sure these components meet the requirements.
- Install or upgrade your clustered SQLdm Services to the latest version of SQLdm.

Install the Idera Newsfeed Platform on each cluster node

Install the platform on each cluster node to set up the Idera Newsfeed Platform to work correctly when the primary cluster node fails.

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 SQLdm Mobile and Newsfeed on the Quick Start window.
4. On the Welcome window of the setup program, 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 you want the setup program to use to create the Repository database, test the connection, and then click Next.
10. Indicate which credentials you want the SQLdm Mobile & Newsfeed Service account to 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 uses 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 maps 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 is 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

Make sure 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 identifies and resolves SQL Server performance problems before they happen. Learn more >>

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

• 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 makes sure that all four services failover together, and does not require additional resources.
• The SQLdm Mobile & Newsfeed Service runs 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 need to provide credentials for the service account and the name of a SQL Server instance that can host the SQLdm Mobile Repository database. Make sure these components meet the requirements.
• Install or upgrade your clustered SQLdm Services to the latest version of SQLdm.

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 SQLdm Mobile and Newsfeed on the Quick Start window.
4. On the Welcome window of the setup program, 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 Collection Service
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 `\Idera\SQLdm Mobile & Newsfeed\` 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 you want to 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:
   - SQLdm Mobile & Newsfeed Service
   - SQLdm Management Service
   - SQLdm Collection Service
   - SQLdm Predictive Analytics

Test your cluster configuration using the SQLdm Console

Make sure 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 identifies and resolves SQL Server performance problems before they happen. Learn more > >

Deploy the Idera Newsfeed to a Windows 2012 cluster

Use the following instructions to deploy the Idera Newsfeed Platform in a Windows Server 2012 clustered environment.

Things to consider before you install the Idera Newsfeed Platform in a clustered environment

- 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 makes sure that all four services failover together, and does not require additional resources.
- The SQLdm Mobile & Newsfeed Service runs 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 need to provide credentials for the service account and the name of a SQL Server

- SQLdm Predictive Analytics
instance that can host the SQLdm Mobile Repository database. Make sure these components meet the requirements.

- Install or upgrade your clustered SQLdm Services to the latest version of SQLdm.

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 SQLdm Mobile and Newsfeed on the Quick Start window.
4. On the Welcome window of the setup program, 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 Services, 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

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 ...\Idera\SQLdm Mobile & Newsfeed\ 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 Roles, and then click Configure Role.
   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 you want to 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.
4. To configure the Registry Replication entry, open PowerShell with administrator rights and execute the following command:
   Add-ClusterCheckpoint -ResourceName "(Newsfeed Role Name)" -RegistryCheckpoint "SOFTWARE\Idera\SQLdm Mobile and Newsfeed\Default\DataPath"

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.
   - On the General tab, select Use Network Name for the computer name.
   - On the Dependencies tab, specify the Disk and Name used by the service.
   - 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. ClickNext to finish the wizard.
6. In Windows Services, bring the following services online:
   - SQLdm Mobile & Newsfeed Service
   - SQLdm Management Service
   - SQLdm Collection Service
   - SQLdm Predictive Analytics

Test your cluster configuration using the SQLdm Console

Make sure 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
1. 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 identifies and resolves SQL Server performance problems before they happen. Learn more > >

<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 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>No.</td>
<td>Description</td>
<td>Type</td>
<td>Details</td>
<td>Alert Message</td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>------</td>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>10</td>
<td>SQL Server Agent Status</td>
<td>Services</td>
<td>Status of the SQL Server Agent service.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Database Read/Write Error Occurred</td>
<td>Databases</td>
<td>An I/O error occurred while attempting to read from or write to a database file.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SQL Server Status</td>
<td>Services</td>
<td>Status of the SQL Server service.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>SQL Server Memory Usage (Percent)</td>
<td>Resources</td>
<td>Percentage of memory used by the SQL Server instance out of the total memory on the SQL Server computer.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Database Status</td>
<td>Databases</td>
<td>Operational status of the database, such as Normal, Offline, or Suspect.</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Unsubscribed Transactions (Seconds)</td>
<td>Services</td>
<td>Time in seconds that a replication transaction is holding at the Distributor and waiting for a Subscriber to receive it. This alert is for visual indicators in the Desktop Client only. No notifications are sent.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>DTC Status</td>
<td>Services</td>
<td>Status of the Distributed Transaction Coordinator service.</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Full-Text Search Status</td>
<td>Services</td>
<td>Status of the Full-Text Search service.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Index Row hits (Percent)</td>
<td>Databases</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>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Last Full-Text Catalog Update (Hours)</td>
<td>Services</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>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>Value</td>
<td>Category</td>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>----------</td>
<td>--------</td>
<td>-------------</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, 4011 - Other, 4012 - Disabled, 4013 - Unavailable</td>
<td>Operational</td>
<td>OS Metrics Collection Status</td>
<td>Status of SQL diagnostic manager OS Metrics Collection.</td>
</tr>
<tr>
<td>24</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>25</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>26</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>27</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>28</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>29</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>30</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>31</td>
<td>4090</td>
<td>Resources</td>
<td>OS Average Disk Queue Length (Count)</td>
<td>Average number of both read and write requests queued for all the disks on the SQL Server computer.</td>
</tr>
<tr>
<td>32</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>33</td>
<td>2838</td>
<td>Sessions</td>
<td>Session CPU Time (Seconds)</td>
<td>Time in seconds that the SQL Server session is blocking other sessions.</td>
</tr>
<tr>
<td>34</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>Column 1</td>
<td>Column 2</td>
<td>Column 3</td>
<td>Column 4</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>35</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>48</td>
<td>4001 - Enabled 4002 - Disabled</td>
<td>Operational</td>
<td>Maintenance Mode Enabled</td>
<td>Maintenance Mode is enabled for the SQL Server instance. SQL diagnostic manager does not collect scheduled data or process alerts.</td>
</tr>
<tr>
<td>49</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. You can run managed code in the .NET Framework.</td>
</tr>
<tr>
<td>50</td>
<td>4210</td>
<td>Operational</td>
<td>OLE Automation Disabled</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 use OLE automation to 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>Row</td>
<td>Column</td>
<td>Category</td>
<td>Event Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>----------</td>
<td>------------</td>
<td>-------------</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>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 Unrestored 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>Page</td>
<td>Resource ID</td>
<td>Resource Type</td>
<td>Property Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>---------------</td>
<td>---------------</td>
<td>-------------</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>
<tr>
<td>77</td>
<td>4500</td>
<td>Services</td>
<td>Cluster Failover</td>
<td>A cluster failover has occurred.</td>
</tr>
<tr>
<td>78</td>
<td>4510</td>
<td>Services</td>
<td>Cluster Active Node</td>
<td>The active node of the cluster is a non-preferred node. The preferred node may be configured under Server Properties &gt; Cluster Settings.</td>
</tr>
<tr>
<td>79</td>
<td>5120</td>
<td>Services</td>
<td>Distribution Latency (Seconds)</td>
<td>Time in seconds that a replication transaction has been holding at the Publisher and waiting to be received by a Distributor.</td>
</tr>
<tr>
<td>80</td>
<td>2840</td>
<td>Sessions</td>
<td>Deadlock</td>
<td>A deadlock occurred on the monitored server. Note that the Activity Monitor &gt; Capture deadlocks (SQL 2005+) check box must be enabled for this alert to function.</td>
</tr>
<tr>
<td>81</td>
<td>4140</td>
<td>Resources</td>
<td>Procedure Cache Hit Ratio</td>
<td>The percentage of procedure cache hits to cache lookups.</td>
</tr>
<tr>
<td>82</td>
<td>4150</td>
<td>Resources</td>
<td>Average Disk Milliseconds Per Read</td>
<td>Average time in milliseconds of a read operation of data from the disk. This is a per-disk metric.</td>
</tr>
<tr>
<td>83</td>
<td>4160</td>
<td>Resources</td>
<td>Average Disk Milliseconds Per Transfer</td>
<td>Average time in milliseconds of a disk transfer operation. This is a per-disk metric.</td>
</tr>
<tr>
<td>84</td>
<td>4170</td>
<td>Resources</td>
<td>Average Disk Milliseconds Per Write</td>
<td>Average time in milliseconds of a write operation of data to the disk. This is a per-disk metric.</td>
</tr>
<tr>
<td>85</td>
<td>4180</td>
<td>Resources</td>
<td>Disk Reads Per Second</td>
<td>Rate of read operations on the disk. This is a per-disk metric.</td>
</tr>
<tr>
<td>Row</td>
<td>ID</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>----</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>4185</td>
<td>Resources</td>
<td>Disk Transfers Per Second</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rate of transfer operations on the disk. This is a per-disk metric.</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>4190</td>
<td>Resources</td>
<td>Disk Writes Per Second</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rate of write operations on the disk. This is a per-disk metric.</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>3140</td>
<td>Services</td>
<td>SQL Server Agent Job Completion</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The SQL Server Agent job/step has completed.</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>5130</td>
<td>Databases</td>
<td>Version Store Generation Ratio</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Amount in percent by which the tempdb version store generation rate exceeds the cleanup rate. This alert is only raised on instances running SQL Server 2005 or above.</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>5131</td>
<td>Databases</td>
<td>Version Store Size (MB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Size in megabytes of the tempdb version. This alert is only raised on instances running SQL Server 2005 or above.</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>5135</td>
<td>Databases</td>
<td>Longest Running Version Store Transaction (Minutes)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Time in minutes that the longest version store transaction has been opened and preventing version store cleanup. This alert may only be raised on instances running SQL Server 2005 or above.</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>5136</td>
<td>Databases</td>
<td>Session Tempdb Space Usage (MB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Amount of tempdb space being used by a running session. This alert is only raised on instances running SQL Server 2005 or above.</td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>5132</td>
<td>Databases</td>
<td>Tempdb Contention (ms)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Current wait time for tempdb allocation maps (GAM SGAM and PFS) in milliseconds. This alert is only raised on instances running SQL Server 2005 or above.</td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>1115</td>
<td>Databases</td>
<td>Log File Autogrow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Log file autogrow occurred on the specified database. Enable the &quot;Activity Monitor&gt;Capture Autogrow check box for this alert to function.</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>1116</td>
<td>Databases</td>
<td>Data File Autogrow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data file autogrow occurred on the specified database. Enable the Activity Monitor&gt;Capture Autogrow check box for this alert to function.</td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Document ID</td>
<td>Category</td>
<td>Event Type</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>----------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>96</td>
<td>5200</td>
<td>Virtualization</td>
<td>VM Resource Configuration Change</td>
<td>A resource configuration change occurred for the virtual machine hosting the monitored SQL Server instance. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>97</td>
<td>5201</td>
<td>Virtualization</td>
<td>VM Host Server Change</td>
<td>The virtual machine hosting the monitored SQL Server instance moved to a different host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>98</td>
<td>5202</td>
<td>Virtualization</td>
<td>VM CPU Usage (Percent)</td>
<td>The percentage of available virtual CPU time used by the virtual machine hosting the monitored SQL Server. This may differ from the processor utilization percentage reported by the guest OS. A high percentage may indicate a large number of active client sessions. This alert is disabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>99</td>
<td>5250</td>
<td>Virtualization</td>
<td>Host CPU Usage (Percent)</td>
<td>The percentage of available CPU time used by the host server hosting the virtual machine on which the monitored SQL Server is currently running. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>Number</td>
<td>Code</td>
<td>Module</td>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>100</td>
<td>5203</td>
<td>Virtualization</td>
<td>VM Memory Usage (Percent)</td>
<td>The percentage of available memory used by the virtual machine hosting the monitored SQL Server. This may differ from the memory usage percentage reported by the guest OS. This alert is disabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>101</td>
<td>5251</td>
<td>Virtualization</td>
<td>Host Memory Usage (Percent)</td>
<td>The percentage of available memory used by the host server hosting the virtual machine on which the monitored SQL Server is currently running. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>102</td>
<td>5204</td>
<td>Virtualization</td>
<td>VM CPU Ready Wait Time</td>
<td>The time in milliseconds that a virtual CPU has been waiting for a physical CPU to become available for processing for the virtual machine on which the monitored SQL Server is currently running. A spike in this value may indicate CPU pressure on the host server. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>103</td>
<td>5205</td>
<td>Virtualization</td>
<td>VM Reclaimed/Ballooned Memory (KB)</td>
<td>The amount of memory that has been reclaimed by host and is no longer available to the virtual machine on which the monitored SQL Server instance is running. A spike in this value may be a sign of memory pressure on the host. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>Page</td>
<td>DocID</td>
<td>Category</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>----------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>104</td>
<td>5206</td>
<td>Virtualization</td>
<td>VM Memory Swap Delay Detected</td>
<td>Memory being swapped to disk is causing a performance delay on the virtual machine on which the monitored SQL Server instance is running. This is triggered if the Memory Swap In Rate (KB/s) or CPU Swap Rate (ms) counters are non-zero. This alert is enabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>105</td>
<td>5252</td>
<td>Virtualization</td>
<td>Host Memory Swap Detected</td>
<td>Memory is being swapped to disk for the host server hosting the virtual machine on which the monitored SQL Server is currently running. This may be an indication of memory pressure on the host. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter which have been enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>106</td>
<td>5207</td>
<td>Virtualization</td>
<td>VM Resource Limits Detected</td>
<td>Either CPU or memory resource limits are set for the virtual machine hosting the monitored SQL Server instance. Resource limits are soft limits set by the VM Administrator on the total amount of resources that a VM is allowed to consume. This alert is disabled by default and is only relevant to instances hosted on VMWare vCenter virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>107</td>
<td>5208</td>
<td>Virtualization</td>
<td>VM Power State</td>
<td>This is the state of the VM as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>Page</td>
<td>Row</td>
<td>Section</td>
<td>Metric/Details</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>---------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>5253</td>
<td>Virtualization</td>
<td>Host Power State</td>
<td>This is the state of the host as reported by the vCenter or Hyper-V host servers. This alert is enabled by default and is only relevant to instances hosted on virtual machines enabled for VM monitoring in the SQLdm VM Configuration window.</td>
</tr>
<tr>
<td>109</td>
<td>1113</td>
<td>Databases</td>
<td>Database Full (Size)</td>
<td>The amount of disk space (in MB) currently used by the database as the sum of data text and indexes. This value is the actual amount of used data within the database file not the size of the database file itself.</td>
</tr>
<tr>
<td>110</td>
<td>1214</td>
<td>Databases</td>
<td>Log Full (Size)</td>
<td>The amount of disk space (in MB) currently used by the transaction log for this database. This value is the actual amount of data within the log not the size of the database log file itself.</td>
</tr>
<tr>
<td>111</td>
<td>4101</td>
<td>Resources</td>
<td>OS Disk Free Space (Size)</td>
<td>Amount of space (in MB) available on a logical disk on the SQL Server computer.</td>
</tr>
<tr>
<td>116</td>
<td>5301</td>
<td>Databases</td>
<td>Availability Group Role Change</td>
<td>Primary role changed to a secondary replica. Alert includes affected database(s) for the entire availability group.</td>
</tr>
<tr>
<td>117</td>
<td>5302</td>
<td>Databases</td>
<td>Availability Group Estimated Data Loss (Seconds)</td>
<td>The time period in seconds that data is not synchronized to the secondary replica. Value reflects the time difference of the last commit between the primary and secondary replicas. Alert includes affected database(s).</td>
</tr>
<tr>
<td>118</td>
<td>5303</td>
<td>Databases</td>
<td>Availability Group Synchronization Health</td>
<td>Reflects database synchronization state in the availability group. Possible values include: not healthy partially healthy and healthy.</td>
</tr>
<tr>
<td>119</td>
<td>5304</td>
<td>Databases</td>
<td>Availability Group Estimated Recovery Time (Seconds)</td>
<td>The required time in seconds that the secondary replica needs to catch up with the primary replica. Alert includes affected database(s).</td>
</tr>
</tbody>
</table>
Troubleshoot WMI connectivity issues

In order to use WMI, the user account under which the Collection Service runs must have administrator permissions on the remote server. The most frequently encountered problems with WMI relate to RPC traffic failing to get through to the remote computer. Another issue involves DCOM/WMI permissions.

The following links 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 to help you test and troubleshoot WMI issues.

To use WbemTest:

1. Run wbmtest.exe.
2. Click Connect.
3. In the Namespace text box, enter `\server\root\cimv2` where server is the name of the server to which you want to connect.
4. Click Connect.
5. Click Query.
6. Enter `select * from win32_process.`
7. Click Apply.

If WbemTest connects to the remote server and issues 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 the Microsoft document, 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. Make sure that 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 and output similar to:
   ```
   Completed 1 calls in 15 ms
   66 T/S or 15.000 ms/T
   ```
4. Make sure that local or internal network firewalls do not block traffic by either disabling the firewall or configuring 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 impersonation of any connection as Guest.
   a. Open the Local Security Policy console from Administrative Tools.
   d. Change the settings from Guest Only to Classic.
2. Make sure 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. Make sure Enable Distributed COM on this computer is checked.
3. Verify the configuration of the correct DCOM remote launch and activation permissions:
   a. Run `DcomCnfg` on the remote server.
   b. Click Component Services.
   c. Expand Computers.
   d. Right click My Computer and select Properties.
   e. Make sure 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. Make sure 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. Make sure 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. Make sure 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.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more >>
Integrate SQLdm with SCOM

The Idera SQLdm Management Pack (SQLdm MP) integrates and extends the Microsoft System Center Operations Manager (SCOM) monitoring capabilities by providing database administrators and IT operators with additional insight into the availability and performance of their SQL Server interfaces, all from one console.

SQLdm MP includes automatic discovery of SQLdm installations and of the SQL Server instances managed from those installations. Once discovered, SQLdm MP propagates the current status, health, and events of monitored instances up to SCOM.

Understanding the SQLdm Management Pack health model

The SQLdm Management Pack reports the health of each SQL Server instance monitored by SQLdm, as well as the server health of SQLdm itself. The following main characteristics help define the SQLdm health model:

SQLdm Server

The health of SQLdm is tracked by monitoring the state of three underlying services, using the following three service monitors:

- Collection Service Monitor
- Management Service Monitor
- Predictive Service Monitor

The automatically discovered Object ID for the SQLdm Service is: Idera.SQLDM.Service

SQLdm Instance

The health of each SQL server instance being monitored by SQLdm is rolled up by reflecting the Alerts counts opened for that instance within SQLdm.

If a SQL server instance currently has one or more Critical alerts open, then the Health of that instance is recorded as Critical. If one or more Warning alerts are open, but no Critical alerts show, then the health of the instance is recorded as Warning.

In Health Explorer the Diagnostic Manager Alert status which drove a specific Health state change is recorded and displayed in the State Change Events tab.

The automatically discovered Object ID for a SQL Instance is: Idera.SQLDM.SqlInstance

Install and configure the SQLdm Management Pack for SCOM

Before importing the SQLdm Management Pack for SCOM, you must import the relevant Microsoft Windows management packs for your current operating system. For example, if SQLdm is installed on Windows Server 2008 R2, then you must install the management pack for that OS before continuing.

You can find the SQLdm MP (Idera.SQLDM.mp) in the root of the installation kit.

Configuring your installation to support SCOM

The SQLdm Management Pack requires that the SCOM agent has at least read access to the SQLdm Repository. Grant this access before installing the SQLdm MP and enabling discovery of any SQLdm installed instances.

SQLdm MP also requires that a SCOM agent (version 2007 or above) be installed locally on each server where an instance of SQLdm is present. If an agent is not already installed, then use the Discovery Wizard in the SCOM Console to push the agent installation.

Importing the SQLdm Management Pack for SCOM

The SQLdm Management Pack import process loads into the SCOM database the SQLdm MP settings, classes, and rules necessary for monitoring SQLdm. The import replicates the relevant rules and objects to all systems where SQLdm is installed.
The following steps help you import the SQLdm Management Pack when it is downloaded to a hard drive local to the SCOM Console. For more information about other methods of importing the Management Pack, see How to Import a Management Pack in Operations Manager 2007.

To import the SQLdm Management Pack:

1. Open the Operations Console for Operations Manager, and then click the Administration button.
2. In the Actions pane or from Actions Actions menu option, click Import Management Packs.
3. Click Add, and then click Add from disk.
4. In the Online Catalog Connection dialog, click No. If you did not preload the required OS Management Packs and you have a current Internet connection, click Yes.
5. Browse for and then select the Idera SQL Diagnostic Manager Management Pack.
6. Click Install. Discovery of SQLdm servers and monitored SQL Server instances begins automatically as the Management Pack is deployed to the relevant agents across the network.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >

View SQLdm details in SCOM

With the SQLdm Management Pack, all the key monitors employed by SQL DBAs are now fully visible to System Center Operations Manager operators, synchronizing IT operations and DBA functions and identifying situations that could impact SQL Server availability and performance.

The following table lists the SQLdm views available within SCOM:

<table>
<thead>
<tr>
<th>View Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL DM Server Health</td>
<td>This Health view lists all discovered SQL Diagnostic Manager servers in the environment. The health state of each instance is displayed, together with information about the specific SQL Server instance.</td>
</tr>
<tr>
<td>SQL Instance Alerts</td>
<td>Displays a list of current Alerts generated when a specific SQL Server instance monitored by SQL Diagnostic Manager has entered an unhealthy state.</td>
</tr>
<tr>
<td>SQL Instance Events</td>
<td>This view offers a list of all Alert transitions recorded by SQL Diagnostic Manager for each SQL Server instance which is being monitored. This view provides a complete history of the status of all monitored SQL Server instances.</td>
</tr>
<tr>
<td>SQL Instance Health</td>
<td>The Health view displays the current health state of each SQL Server instance monitored by a SQL Diagnostic Manager server.</td>
</tr>
<tr>
<td>Dashboard</td>
<td>This view combines the SQL DM Server Health, SQL Instance Health and the SQL Server instance events views into a single pane.</td>
</tr>
</tbody>
</table>

To access the SQLdm views available in SCOM:

1. In the Monitoring pane of the System Center Operations Manager Console, locate the Idera SQLdm folder.
2. Expand the tree view by clicking + sign next to SQLdm.

SQL Diagnostic Manager identifies and resolves SQL Server performance problems before they happen. Learn more > >