SolarWinds

Configuring & Integrating AppInsight for SQL

solarwinds

Unexpected Simplicity

AppInsight for SQL

SAM offers a detailed view of your SQL databases' performance without the use of agents or templates by using the *AppInsight for SQL* embedded application. *AppInsight for SQL* provides a level of detail and expert knowledge[†] far beyond what a SQL template can provide, allowing you to monitor virtually every aspect of your SQL instances and databases.

Like any unassigned application in SAM, AppInsight for SQL is considered a template until it is applied. Therefore, it is a member of the Application Monitor Templates collection.

Once applied to a node, AppInsight for SQL is considered an application. Like any SAM application, AppInsight for SQL is comprised of multiple component monitors, also known as performance counters.

AppInsight for SQL general information:

- AppInsight for SQL Requirements and Permissions
- AppInsight for SQL Licensing
- Template/Application Relationship
- Adding AppInsight for SQL to a Node
- Editing the AppInsight for SQL Template
- Understanding the AppInsight for SQL Details View
- Navigating the AppInsight for SQL Details View

Information about individual AppInsight for SQL resources can be found at the following locations:

- AppInsight for SQL Details view, which is comprised of the following sub-views:
    - AppInsight for SQL Details View (Summary Tab)
    - AppInsight for SQL Details View (Queries Tab)
    - Database Details View
- Performance Counter Details View

† Portions of the performance counter information used in the AppInsight for SQL feature are provided courtesy of multiple sources listed in the copyright and attribution section at the beginning of this document.

AppInsight for SQL Requirements and Permissions

AppInsight for SQL data is collected at the same default five minute polling interval as traditional application templates. Following are the requirements and permissions needed for AppInsight for SQL:

Important: AppInsight for SQL does not require named-pipes; however, it does require TCP. For example, SAM uses TCP detection during discovery. You may receive an error message pertaining to "named-pipes.“ This is the result of the the last client protocol that is tried during connection to the SQL server.

AppInsight for SQL Requirements

AppInsight for SQL supports the following versions of Microsoft SQL Server:

Microsoft SQL Server Version	Versions Supported
Microsoft SQL Server 2008	Without SP SP1 SP2 SP3
Microsoft SQL Server 2008R2	Without SP SP1 SP2
Microsoft SQL Server 2012	Without SP SP1

AppInsight for SQL Permissions

The minimum SQL permissions required to use AppInsight for SQL are as follows:

- Must be a member of the db_datareader role on the msdb system database.
- Must have VIEW SERVER STATE permissions.
- View any definition.
- Connect permission to Master database.

- Execute permission on the Xp_readerrorlog stored procedure.
- Connect permission to the Msdb database.
- Must be member of db_datareader role in the Msdb database.
- Connect permission to all databases.

Note: AppInsight for SQL supports both the SNMP and WMI protocols and uses SQL to gather information about the application. Additional information is available for nodes managed via WMI.

The following script will configure permissions:

```
USE master

GRANT VIEW SERVER STATE TO AppInsightUser

GRANT VIEW ANY DEFINITION TO AppInsightUser

EXEC sp_adduser @loginame = 'AppInsightUser' ,@name_in_
db = 'AppInsightUser'

GRANT EXECUTE ON xp_readerrorlog TO AppInsightUser

USE msdb

EXEC sp_adduser @loginame = 'AppInsightUser' ,@name_in_
db = 'AppInsightUser'

EXEC sp_addrolemember N'db_datareader',
N'AppInsightUser'

EXECUTE sp_MSforeachdb 'USE [?]; EXEC sp_adduser
@loginame  = ''AppInsightUser'', @name_in_db =
''AppInsightUser'''
```

AppInsight for SQL Licensing

When using AppInsight for SQL, 50 component monitors will count against your licensed number of component monitors per SQL instance.

For example, if you have an active license for 1,500 component monitors and use AppInsight for SQL to monitor 88 databases over ten instances of SQL, 500 component monitors will count against your total license. The number of databases you have on these instances is irrelevant.

> *(50 component monitors X 10 SQL Server instances = 500 component monitors used.)*

This will leave you with 1,000 component monitors available for use elsewhere.

(1,500 component monitors – 500 component monitors used for AppInsight for SQL = 1,000 component monitors remaining).

There is no restriction on the number of databases you can have monitored. Also note that if you choose not to use this application, you will not be penalized any number of component monitors.

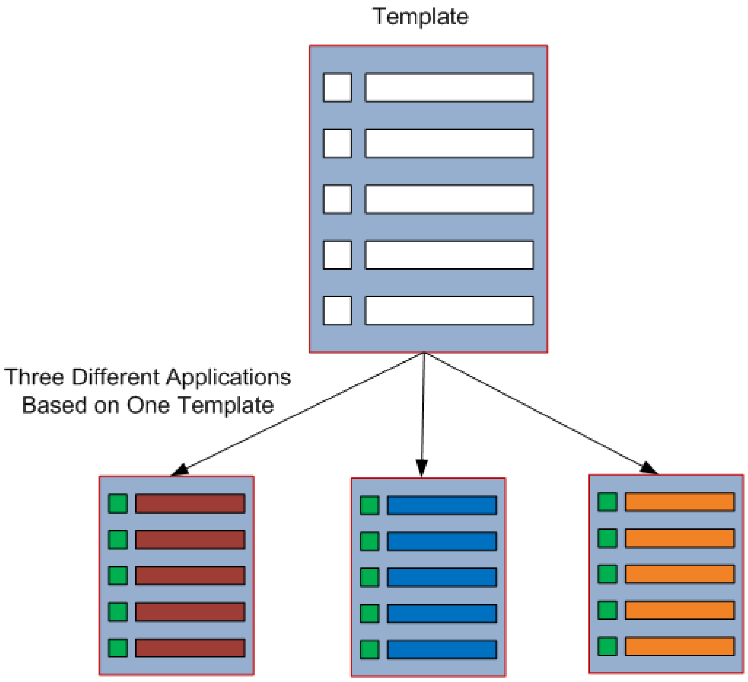
For more information, see "Licensing" on page 123.

Template/Application Relationship

The following illustration explains the Template and Application relationship and is true for all templates, including AppInsight for SQL.

Here you can see that if you change something at the template level, the applications based on that template will be affected. Conversely, if you change something on the application level, only the individual application will be affected.

This inheritance relationship is beneficial if you need to make a great deal of changes quickly. For example, rather than change one item on 100 applications that are based on a single template (which requires 100 changes), you can more easily change the one item on the template. That one change in the template will trickle down to all 100 applications that are based on the template.

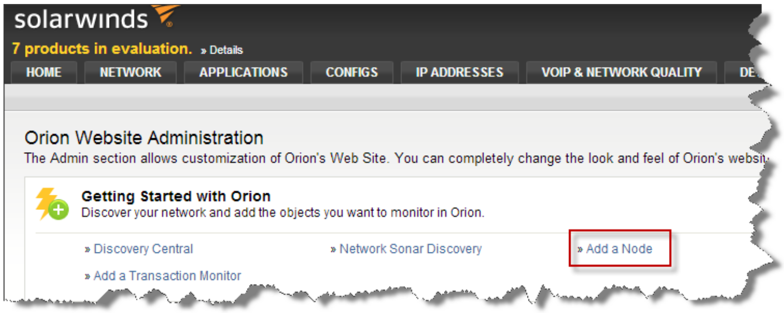


Adding AppInsight for SQL to a Node

There are multiple ways to add AppInsight for SQL to a node, via **Node Discovery**, manually adding it via the *Manage Application Monitor Templates* screen, adding it via the *Management* resource, adding it via **Network Sonar Discovery**, or adding it via a **Scheduled Discovery**.

To Add AppInsight for SQL to a Node via Discovery:

1. From the web console, navigate to **Settings > Add a Node** in the **Getting Started with Orion** section.



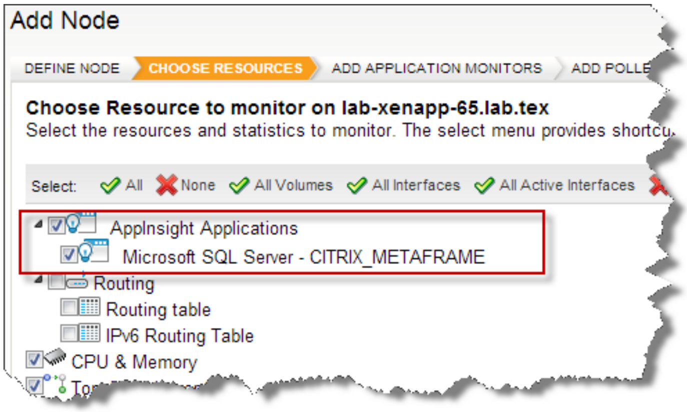
Orion Website Administration

The Admin section allows customization of Orion's Web Site. You can completely change the look and feel of Orion's websi

Getting Started with Orion

Discover your network and add the objects you want to monitor in Orion.

2. Fill out the information on the **Define Node** step, and then click **Next**.
3. On the **Choose Resources** step, check the **AppInsight for SQL** applications you want. AppInsight for SQL applications have a unique blue and white icon, as shown:


Add Node

Choose Resource to monitor on lab-xenapp-65.lab.tex

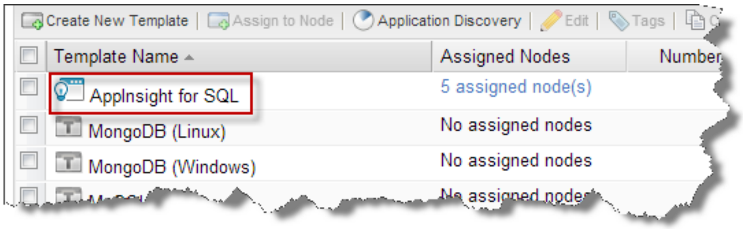
Select the resources and statistics to monitor. The select menu provides shortcu

- ☑ AppInsight Applications
  - ☑ Microsoft SQL Server - CITRIX_METAFRAME
- □ Routing
  - □ Routing table
  - □ IPv6 Routing Table
- ☑ CPU & Memory
- ☑ To

4. Click **Next** and complete the wizard as instructed.

To Add AppInsight for SQL to a Node via Manage Application Templates:

1. From the web console, navigate to **Settings > SAM Settings > Manage Templates**.



☐	Template Name ▲	Assigned Nodes	Number
☐	AppInsight for SQL	5 assigned node(s)	
☐	MongoDB (Linux)	No assigned nodes	
☐	MongoDB (Windows)	No assigned nodes	
☐		No assigned nodes	

2. Select **AppInsight for SQL** by checking its box and click **Assign to Node**.
3. Complete the required fields in the wizard and click **Assign Application Monitor**.

To Add AppInsight for SQL via the Management Resource:

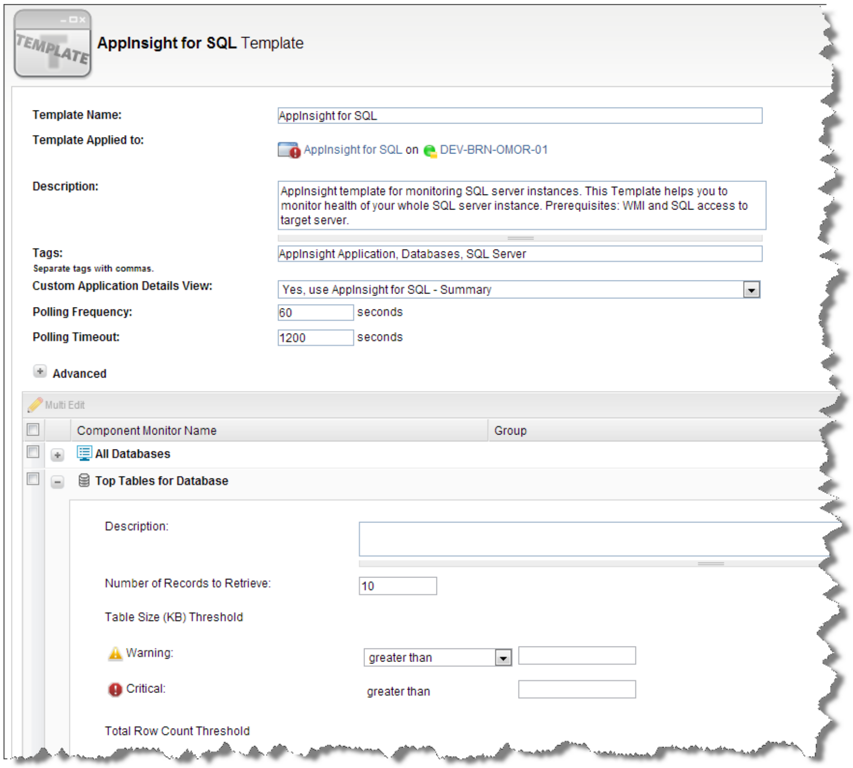
1. Navigate to the *Node Details* view by clicking any node.
2. From the *Management* resource, click **List Resources**.
3. Check the **AppInsight for SQL** box to enable AppInsight for SQL data collection.

Editing the AppInsight for SQL Template

The AppInsight for SQL application behaves like any other application in SAM. And like other applications, AppInsight for SQL's template can be edited.

Editing the AppInsight for SQL Template:

1. From the web console, navigate to **Settings > SAM Settings > Manage Templates**.
2. Check the **AppInsight for SQL** template in the list and then click **Edit**.
3. From here, you can edit any and all component monitors within the template; including threshold values, number of records to retrieve, component monitor description, and so on.



AppInsight for SQL Template

Template Name: AppInsight for SQL

Template Applied to: AppInsight for SQL on DEV-BRN-OMOR-01

Description: AppInsight template for monitoring SQL server instances. This Template helps you to monitor health of your whole SQL server instance. Prerequisites: WMI and SQL access to target server.

Tags: AppInsight Application, Databases, SQL Server
Separate tags with commas.

Custom Application Details View: Yes, use AppInsight for SQL - Summary

Polling Frequency: 60 seconds

Polling Timeout: 1200 seconds

Advanced

Multi Edit

	Component Monitor Name	Group
□	**All Databases**	
□	**Top Tables for Database**	

Description:

Number of Records to Retrieve: 10

Table Size (KB) Threshold

Warning: greater than

Critical: greater than

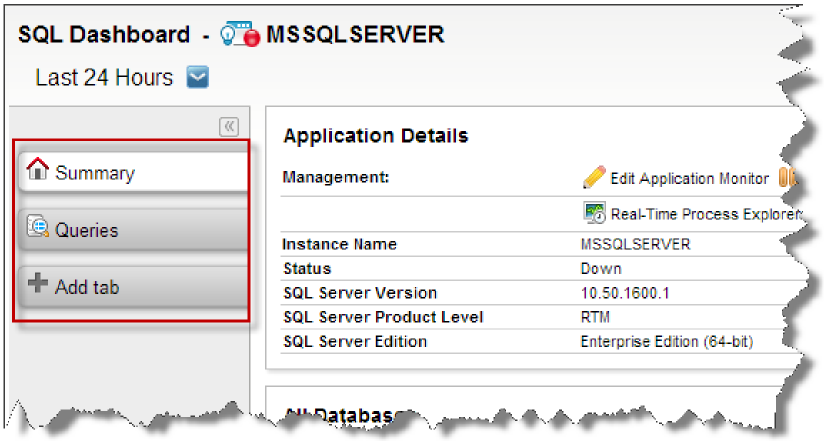
Total Row Count Threshold

4. When done editing, click **Submit** at the bottom of the screen.

Understanding the AppInsight for SQL Details View

The AppInsight for SQL Details view is the equivalent of the *Application Details* view of a traditional SAM application; however, there are some notable differences:

- *Details View* - Unlike a traditional *Application Details* view, the *AppInsight for SQL Details* view is comprised of two tabs, or sub-views, that host a variety of resources and information:
  - *Summary Tab* – This view displays an informative summary about the current SQL instance.
  - *Queries Tab* – This view displays details about the most expensive queries running across all databases on the current SQL instance.

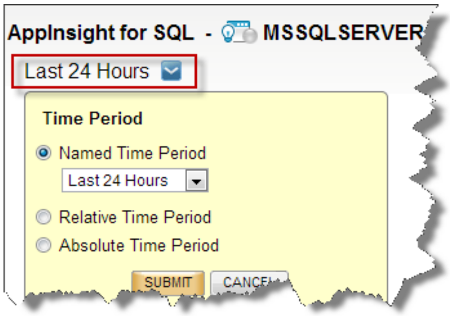


Application Details

Management:	Edit Application Monitor Real-Time Process Explorer
Instance Name	MSSQLSERVER
Status	Down
SQL Server Version	10.50.1600.1
SQL Server Product Level	RTM
SQL Server Edition	Enterprise Edition (64-bit)

- *Database Details View* - This view displays details about the selected database.
- *Performance Counter Details View* - This view displays details about the selected performance counter.

  From here, you can select from the dropdown menu and choose to display different amounts of data based on a time range, as shown below:



Time Period

● Named Time Period

Last 24 Hours

○ Relative Time Period

○ Absolute Time Period

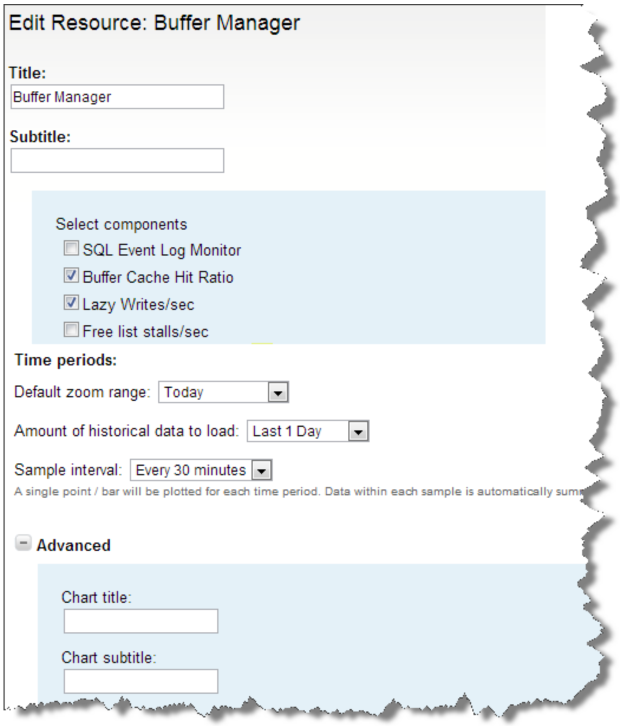
Editing Resources

Both the titles and subtitles of each resource can be changed by clicking **Edit** in the upper right-hand corner of any resource.

The time and date displayed at the top of each resource (where applicable) shows the range of data currently loaded into the resource. This, along with the Sample Interval Range, can be changed by clicking **Edit** from within the resource.

Additionally, certain resources will allow you to add and remove component monitors from the resource. You can do this by clicking the **Edit** button, and then checking the component monitors you want to keep.

Note: Pertinent data for each resource is grouped together by default.



Edit Resource: Buffer Manager

Title: Buffer Manager

Subtitle:

Select components

- ☐ SQL Event Log Monitor
- ☑ Buffer Cache Hit Ratio
- ☑ Lazy Writes/sec
- ☐ Free list stalls/sec

Time periods:

Default zoom range: Today

Amount of historical data to load: Last 1 Day

Sample interval: Every 30 minutes

A single point / bar will be plotted for each time period. Data within each sample is automatically sum

⊟ Advanced

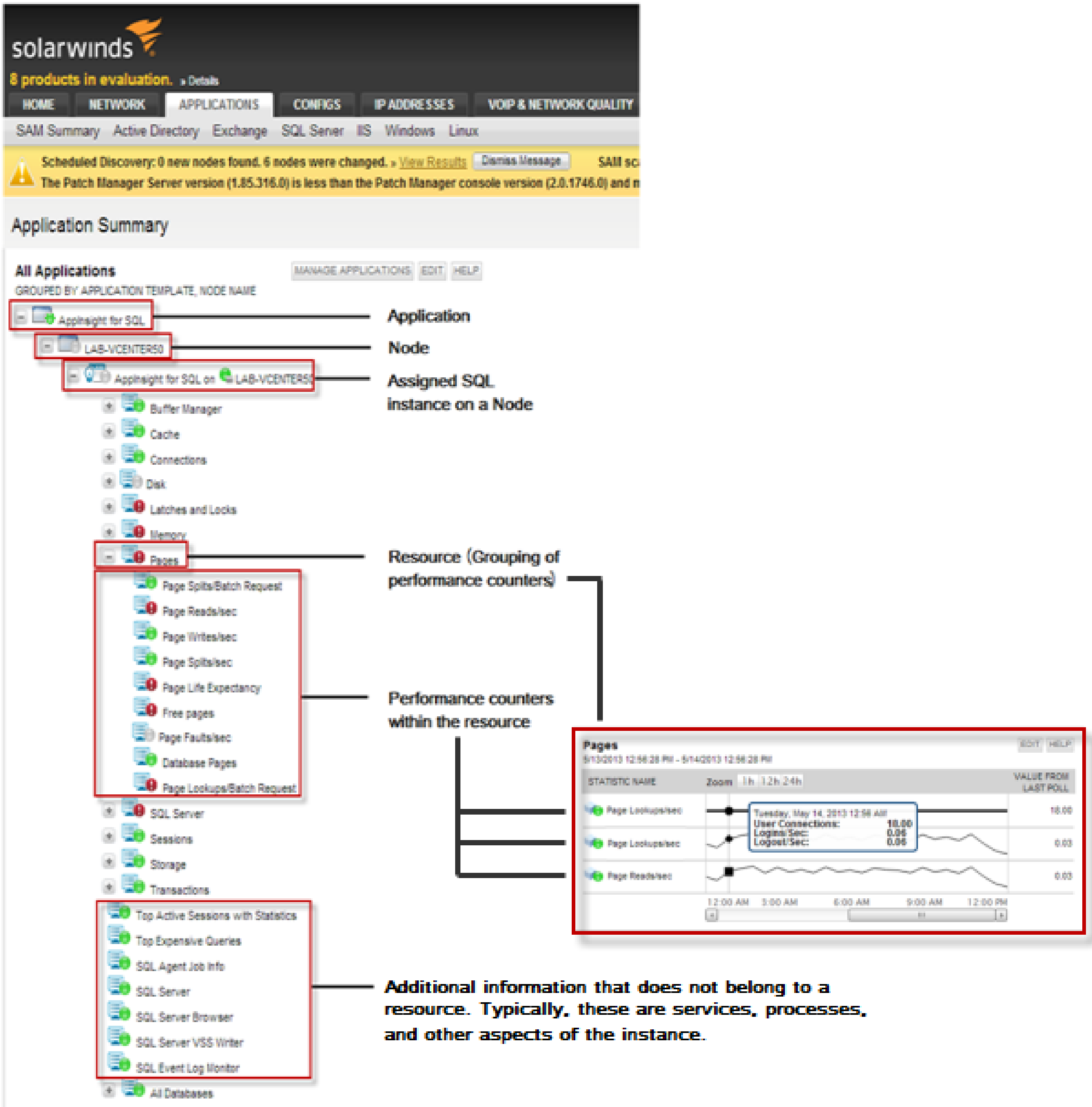
Chart title:

Chart subtitle:

Navigating the AppInsight for SQL Details View

AppInsight for SQL acts like any other application and can therefore be found in the All Applications resource on the Application Summary view. By default, sub-views are turned on once you navigate to the *AppInsight for SQL Details* page.

Below is the AppInsight for SQL application with the **All Applications** resource detailed on the *Summary* tab of the *Application Details* page:



Scheduled Discovery: 0 new nodes found. 6 nodes were changed. » View Results SAM sc

The Patch Manager Server version (1.85.316.0) is less than the Patch Manager console version (2.0.1746.0) and n

Tuesday, May 14, 2013 12:56 AM
User Connections: 10.00
Logins/Sec: 0.06
Logout/Sec: 0.06

Application

Node

Assigned SQL instance on a Node

Resource (Grouping of performance counters)

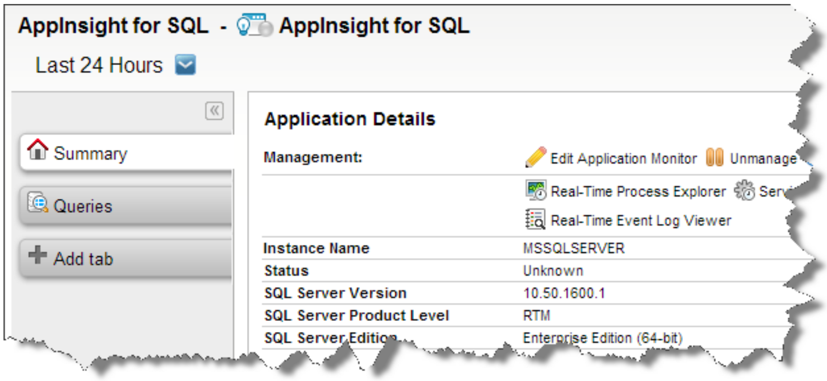
Performance counters within the resource

Additional information that does not belong to a resource. Typically, these are services, processes, and other aspects of the instance.

To Navigate to the AppInsight for SQL Details View:

1. Navigate to the *All Applications* resource.
2. Expand the *AppInsight for SQL* tree by clicking **[+].**

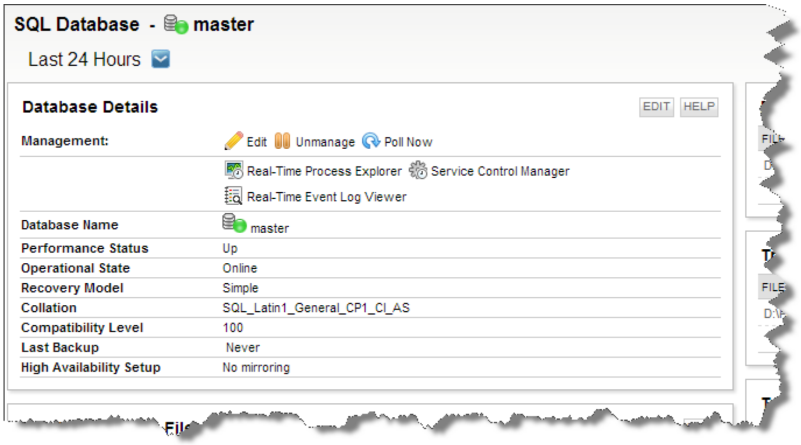
3. Select a SQL instance to view by clicking it.


AppInsight for SQL - AppInsight for SQL
Last 24 Hours
Summary
Queries
Add tab
Application Details
Edit Application Monitor
Unmanage
Real-Time Process Explorer
Ser
Real-Time Event Log Viewer

Management:	Edit Application Monitor Unmanage Real-Time Process Explorer Ser Real-Time Event Log Viewer
Instance Name	MSSQLSERVER
Status	Unknown
SQL Server Version	10.50.1600.1
SQL Server Product Level	RTM
SQL Server Edition	Enterprise Edition (64-bit)

To Navigate to the Database Details view:

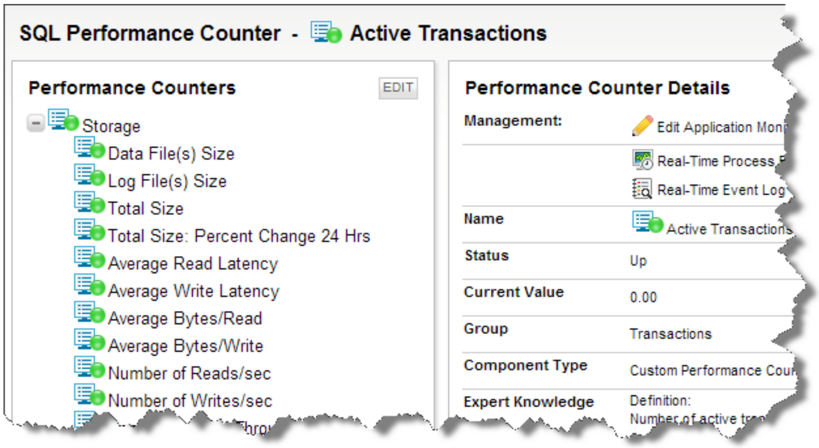
1. From the *AppInsight for SQL Details* view, click a database in **All Databases** resource.



Management:	Edit Unmanage Poll Now Real-Time Process Explorer Service Control Manager Real-Time Event Log Viewer
Database Name	master
Performance Status	Up
Operational State	Online
Recovery Model	Simple
Collation	SQL_Latin1_General_CP1_CI_AS
Compatibility Level	100
Last Backup	Never
High Availability Setup	No mirroring

To Navigate to the Performance Counter Details view:

1. From the *AppInsight for SQL Details* view, click a performance counter in any resource.



SQL Performance Counter - Active Transactions

Performance Counters EDIT

- Storage
  - Data File(s) Size
  - Log File(s) Size
  - Total Size
  - Total Size: Percent Change 24 Hrs
  - Average Read Latency
  - Average Write Latency
  - Average Bytes/Read
  - Average Bytes/Write
  - Number of Reads/sec
  - Number of Writes/sec

Performance Counter Details

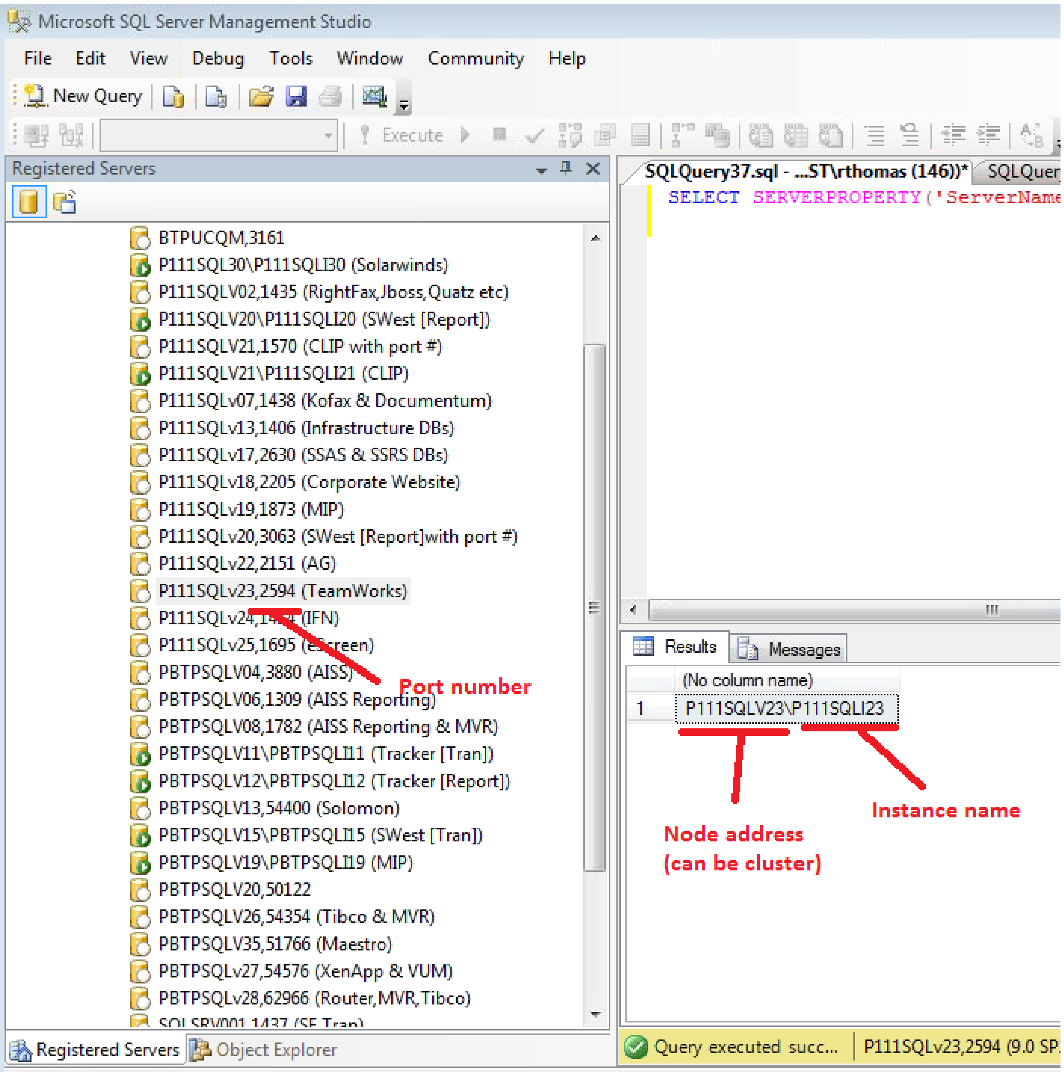
Management:	Edit Application Moni Real-Time Process E Real-Time Event Log
Name	Active Transactions
Status	Up
Current Value	0.00
Group	Transactions
Component Type	Custom Performance Cou
Expert Knowledge	Definition: Number of active tra

Monitoring Clusters with AppInsight for SQL

To start monitoring a SQL server instance which is clustered, take the following steps:

1. Connect to the instance that you want monitor using SQL Management Studio
2. Execute the following query to make sure you have the proper target node and instance name:
   ```
   SELECT SERVERPROPERTY('ServerName')
   ```

3. Determine the IP address of the target node (e.g. “`ping P111SQLV23`”)

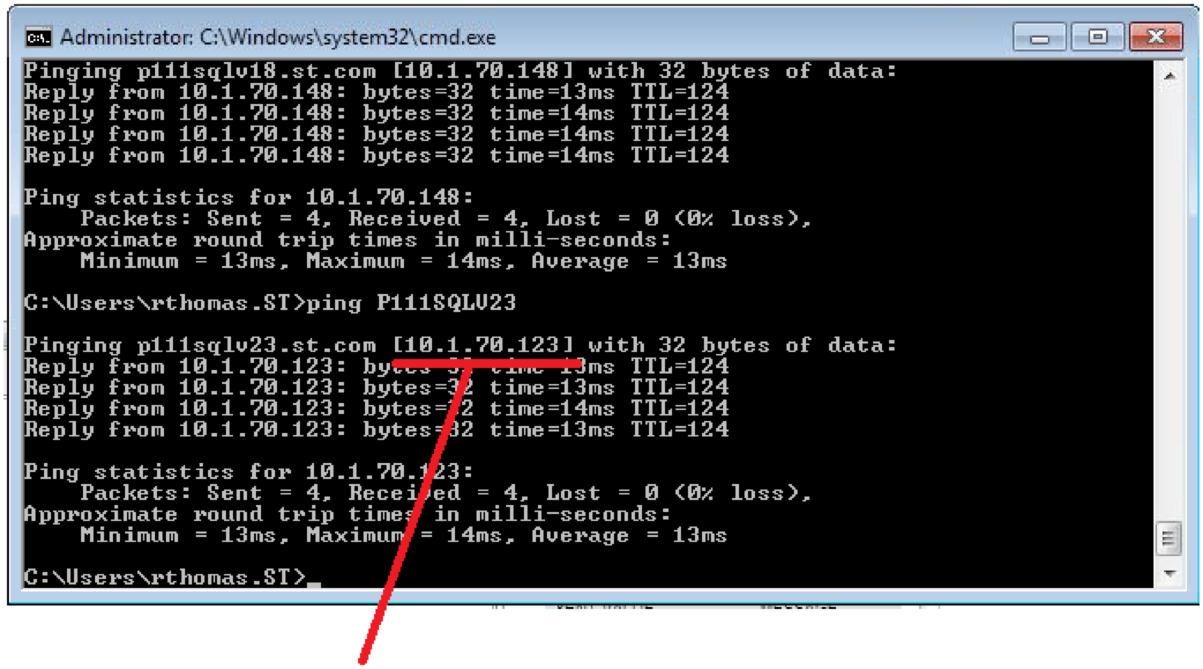


BTPUCQM,3161
P111SQL30\P111SQLI30 (Solarwinds)
P111SQLV02,1435 (RightFax,Jboss,Quatz etc)
P111SQLV20\P111SQLI20 (SWest [Report])
P111SQLV21,1570 (CLIP with port #)
P111SQLV21\P111SQLI21 (CLIP)
P111SQLv07,1438 (Kofax & Documentum)
P111SQLv13,1406 (Infrastructure DBs)
P111SQLv17,2630 (SSAS & SSRS DBs)
P111SQLv18,2205 (Corporate Website)
P111SQLv19,1873 (MIP)
P111SQLv20,3063 (SWest [Report]with port #)
P111SQLv22,2151 (AG)
P111SQLv23,2594 (TeamWorks)
P111SQLv24,14[illegible] (IFN)
P111SQLv25,1695 (eScreen)
PBTPSQLV04,3880 (AISS)
PBTPSQLV06,1309 (AISS Reporting)
PBTPSQLV08,1782 (AISS Reporting & MVR)
PBTPSQLV11\PBTPSQLI11 (Tracker [Tran])
PBTPSQLV12\PBTPSQLI12 (Tracker [Report])
PBTPSQLV13,54400 (Solomon)
PBTPSQLV15\PBTPSQLI15 (SWest [Tran])
PBTPSQLV19\PBTPSQLI19 (MIP)
PBTPSQLV20,50122
PBTPSQLV26,54354 (Tibco & MVR)
PBTPSQLV35,51766 (Maestro)
PBTPSQLv27,54576 (XenApp & VUM)
PBTPSQLv28,62966 (Router,MVR,Tibco)
SQLSRV001,1437 (SE Tran)

```
SELECT SERVERPROPERTY('ServerName
```

	(No column name)
1	P111SQLV23\P111SQLI23

Query executed succ... | P111SQLv23,2594 (9.0 SP

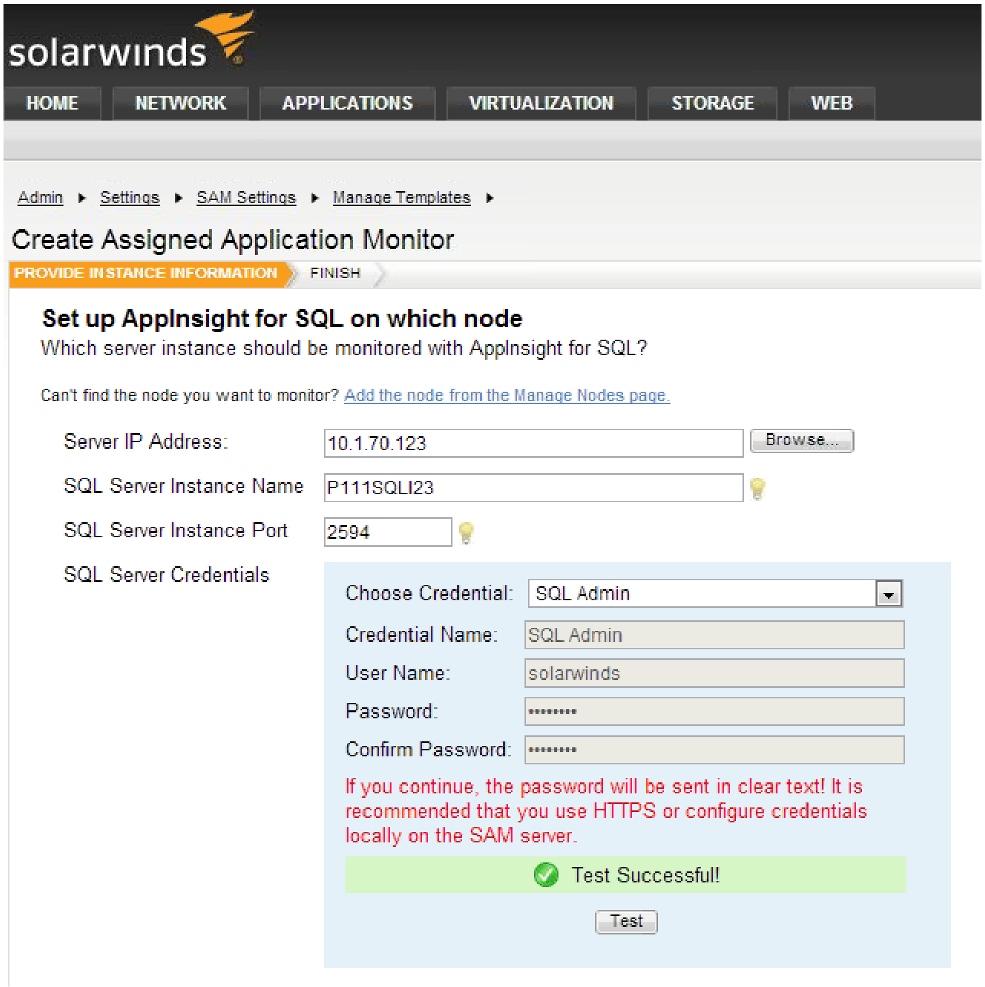


```
Pinging p111sqlv18.st.com [10.1.70.148] with 32 bytes of data:
Reply from 10.1.70.148: bytes=32 time=13ms TTL=124
Reply from 10.1.70.148: bytes=32 time=14ms TTL=124
Reply from 10.1.70.148: bytes=32 time=14ms TTL=124
Reply from 10.1.70.148: bytes=32 time=14ms TTL=124

Ping statistics for 10.1.70.148:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 13ms, Maximum = 14ms, Average = 13ms

C:\Users\rthomas.ST>ping P111SQLV23

Pinging p111sqlv23.st.com [10.1.70.123] with 32 bytes of data:
Reply from 10.1.70.123: bytes=32 time=13ms TTL=124
Reply from 10.1.70.123: bytes=32 time=13ms TTL=124
Reply from 10.1.70.123: bytes=32 time=14ms TTL=124
Reply from 10.1.70.123: bytes=32 time=13ms TTL=124

Ping statistics for 10.1.70.123:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 13ms, Maximum = 14ms, Average = 13ms

C:\Users\rthomas.ST>_
```

IP address

4. Make sure that the node with this address is being monitored by SAM. If it is not, add it.
   Note: If the IP address represents a cluster, the node name of the active cluster member will be populated. It is recommended that you change the node name to something more easily understood, for example:

“`P111SQLV23 (cluster)`” during the final step of the Add Node Wizard.


solarwinds
HOME
NETWORK
APPLICATIONS
VIRTUALIZATION
STORAGE
WEB
Admin ▸ Settings ▸ SAM Settings ▸ Manage Templates ▸
PROVIDE INSTANCE INFORMATION
FINISH
Browse...
Test

Create Assigned Application Monitor

Set up AppInsight for SQL on which node

Which server instance should be monitored with AppInsight for SQL?

Can't find the node you want to monitor? Add the node from the Manage Nodes page.

Server IP Address: 10.1.70.123

SQL Server Instance Name: P111SQLI23

SQL Server Instance Port: 2594

SQL Server Credentials

Choose Credential: SQL Admin

Credential Name: SQL Admin

User Name: solarwinds

Password: ••••••••

Confirm Password: ••••••••

If you continue, the password will be sent in clear text! It is recommended that you use HTTPS or configure credentials locally on the SAM server.

✓ Test Successful!

5. Enter the proper information and manually assign the *Appinsight for SQL* template to the node with the appropriate IP address.
6. The application should begin polling.

AppInsight for SQL Alerts

The following two alerts are included with AppInsight for SQL:

- **Alert me when my database file disk I/O latency is high.**
- **Alert me when a node warranty expires in 30 days.**

AppInsight for SQL Details View (Summary Tab)

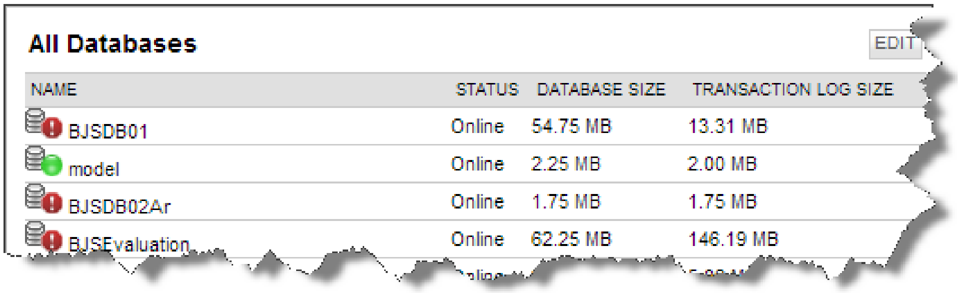
The Summary tab provides a great deal of information about the current SQL instance. The available resources unique to this view include:

- All Databases
- Application Availability
- Application Details
- Buffer Manager
- Cache
- Connections
- Disk
- Latches and Locks
- Memory
- Pages
- Sessions
- SQL Agent Job Status
- SQL Errors
- SQL Event Log Message Details
- SQL Server
- Top 10 Databases by Active User Connections
- Top 10 Most Expensive Queries by CPU Time

Note: Chart resources require several values for lines on charts to be visible.

All Databases

This resource shows the status of all monitored databases on the current SQL instance. The data can be sorted by clicking the headers of each column.



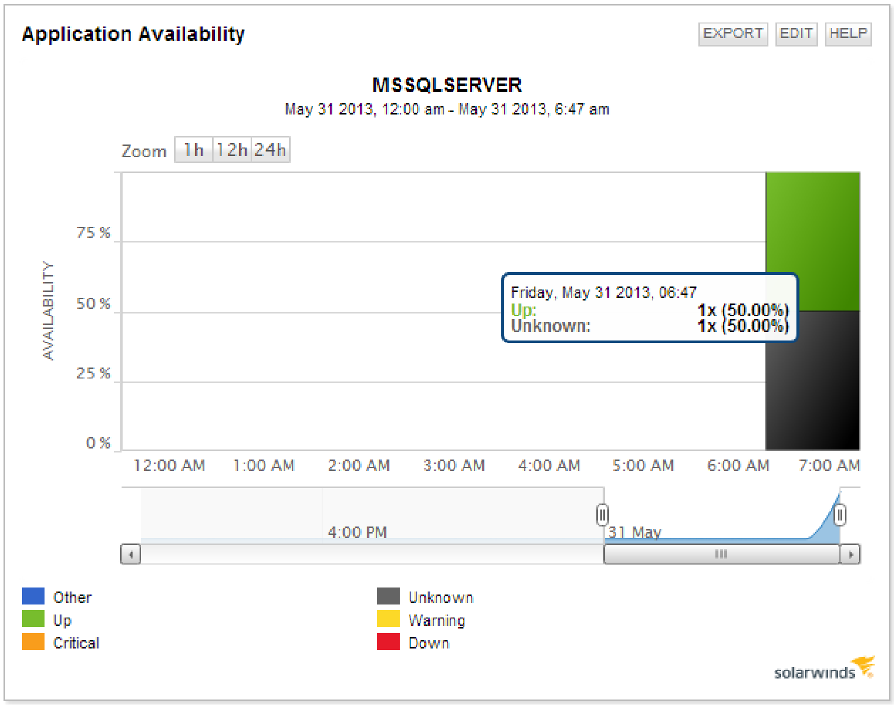
NAME	STATUS	DATABASE SIZE	TRANSACTION LOG SIZE
BJSDB01	Online	54.75 MB	13.31 MB
model	Online	2.25 MB	2.00 MB
BJSDB02Ar	Online	1.75 MB	1.75 MB
BJSEvaluation	Online	62.25 MB	146.19 MB

- **Name:** Displays the name of the listed database, in addition to the current performance status as indicated by the icon. Clicking the name of the database will take you to the *Database Details* page.
- **Status:** Displays the *Online* or *Offline* status of the listed database.
- **Database Size:** Displays the current size of the listed database.

- **Transaction Log Size:** Displays the size of the transaction log associated with the listed database.

Application Availability

This resource graphically shows the various states of an application in a percentage format and is made up of three sections: the legend, the main chart, and the lower chart. The colors of the main chart correspond to the labels in the legend.



Note: This resource will not be available when using Internet Explorer 7. Consider upgrading Internet Explorer or using FireFox or Chrome as your web browser.

Zooming

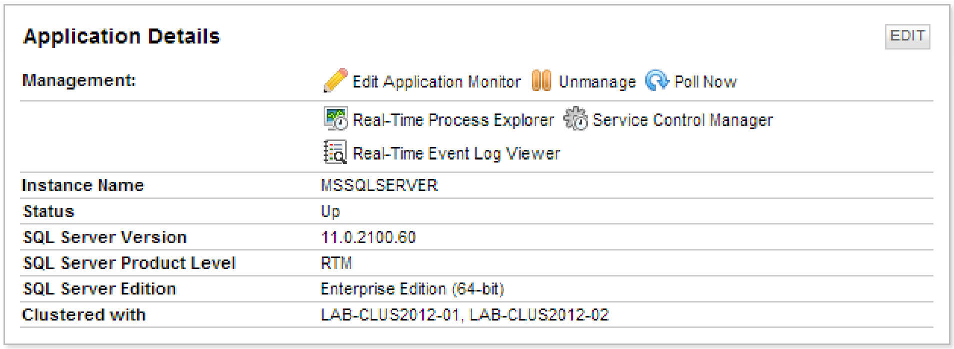
You can have the chart show a pre-determined, hourly time period of data by clicking on any one of the three Zoom buttons; **1h**, **12h**, or **24h**. Alternatively, you can have the chart show a specific time range by dragging the mouse over a small area of the chart.

Once you have a time period selected, the lower chart will zoom to the starting and ending values of the selected time period. With the lower chart you can zoom in further by fine tuning the view with the sliders. The main view of the chart will display the selected time period between the two sliders of the lower chart.

Moving the mouse over the main chart will reveal a tooltip with more detailed information about the specific time period the mouse is hovering over. By default, all statistics are shown in the main chart.

Application Details

This resource provides tools and details concerning the current application.



Application Details

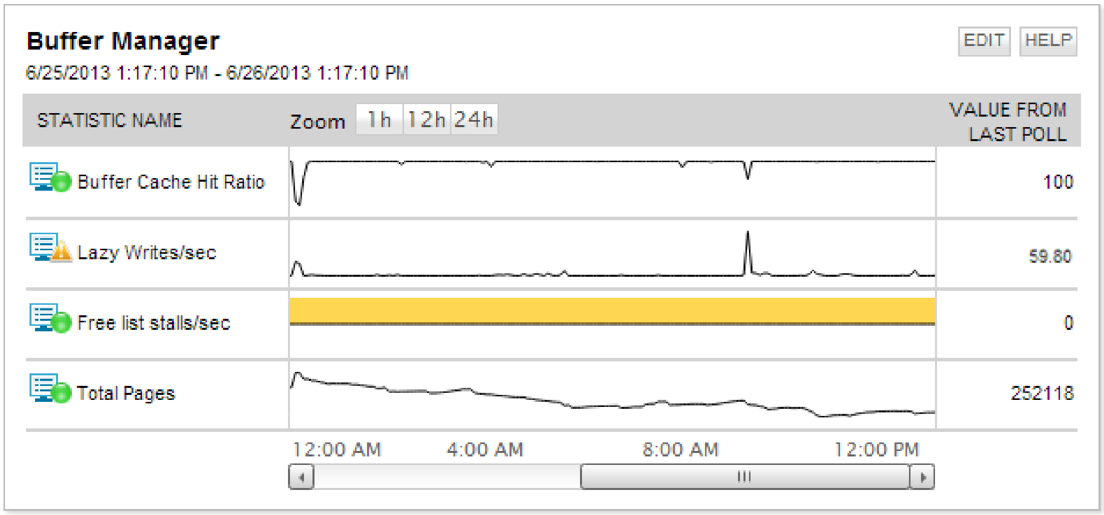
Management:	Edit Application Monitor, Unmanage, Poll Now Real-Time Process Explorer, Service Control Manager Real-Time Event Log Viewer
Instance Name	MSSQLSERVER
Status	Up
SQL Server Version	11.0.2100.60
SQL Server Product Level	RTM
SQL Server Edition	Enterprise Edition (64-bit)
Clustered with	LAB-CLUS2012-01, LAB-CLUS2012-02

- **Management:** This provides tools allowing you to more easily manage and troubleshoot the current application.
- **Instance Name:** Displays the name of the current SQL instance.
- **Status:** Displays the status of the current SQL instance.
- **SQL Server Version:** Displays the version of the current SQL server.
- **SQL Server Product Level:** Displays the maturity level of the current SQL version.
- **SQL Server Edition:** Displays the edition of the current SQL Server.
- **Clustered with:** If the listed SQL Server is in a cluster, this field will show a listing of the servers that make up the cluster. If the cluster is managed by SAM, a status icon will become visible. If the cluster is not managed in SAM, no status icon will be visible.

Buffer Manager

This resource contains monitors specific to the current status of buffers. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Buffer Manager

6/25/2013 1:17:10 PM - 6/26/2013 1:17:10 PM

STATISTIC NAME	Zoom 1h 12h 24h	VALUE FROM LAST POLL
Buffer Cache Hit Ratio		100
Lazy Writes/sec		59.80
Free list stalls/sec		0
Total Pages		252118

Total pages	**Definition:** This performance counter returns the number of pages in the buffer pool **Information:** The returned value includes database, free, and stolen pages. Note: This counter is not available in SQL 2012.
Buffer Cache Hit Ratio	**Definition:** Indicates the percentage of pages found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups over the last few thousand page accesses. **Information:** After a long period of time, the ratio moves very little. Because reading from the cache is much less expensive than reading from disk, you want this ratio to be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory

	available to SQL Server. The higher this ratio, the less often SQL Server has to go to the hard disk to fetch data, and performance is boosted overall. This counter averages the Buffer Cache Hit Ratio from the time the last instance of SQL Server was restarted. This counter is not a real-time measurement, but an average of all the days since SQL Server was last restarted. In OLAP applications, the ratio could be much lower because of the nature of how OLAP works. **Possible problems:** If the returned value is low, this could indicate that your SQL server may not have enough memory to function at peak performance. **Remediation:** Check your SQL server and verify its memory is being used efficiently. Applications other than SQL may be using a great deal of memory. Try and recover memory by closing unnecessary applications. Installing additional memory may also help.
Lazy Writes/Sec	**Definition:** The lazy writer is a system process that flushes out buffers that contain changes that must be written back to disk before the buffer can be reused for a different page and makes them available to user processes. **Information:** This counter tracks how many times per second that the Lazy Writer process is moving dirty pages from the buffer to disk in order to free up buffer space. The Lazy Writer eliminates the need to perform frequent checkpoints in order to create available buffers. Generally speaking, this should not be a high value, say more than 20 per second. Ideally, it should be close to zero. If it is zero, this indicates that your SQL Server's buffer cache is large and your SQL Server does not need to free up dirty pages. **Possible problems:**

	If the returned value is high, this can indicate that your SQL Server's buffer cache is small and that your SQL Server needs to free up dirty pages. **Remediation:** Check your SQL server and verify its memory is being used efficiently. Applications other than SQL may be using a great deal of memory. Try and recover memory by closing unnecessary applications. Installing additional memory may also help.
Free List Stalls/Sec	**Definition:** Indicates the number of requests per second that had to wait for a free page. **Information:** This displays the frequency with which requests for available database pages are suspended because no buffers are available. **Possible problems:** If the returned value is high, this indicates that not enough memory is available for the SQL Server. **Remediation:** Check your SQL server and verify its memory is being used efficiently. Applications other than SQL may be using a great deal of memory. Try and recover memory by closing unnecessary applications. Installing additional memory may also help.

Zooming

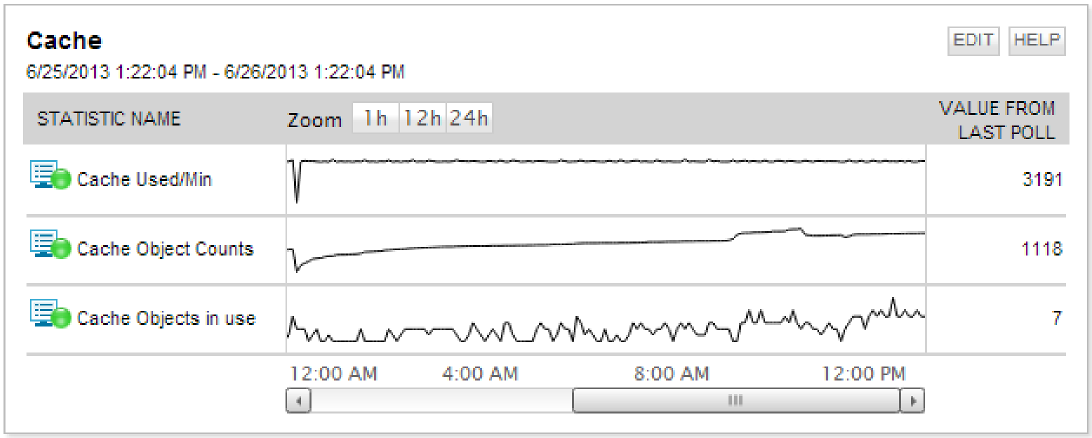
You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Cache

This resource contains monitors specific to the current status of the cache. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Cache

6/25/2013 1:22:04 PM - 6/26/2013 1:22:04 PM

STATISTIC NAME	VALUE FROM LAST POLL
Cache Used/Min	3191
Cache Object Counts	1118
Cache Objects in use	7

Cache Used/Min	**Definition:** This displays the number of times the cache object has been looked up. **Information:** This is not incremented when parameterized queries find a plan in the cache. However, this can be incremented multiple times when using Showplan. Good plan reuse is one sign of a healthy system. Compiling a query plan can be a CPU-intensive operation, especially with complex queries, so reuse is encouraged.

	Possible problems: If the returned value is low, this can indicate ineffective reusing of plan cache which could cause a CPU spike. Conversely, if the returned value is high for a large number of plans, this could indicate that your cache space is being taken up with plans that were compiled and used once to execute an ad hoc query, then never used again. **Remediation:** A database setting that can be helpful to plan reuse is forced parameterizaton. (You can set this using the following command: ALTER DATABASE <databaseName> SET PARAMETERIZATION FORCED). Normally, the query optimizer is very conservative when deciding what queries can be parameterized, but this setting makes the optimizer be more liberal in choosing parameters.
Cache Object Counts	**Definition:** This performance counter returns the number of cache objects in the cache. **Information:** The Plan Cache object provides counters to monitor how SQL Server uses memory to store objects such as stored procedures, ad hoc and prepared Transact-SQL statements, and triggers. Multiple instances of the Plan Cache object can be monitored at the same time, with each instance representing a different type of plan to monitor. **Possible problems:** High numbers of total cached objects use portions of the physical memory available to a SQL instance on a per database basis. This can result in one database cache impacting the performance of other local databases due to memory contention. **Remediation:** Increase the memory available to SQL services, reduce the number of databases on this instance of SQL, or examine the volume of ad hoc queries running against the server.

Cache Objects in Use	**Definition:** This performance counter returns number of cache objects in use. **Information:** The Plan Cache object provides counters to monitor how SQL Server uses memory to store objects such as stored procedures, ad hoc and prepared Transact-SQL statements, and triggers. Multiple instances of the Plan Cache object can be monitored at the same time, with each instance representing a different type of plan to monitor. **Possible problems:** High numbers of cached objects in use consume the memory available to a SQL server at a higher rate than non-active objects on a per database basis. This can result in one database cache impacting the performance of other local databases due to memory contention. **Remediation:** Increase the memory available to SQL server, reduce the number of active objects, consolidate stored procedures, consolidate and convert ad hoc queries to stored procedures where possible, or reduce the number of databases on the server.

Zooming

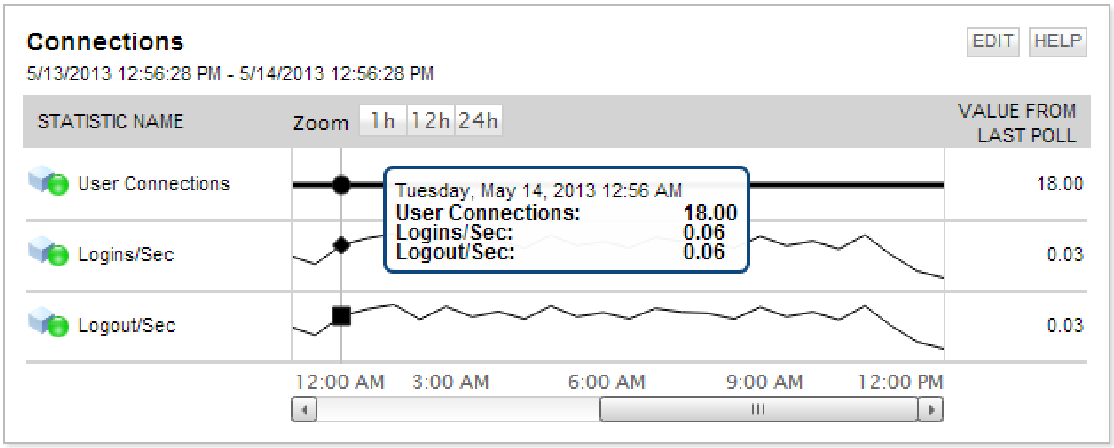
You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Connections

This resource contains monitors specific to the current status of the active connections. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



5/13/2013 12:56:28 PM - 5/14/2013 12:56:28 PM

STATISTIC NAME	VALUE FROM LAST POLL
User Connections	18.00
Logins/Sec	0.03
Logout/Sec	0.03

Tuesday, May 14, 2013 12:56 AM
User Connections: 18.00
Logins/Sec: 0.06
Logout/Sec: 0.06

User Connections	**Definition:** The User Connections performance counter identifies the number of different users that are connected to your SQL Server at the time the sample was taken. **Information:** You need to watch this counter over time to understand your baseline user connection numbers. Since each user connection consumes some memory space, a high number of user connections can impact throughput and cause a performance slow-down. Once you have an idea of your high and low thresholds during normal usage of your system, you can then look for times when this counter exceeds these high and low marks. **Possible problems:**

	If the returned value of this counter goes down and the load on the system remains stable, you might have a bottleneck that is not allowing your server to handle the normal load. Keep in mind that this counter value might go down just because less people are using your SQL Server instance. If you see this number jump by 500% from your baseline, you may be seeing a slowdown of your server activity. **Remediation:** You may want to boost the SQL Server configuration setting, Maximum Worker Threads to a figure higher than the default setting of 255. The setting for Maximum Worker Threads should be higher than the maximum number of user connections your server ever reaches.
Logins/Sec	**Definition:** This performance counter returns the total number of logins started, per second, and does not include pooled connections. **Information:** Opening and closing connections is an expensive process. A pooled connection is one which is kept open by an application for other requests to re-use. **Possible problems:** If the returned value is high, this can indicate that the application is not correctly using connection pooling. **Remediation:** Review the Connection Polling configuration.
Logout/Sec	**Definition:** This performance counter returns the total number of logout operations started, per second. **Information:** Opening and closing connections is an expensive process. When applications do not use a connection pool, each request needs to establish its own connection before the query can be

	executed. It then has to close it. A pooled connection is one which is kept open by an application for other requests to re-use. **Possible problems:** If the returned value is high, this can indicate that the application is not correctly using connection pooling. **Remediation:** Review the Connection Polling configuration.

Zooming

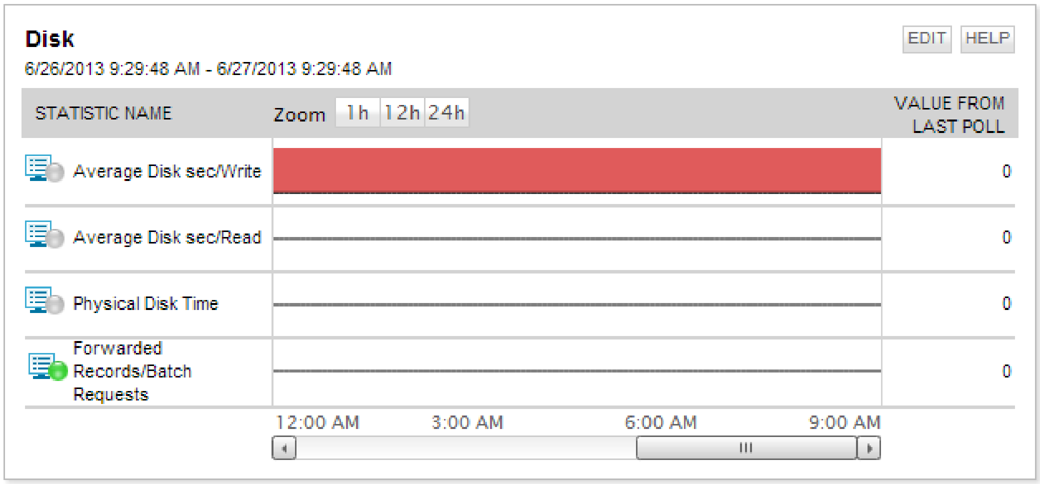
You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Disk

This resource contains monitors specific to the current status of the disk. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Disk

6/26/2013 9:29:48 AM - 6/27/2013 9:29:48 AM

STATISTIC NAME	VALUE FROM LAST POLL
Average Disk sec/Write	0
Average Disk sec/Read	0
Physical Disk Time	0
Forwarded Records/Batch Requests	0

Average Disk Sec/Read	**Definition:** This performance counter shows the average time, in seconds, of a read of data from the disk. **Information:** The returned value indicates the average time of read data from the disk. 4-8ms is ideal. The returned value is considered acceptable up to 20ms. Any higher value needs further investigation. **Possible problems:** If a value greater than 15-20ms is reported, this may indicate disk bottlenecks. **Remediation:** Increase the number of hard disks.
Average Disk Sec/Write	**Definition:** This performance counter shows the average time, in seconds, of a write of data to the disk. **Information:** The returned value indicates the average time of write data from the disk. 4-8ms is ideal. The returned value is considered acceptable up to 20ms. Any higher value needs

	further investigation. **Possible problems:** Values greater than 15-20ms nay indicate disk bottlenecks. **Remediation:** Increase the number of hard disks.
Forwarded Records/Batch Requests	**Definition:** This performance counter identifies the use of a pointer which has been created when variable length columns have caused a row to move to a new page in a heap. **Possible problems:** Rows with Varchar columns can experience expansion when Varchar values are updated with a longer string. In the case where the row cannot fit in the existing page, the row migrates and access to the row will traverse a pointer. This only happens on heaps (tables without clustered indexes). **Remediation:** Evaluate clustered indexes for heap tables. In cases where clustered indexes cannot be used, drop non-clustered indexes, build a clustered index to Reorg pages and rows, drop the clustered index, then recreate non-clustered indexes.
Forwarded Records/Sec	**Definition:** This performance counter returns the number of records fetched through forwarded record pointers. **Information:** Tables without a clustered index. If you start with a short row, and update the row creating a wider row, the row might not fit on the data page. A pointer is put in its location and the row is forwarded to another page. **Possible problems:** Look at the code to determine where the short row is inserted

	followed by an update. **Remediation:** Can be avoided by: 1. Using default values so that an update does not result in a longer row that is the root cause of forwarded records. 2. Use Char instead of Varchar. This fixes the length so that an update does not result in a longer row.
Physical Disk Time	**Definition:** This performance counter returns the ratio of elapsed time when the disk drive was busy with read or write requests. **Information:** This performance counter is deceptive because it makes no accommodation for multiple spindles. Thus, the more spindles (i.e. physical hard disks) you have, the higher the percentile values can go. Conversely, if these spindles are shared across LUNs or other services, you may have high numbers on this counter without any correlation to SQL Server activity. The value for this counter should be below 50%. **Possible problems:** If this performance counter sustains an average above 70%, you may have contention with your drive or RAM. **Remediation:** You should increase number of hard drives used by SQL server.

Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Latches and Locks

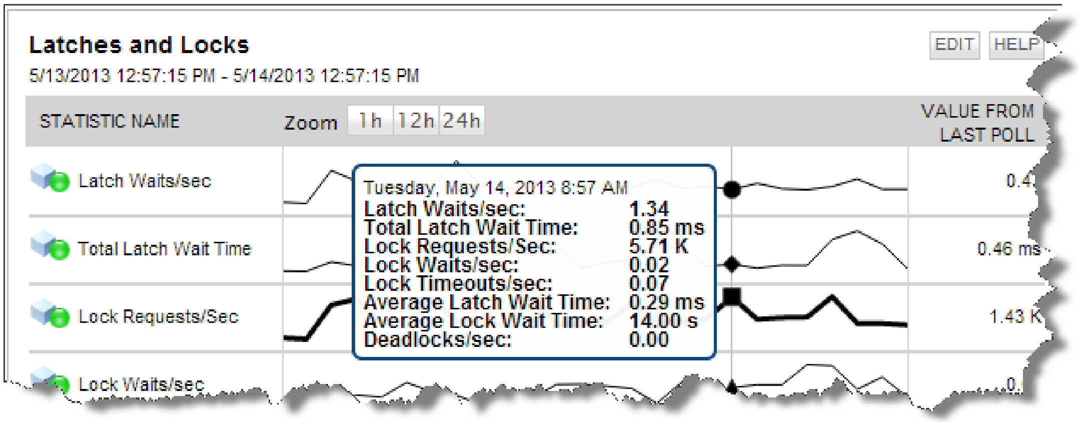
This resource measures the time it takes for latches and locks to perform various operations.

Latches perform the task of thread synchronization. For example, if a thread is reading a page from disk and creating a memory structure to contain it, it will create one or more Latches to prevent corruption of these structures. Once the operation is complete, the Latches will be released and other threads will be able to access that page and memory structure again. For the most part, latches are transient, taking a few milliseconds to complete.

A Lock prevents different users from overwriting each other's changes. Generally speaking, a lock will last for the duration of the transaction.

The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name c*olumn will take you to the *Performance Counter Details page* for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Latches and Locks

5/13/2013 12:57:15 PM - 5/14/2013 12:57:15 PM

STATISTIC NAME	VALUE FROM LAST POLL
Latch Waits/sec	0.4
Total Latch Wait Time	0.46 ms
Lock Requests/Sec	1.43 K
Lock Waits/sec	0.

Tuesday, May 14, 2013 8:57 AM

Latch Waits/sec:	1.34
Total Latch Wait Time:	0.85 ms
Lock Requests/Sec:	5.71 K
Lock Waits/sec:	0.02
Lock Timeouts/sec:	0.07
Average Latch Wait Time:	0.29 ms
Average Lock Wait Time:	14.00 s
Deadlocks/sec:	0.00

Latch Waits/Sec	**Definition:** This performance counter displays the number of latch requests that could not be granted immediately.
Total Latch Wait Time/Latch Waits	**Definition:** This performance counter returns the ratio of Total Latch Wait Time, in milliseconds, for latch requests in the last second to amount of latches, in a one second period that had to wait. Latches are lightweight means of holding a very transient server resource, such as an address in memory. **Information:** This ratio should be less than 10. Monitoring the latches to determine user activity and resource usage can help you to identify performance bottlenecks. **Possible problems:** If the returned value is high, it is likely there is an I/O or memory bottleneck. **Remediation:** Check your SQL server and verify its memory is being used efficiently. Applications other than SQL may be using a great deal of memory. Try and recover memory by closing unnecessary applications. Installing additional memory may also help.

Lock Requests/Sec	**Definition:** This performance counter returns the number of requests for a type of lock, per second. **Possible problems:** If the returned value is high, this can indicate that the queries are accessing large numbers of rows. If you notice a high Average Wait time, then this could be an indication of blocking. **Information:** Blocking is an unavoidable characteristic of any relational database management system (RDBMS) with lock-based concurrency. On SQL Server, blocking occurs when one Server Process ID (SPID) holds a lock on a specific resource and a second SPID attempts to acquire a conflicting lock type on the same resource. Typically, the time frame for which the first SPID locks the resource is very small. When it releases the lock, the second connection is free to acquire its own lock on the resource and continue processing. This is normal behavior and may happen many times throughout the course of a day with no noticeable effect on system performance. **Remediation:** High Read queries should be reviewed. The simple way to fix locking is just kill the connection that is causing the lock. Fixing locking, blocking, and deadlocking issues is often a matter of redesign. You should examine either the schema of your database and/or the way the application is designed. One of the most efficient ways to deal with locking is to ensure that the design of your system does not have to deal with excessive locks. The best way to do this is to normalize your tables, using more atomic structures that allow the query to get in and get out faster. Another method is to code with the locks in mind. If the design requires less normalization, you will have to evaluate the programs to ensure that they have the right isolation levels.

Lock Waits/Sec	**Definition:** This performance counter reports the number of times users waited to acquire a lock over the past second. **Information:** Note that this counter only gets incremented only when you “wake up” after waiting on the lock. **Possible problems:** Non-zero values indicate that there is at least some level of blocking occurring. If you combine this with the Lock Wait Time counter, you can get some idea of how long the blocking lasted. A zero value for this counter can definitively rule out blocking as a potential cause. **Remediation:** High Read queries should be reviewed.
Lock Timeouts/Sec	**Definition:** This performance counter returns the number of lock requests per second that have timed out, including internal requests for NoWait locks. **Information (timeout_period):** This is the number of milliseconds that will pass before Microsoft SQL Server returns a locking error. A value of -1 (default) indicates no time-out period (that is, wait forever). When a wait for a lock exceeds the time-out value, an error is returned. A value of 0 means to not wait at all and return a message as soon as a lock is encountered. **Possible problems:** If you see a value above 0 for this counter, your users will experience problems as their queries are not completing. **Remediation:** You should review your queries to determine which queries are causing this situation.

Lock Wait Time	**Definition:** This performance counter displays the total time spent waiting across all transactions, in milliseconds, to acquire a lock in the last second. **Information:** Though this counts how many milliseconds SQL Server is waiting on locks during the last second, this counter actually starts recording at the end of locking event. Peaks most likely represent one large locking event. **Possible problems:** If the returned value is greater than 60 seconds (60,000ms) then there may be extended blocking which could be an issue. **Remediation:** Thoroughly analyze the blocking script output. Some applications are written for timing out after 60 seconds. Because SQL Server records a lock at the end of a locking event, remember that an application with large transactions may have inflated lock wait times while still performing as expected. For example, an application that issues multi-million record updates might have very long lock wait times while performing exactly as it was designed.
Average Latch Wait Time	**Definition:** This performance counter reports the average latch wait time, in milliseconds, for latch requests that had to wait. **Information:** SQL Server two lightweight mutual exclusion solutions—Latches and Spinlocks—which are less robust than locks but are less resource intensive. The value of this counter should generally correlate to Latch Waits/sec and move up or down with it accordingly. **Possible problems:** If you see this number jump above 300, you may have

	contention for your server's resources. High values for this counter could potentially block other users. **Remediation:** You should examine tasks currently waiting using the following command: sys.dm_os_waiting_tasks DMV.
Average Lock Wait Time	**Definition:** This performance counter reports the average amount of Wait time, in milliseconds, for each lock request that resulted in a wait. **Information:** Generally speaking, the lower the value, the better. This value should correlate to the Lock Waits/sec counter and move up or down with it accordingly. **Possible problems:** An average wait time longer than 500ms may indicate excessive blocking. **Remediation:** You should determine which queries are generating locks to identify where the blocking is originating.
Total Latch Wait Time	**Definition:** Total latch wait time (in milliseconds) for latch requests in the last second. **Information:** Short term light weight synchronization object. Latches are not held for the duration of a transaction. Typical latching operations during row transfers to memory, controlling modifications to row offset table, and so on. **Possible problems:** If high, check Perfmon Disk and Memory objects for: • I/O bottlenecks • Memory pressure

	Remediation: This can be mitigated by adding more memory or I/O capacity.

Zooming:

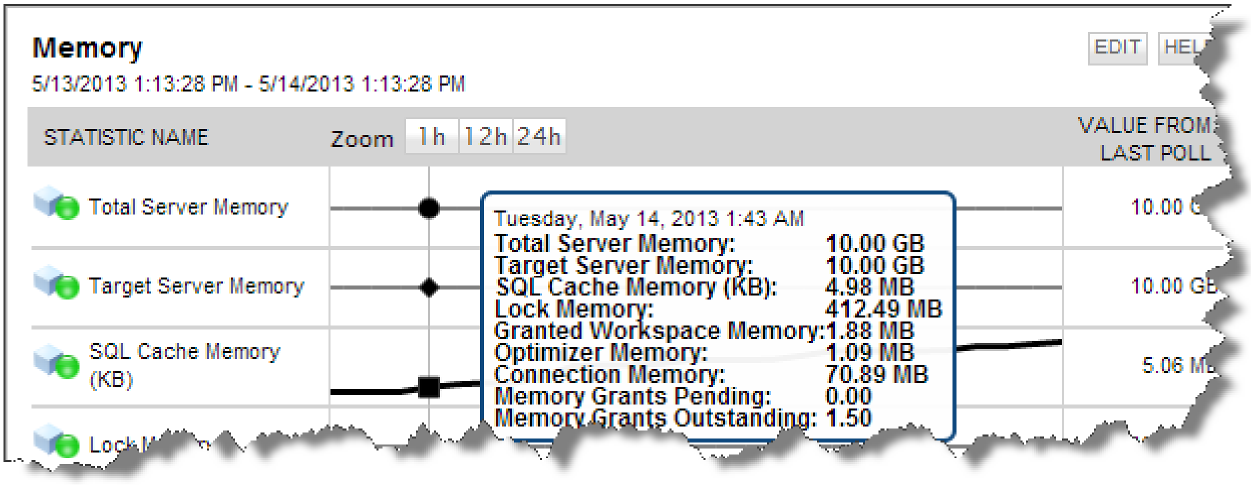
You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Memory

This resource displays the current status of SQL Server memory. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Memory

5/13/2013 1:13:28 PM - 5/14/2013 1:13:28 PM

STATISTIC NAME	VALUE FROM LAST POLL
Total Server Memory	10.00 G
Target Server Memory	10.00 GB
SQL Cache Memory (KB)	5.06 MB
Lock M	

Tuesday, May 14, 2013 1:43 AM

Total Server Memory:	10.00 GB
Target Server Memory:	10.00 GB
SQL Cache Memory (KB):	4.98 MB
Lock Memory:	412.49 MB
Granted Workspace Memory:	1.88 MB
Optimizer Memory:	1.09 MB
Connection Memory:	70.89 MB
Memory Grants Pending:	0.00
Memory Grants Outstanding:	1.50

Total Server Memory	**Definition:** This performance counter measures the current amount of memory that SQL Server is using. **Information:** If the value of this counter continues to grow larger, the server has not yet reached its steady state and is still trying to populate the cache and get pages loaded into memory. Performance will likely be somewhat slower if this value continually grows larger since more disk I/O is required. This behavior is normal. Eventually Total Server Memory should approximate Target Server Memory. **Possible problems:** If the Total Server Memory counter is greater than or equal to the Target Server Memory counter, this can indicate that your SQL Server may be under memory pressure. **Remediation:** Installing additional memory into your SQL server should resolve the problem.
Target Server Memory	**Definition:** This monitor measures the total amount of dynamic memory the server can consume.

	Information: This performance counter tells you how much memory SQL Server would like to use to operate efficiently. Compare with Total Server Memory. **Possible problems:** If the Total Server Memory counter is greater than or equal to the Target Server Memory counter, this could indicate that your SQL Server may be under memory pressure. **Remediation:** Installing additional memory into your SQL server should resolve the problem.
Target - Total Server Memory	**Definition:** This performance counter shows the difference between the total amount of dynamic memory the server can consume and the current amount of memory that SQL Server is using. **Possible problems:** If this performance counter is greater than or equal to the Target Server Memory performance counter, this indicates that SQL Server may be under memory pressure. **Remediation:** Installing additional memory into SQL server should resolve the problem.
SQL Cache Memory (KB)	**Definition:** This performance counter measures the total amount of dynamic memory the server is using for the Dynamic SQL cache. **Information:** Most memory used by SQL Server is allocated to the Buffer Pool, which is used to store data pages. SQL Server steals a proportion of this memory for use in caching query plans. The overall amount of memory available to SQL Server depends

	upon the amount of memory installed on the server, the architecture of the server, the version and edition of SQL Server and the amount of memory pressure being experienced by SQL Server. This pressure can be internal (SQL Server resources need memory) or external (operating system needs memory). SQL Server is designed to respond to memory pressure when necessary. **Possible problems:** Memory contention with the buffer pool. **Remediation:** Increase memory available to SQL server.
Lock Memory	**Definition:** This performance counter returns the total amount of dynamic memory the server is using for locks. **Information:** Lock pages in memory is used to prevent older versions of Windows and SQL from allowing Windows operating system page out of the buffer pool. **Possible problems:** Lock pages in memory determines which accounts can use a process to keep data in physical memory, which prevents the system from paging the data to virtual memory on disk. Exercising this privilege could significantly affect system performance by decreasing the amount of available random access memory (RAM). **Remediation:** Upgrade to Windows 2008 R2 or greater and SQL 2008 or greater.
Granted Workspace Memory	**Definition:** This performance counter returns the total amount of memory currently granted to executing processes, such as Hash, Sort, Bulk Copy, and Index creation operations.

	Information: This performance counter tells you how much memory has currently been granted to running queries. If there is memory pressure because of workspace memory, this value should be at least 25% of the virtual memory available to SQL Server. **Possible problems:** If the memory pressure is severe, the server might return errors such as 701 or 8645. **Remediation:** If this is the case, this might be a good reason to consider using SQL Server 64-bit.
Optimizer Memory	**Definition:** This performance counter returns the total amount of dynamic memory the server is using for query optimization. **Information:** There are no thresholds associated with this performance counter.
Connection Memory	**Definition:** This monitor returns the total amount of dynamic memory the server is using for maintaining connections. **Information:** SQL Server sets aside three packet buffers for every connection made from a client. Each buffer is sized according to the default network packet size specified by the sp_configure stored procedure. If the default network packet size is less than 8KB, the memory for these packets comes from SQL Server's buffer pool. If it is 8KB or larger, the memory is allocated from SQL Server's MemToLeave region. It is worth noting that the default network packet size for the .NET Framework Data Provider for SQL Server is 8KB, so the buffers associated with managed code client connections typically come from SQL Server's

	MemToLeave region. This contrasts with classic ADO applications, where the default packet size is 4KB, and the buffers are allocated form the SQL Server buffer pool.
Memory Grants Pending	**Definition:** This monitor returns the total number of processes waiting for a workspace memory grant. **Information:** Memory resources are required for each user request. If sufficient memory is not available, the user waits until there is adequate memory for the query to run. **Possible problems:** Returned values greater than zero for a sustained period of time is a very strong indicator of memory pressure. **Remediation:** You should first examine the database design, queries, and indexes to ensure the system is properly tuned before installing additional RAM. There may be query inefficiencies in the instance that is causing excessive memory grant requirements. For example, large Sorts or Hashes that can be resolved by tuning the indexing or queries being executed. Compare with Memory Grants Outstanding. If the number of pending grants increases, try the following: • Add more memory to SQL Server • Add more physical memory to the server. • Check for memory pressure. See and correct indexing if you experience “Out of memory” conditions.
Memory Grants Outstanding	**Definition:** This performance counter returns the total number of processes that have successfully acquired a workspace memory grant. **Information:** Look at Memory Grants Outstanding and Memory Grants Pending. If you see a long queue of pending grants as

	compared to outstanding grants, there is likely memory pressure because of query workspace memory. You can confirm this by checking the Granted Workspace Memory (KB) performance counter that tells you how much memory has currently been granted to running queries. **Possible problems:** A returned value that is high can indicate peak user activity. If there is memory pressure because of workspace memory, this value should be at least 25% of the virtual memory available to SQL Server. If the memory pressure is severe, the server might even return errors such as 701 or 8645. **Remediation:** If severe, and using 32-bit, consider using SQL Server 64-bit. See Memory Grants Pending.
Pages/Sec	**Definition:** This performance counter displays the rate at which pages are read from or written to disk to resolve hard page faults. **Information:** This is a primary indicator of the kinds of faults that cause system-wide delays. This should be close to zero on a dedicated SQL Server. You will see spikes during backups and restores, but this is normal. **Possible problems:** High values causes hard page faults, which can cause SQL Server to use the page, as opposed to RAM. **Remediation:** You may want to add additional RAM to stop the paging.
Cache Hit Ratio	**Definition:** This metric is the ratio between Cache Hits and Lookups. Cache Hit Ratio measures how much the plan cache is being used.

	Information: A high percentage here means that your SQL Server is not building a new plan for every query it is executing and is working effectively and efficiently. A low percentage here means that, for some reason, the SQL Server is doing more work than it needs to. This metric needs to be considered alongside the plan cache reuse metric which looks at the spread of plan reuse through your cache. Plan cache is memory used to store objects such as stored procedures, ad hoc and prepared Transact-SQL statements, and triggers. The plan cache is the component of SQL memory that stores query plans for re-use. When a query is issued against SQL, the optimizer will attempt to re-use a cached plan if the traits of the query permit - but can only do so if that plan resides in cache, otherwise it needs to compile a new plan. This is not to say that every plan that is cached is re-used. Changes in schema, a query running parallel that may have run serially before, or changes in parameters may require a new plan to be compiled even if a similar plan exists in cache. Plan compilations are expensive though. Ideally this counter should be near 100%. **Possible problems:** The value of this counter should never fall below 90%. Low cache hit ratio (<20%) along with a sustained query execution rate (Batch Requests/sec) indicates that compiled plans are not being re-used. It should be noted that the hit ratio counter may be skewed by internal lookups performed. **Remediation:** The amount of caching should be reduced by examining the workload to see if queries have been parameterized, or can be rewritten with stored procedures.
Available MBytes	**Definition:** This is the amount of available physical memory on the server. **Information:**

	An acceptable output for this may vary widely based on how much physical memory is in the machine. If you have 2GB of RAM installed on the machine, it is common to see SQL Server use 1.7GB of RAM. If no other processes are running on your SQL Server, ensure you have at least 80MB available for Windows at any given time. **Possible problems:** Low values show that SQL server has lack of memory. **Remediation:** Install additional memory.
Page Usage	**Definition:** This shows the percentage of the page file that is being utilized. **Information:** A Page File is simply a file on the hard drive that handles situations where the system wants to move or “page out” sections of memory. There are several situations that cause this, but the one you should be most concerned about is when the system is out of physical memory. **Possible problems:** Values greater than 70 percent indicate paging and lack of memory. If the system runs out of memory it can “borrow” some storage from the hard drive to release some memory until it needs that data again. The reason that this is bad is because hard drives are amazingly slow in comparison with solid-state memory access. Using the page file slows SQL Server a great deal. **Remediation:** Install additional memory.
Workfiles Created/Sec	**Definition:** This is the number of work files created per second. **Information:**

	For example, work files could be used to store temporary results for hash joins and hash aggregates. The returned value should be less than 20. Tempdb work files are used in processing hash operations when the amount of data being processed is too large to fit into the available memory. **Possible problems:** High values can indicate thrash in the tempdb file as well as poorly coded queries. **Remediation:** It is possible to reduce the value this monitor returns by making queries more efficient by adding/changing indexes. Adding additional memory will also help.
Worktables Created/sec	**Definition:** This performance counter displays the number of work tables created per second. **Information:** For example, work tables could be used to store temporary results for query spool, lob variables, XML variables, and cursors. The returned value should be less than 20. Worktables are used for queries that use various spools (table spool, index spool, and so on). **Possible problems:** High values could cause general slowdown. **Remediation:** Remediation requires rewriting your procedures.

Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

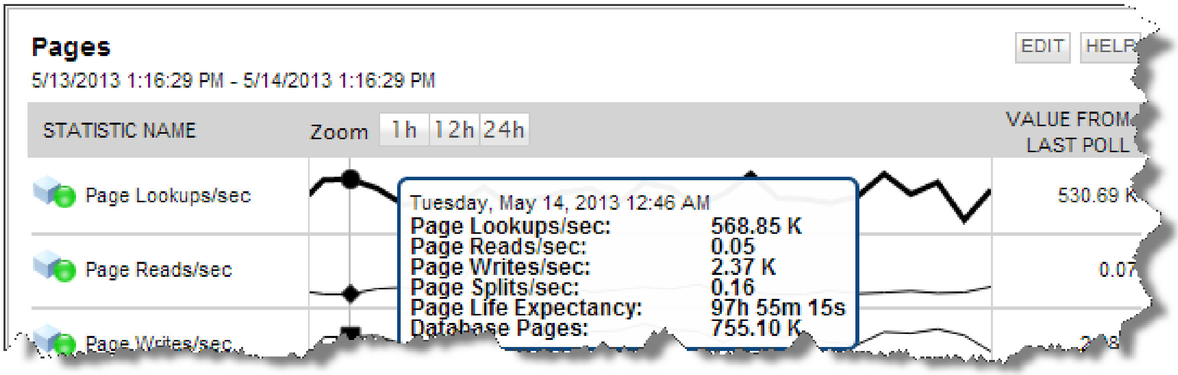
Pages

This resource shows page status of the SQL server.

In SQL Server, the page size is 8 KB. Therefore, SQL Server databases have 128 pages per MB. Each page starts with a 96 byte header that stores information about the page. This information includes the page number, page type, the amount of free space on the page, and the allocation unit ID of the object that owns the page.

The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Pages

5/13/2013 1:16:29 PM - 5/14/2013 1:16:29 PM

STATISTIC NAME	VALUE FROM LAST POLL
Page Lookups/sec	530.69 K
Page Reads/sec	0.07
Page Writes/sec	2.?8

Tuesday, May 14, 2013 12:46 AM

Page Lookups/sec:	568.85 K
Page Reads/sec:	0.05
Page Writes/sec:	2.37 K
Page Splits/sec:	0.16
Page Life Expectancy:	97h 55m 15s
Database Pages:	755.10 K

Page Lookups/Sec:	**Definition:** This performance counter returns the number of requests to find a page in the buffer pool.

	Information: (Page lookups/sec) / (Batch Requests/sec) > 100. **Possible problems:** When the ratio of page lookups to batch requests is much greater than 100, this is an indication that while query plans are looking up data in the buffer pool, these plans are inefficient. **Remediation:** Identify queries with the highest amount of logical I/O's and tune them.
Page Reads/Sec	**Definition:** This performance counter returns the number of physical database page reads issued. **Information:** 80 – 90 physical database page reads per second is normal. **Possible problems:** Returned values that are high could indicate indexing or memory constraint. **Remediation:** Attempt to tune the application so that fewer I/O operations are required. For example, perhaps I/O operations would be reduced if there were the appropriate indexes, or if the database design were de-normalized. If the applications cannot be tuned, you will need to acquire disk devices with more capacity. Because physical I/O operations are expensive, you may be able to minimize the cost either by using a larger data cache, intelligent indexes, more efficient queries, or by changing the database design.
Page Lookups/ Batch Request	**Definition:** This performance counter displays the number of page splits per second that occur as the result of overflowing index pages.

	Information: (Page lookups/sec) / (Batch Requests/sec) > 100. **Possible problems:** When the ratio of page lookups to batch requests is much greater than 100, this is an indication that while query plans are looking up data in the buffer pool, these plans are inefficient. **Remediation:** Identify queries with the highest amount of logical I/O's and tune them.
Page Writes/Sec	**Definition:** This performance counter returns the number of physical database page writes issued. **Information:** 80 – 90 physical database page writes per second is normal. **Possible problems:** If the returned values are high, you should check the Lazy Writer/sec monitor.
Page Faults/Sec	**Definition:** This performance counter returns the average number of pages faulted per second. **Information:** This performance counter gives an idea of how many times information being requested is not where the application expects it to be. The information must either be retrieved from another location in memory or from the pagefile. While a sustained value may indicate trouble, you should be more concerned with hard page faults that represent actual reads or writes to the disk. Disk access is much slower than RAM. **Possible problems:** Any measurement higher than zero delays response time and probably indicates that more RAM is needed.

	Remediation: Add additional memory to your SQL server.
Page Splits/Batch Request	**Definition:** This performance counter displays the number of page splits per second that occur as the result of overflowing index pages. **Information:** The returned value needs to be low as possible. **Possible problems:** High values could mean poor table or index design. **Remediation:** If the number of page splits is high, consider increasing the Fill Factor of your indexes. An increased Fill Factor helps to reduce page splits because there is more room in data pages before it fills up and a page split has to occur. Note that this counter also includes the new page allocations as well and does not necessarily pose a problem. The other place we can confirm the page splits that involve data or index rows moves are the fragmented indexes on page splits.
Page Splits/Sec	**Definition:** This performance counter returns the number of page splits per second. **Information:** The returned value for this monitor should be as low as possible. **Possible problems:** Returned values that are high can indicate the overflowing of index pages. A high value for this counter is not bad in situations where many new pages are being created, since it includes new page allocations.

	Remediation: To avoid Page Splits, review the table and index design so as to reduce non-sequential inserts. You can also implement Fillfactor and Pad_Index to leave more empty space per page.
Page Life Expectancy	**Definition:** This performance counter returns the number of seconds a page will stay in the buffer pool without references. **Information:** This performance monitor reports, on average, how long data pages are staying in the buffer. **Possible problems:** If this value gets below 300 seconds, this is a potential indication that your SQL Server could use more memory in order to boost performance. Consistently having pages that stay in memory for less than that amount of time may indicate the need for more memory. **Remediation:** Add additional memory to your SQL server.
Database Pages	**Definition:** This metric tells you the number of database pages that are currently being occupied in the data cache. **Information:** The higher the buffer manager Database Pages is, the less room there is for SQL Server to cache more data pages. This means that SQL Server may have to free up data cache pages order to make room for pages being moved in from disk to the data cache, which can increase disk IO and hurt performance. There are no specific thresholds for this counter as each server is different. Instead, watch baseline values and look for sudden changes in the baseline value. **Possible problems:** If the value for this performance counter increases above its typical baseline value, this may indicate memory pressure for

	the SQL Server instance. **Remediation:** Investigate buffer management and disk I/O.
Free Memory **Note:** This performance counter is only available in SQL Server 2012.	**Definition:** The amount of memory available for SQL Server to use. Information: When SQL Server is using memory dynamically, it queries the system periodically to determine the amount of free memory. Maintaining this free memory prevents the operating system (OS) from paging. If less memory is free, SQL Server releases memory to the OS. If more memory is free, SQL Server may allocate more memory. SQL Server adds memory only when its workload requires more memory; a server at rest does not increase the size of its virtual address space. The returned value should be as high as possible. **Possible problems:** If Resource Semaphore does not find enough free memory, then it puts the current query into the waiting queue. **Remediation:** • Increase RAM • Increase SQL Server’s Max Server Memory • Consider OS requirements and other instances
Free Pages	**Definition:** This performance counter displays the total number of pages on all free lists. **Information:** This is not main indicator of memory problems and could only used as signal of possible memory pressure. **Possible problems:** If this performance counter is critical and other memory counters are good, it is possible that there are no problems

	with memory. **Remediation:** Check other memory counters. If they have critical values, you may try to install additional memory into SQL server.

Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

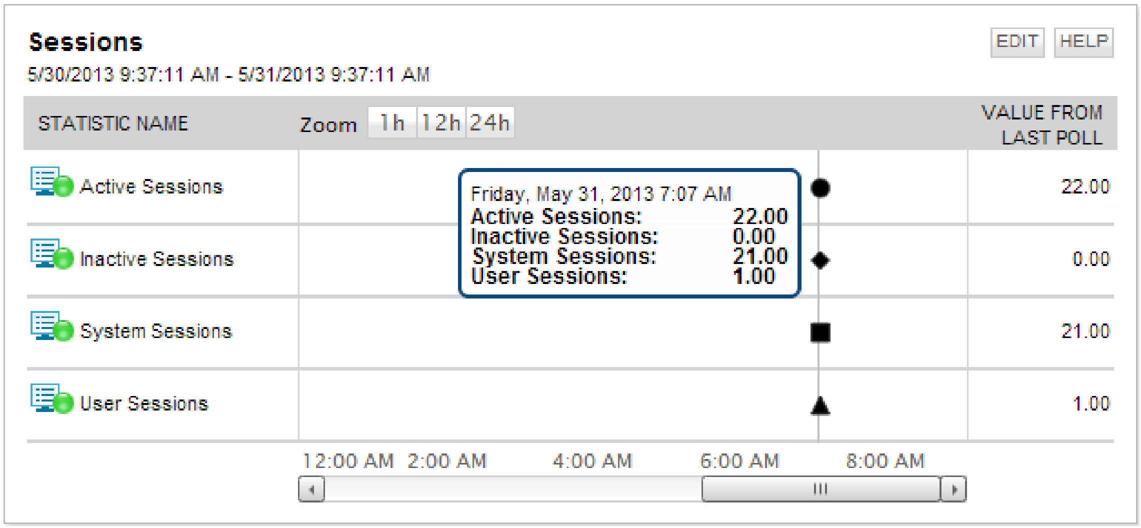
Sessions

This resource displays the current number of sessions.

Sessions include both connections and internal background processes like ghost cleanup , checkpoint, and so on. There is a close correlation between connections and sessions, but they are not necessarily the same.

The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Sessions

5/30/2013 9:37:11 AM - 5/31/2013 9:37:11 AM

STATISTIC NAME	VALUE FROM LAST POLL
Active Sessions	22.00
Inactive Sessions	0.00
System Sessions	21.00
User Sessions	1.00

Friday, May 31, 2013 7:07 AM
Active Sessions: 22.00
Inactive Sessions: 0.00
System Sessions: 21.00
User Sessions: 1.00

Active Sessions	**Definition:** A current connection that is actively processing. **Information:** To find information about an active session, Use Sp_Who2 Active or Sys.Dm_Exec_Sessions (Available in SQL 2005 or greater). **Possible problems:** Blocking session. **Remediation:** Wait until blocking clears or kill the session. If killed, the process would roll back changes. If there are many changes, it could take a while to roll-back.
Inactive Sessions	**Definition:** A current connection that is not currently processing. **Information:** To find information about an inactive sessions, use Sp_Who2. **Possible problems:** Blocking session. **Remediation:** Kill the session if blocking a process needs to finish. If killed, the

	process would roll back changes. If there are many changes, it could take a while to roll-back.
System Sessions	**Definition:** A System initiated connection. **Information:** Normally these numbers are less than 50. **Possible problems:** Blocking session. **Remediation:** Do not kill system sessions.
User Sessions	**Definition:** A user initiated connection. **Information:** Normally these numbers are greater than 50. **Possible problems:** Blocking session. **Remediation:** Kill the session if blocking a process needs to finish. If killed, the process will roll back changes. If there are many changes, it could take a while to roll-back.

Zooming:

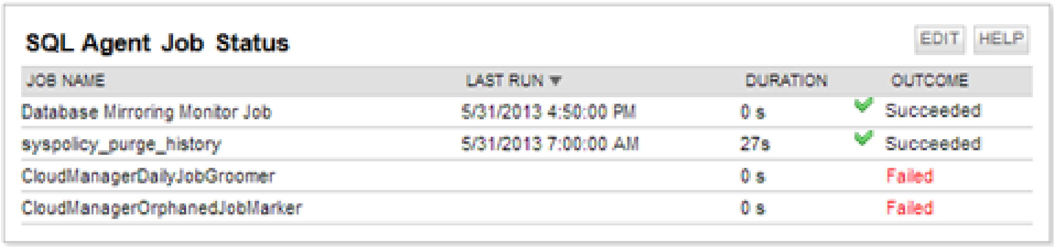
You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

SQL Agent Job Status

The SQL Agent is a Microsoft Windows service that executes scheduled administrative tasks known as "jobs" in SQL Server. This resource shows the status and details of these jobs.

The data can be sorted by clicking the headers of each column.



SQL Agent Job Status

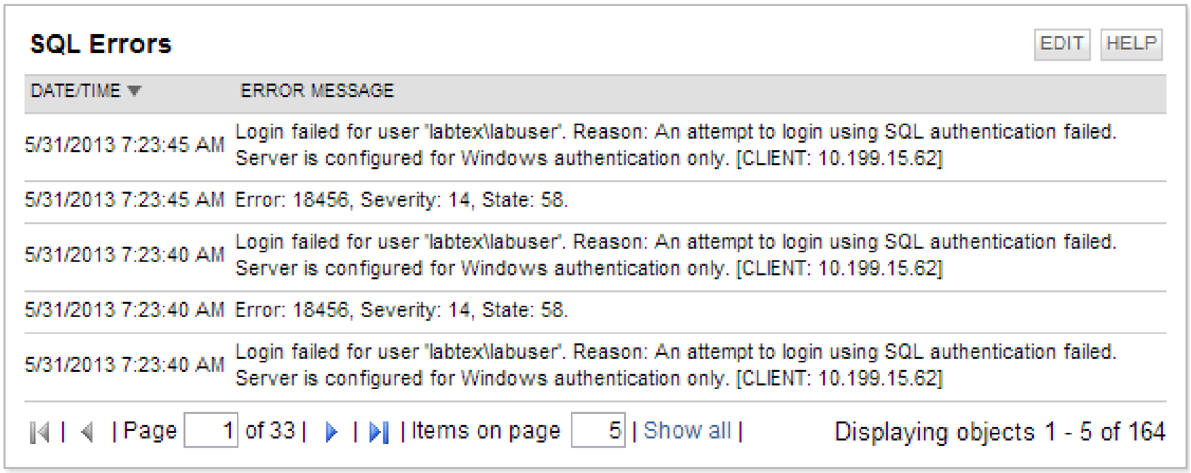
JOB NAME	LAST RUN ▼	DURATION	OUTCOME
Database Mirroring Monitor Job	5/31/2013 4:50:00 PM	0 s	✓ Succeeded
syspolicy_purge_history	5/31/2013 7:00:00 AM	27s	✓ Succeeded
CloudManagerDailyJobGroomer		0 s	Failed
CloudManagerOrphanedJobMarker		0 s	Failed

- **Job Name:** Displays the name of the listed job.
- **Last Run:** Displays the date and time the listed job was last run.
- **Duration:** Displays the amount of time the listed job took to complete.
- **Outcome:** Displays the result from having run the listed job.

SQL Agent Job Info, is used to provide metrics for this resource and cannot be edited.

SQL Errors

This resource displays SQL events that have occurred on the current SQL instance. The data can be sorted by clicking the headers of each column.



SQL Errors

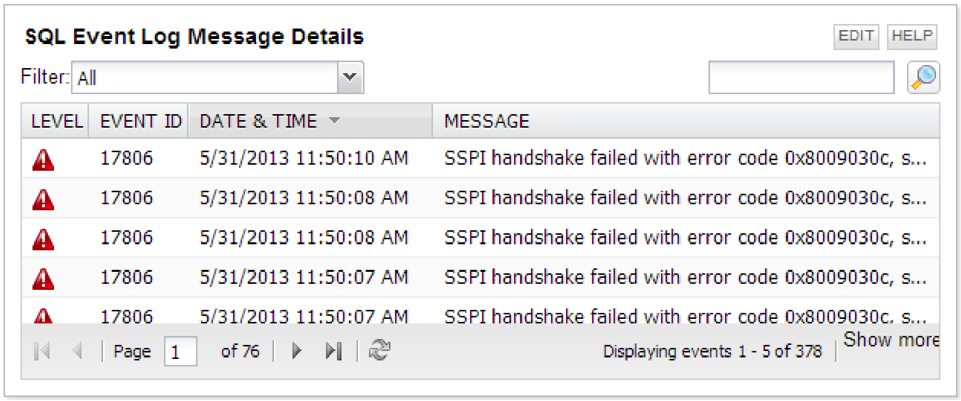
DATE/TIME ▼	ERROR MESSAGE
5/31/2013 7:23:45 AM	Login failed for user 'labtex\labuser'. Reason: An attempt to login using SQL authentication failed. Server is configured for Windows authentication only. [CLIENT: 10.199.15.62]
5/31/2013 7:23:45 AM	Error: 18456, Severity: 14, State: 58.
5/31/2013 7:23:40 AM	Login failed for user 'labtex\labuser'. Reason: An attempt to login using SQL authentication failed. Server is configured for Windows authentication only. [CLIENT: 10.199.15.62]
5/31/2013 7:23:40 AM	Error: 18456, Severity: 14, State: 58.
5/31/2013 7:23:40 AM	Login failed for user 'labtex\labuser'. Reason: An attempt to login using SQL authentication failed. Server is configured for Windows authentication only. [CLIENT: 10.199.15.62]

|◀ | ◀ | Page 1 of 33 | ▶ | ▶| | Items on page 5 | Show all | Displaying objects 1 - 5 of 164

- **Date/Time:** Displays the date and time that the error occurred.
- **Error Message:** Displays the error message that occurred at the listed date and time.

SQL Event Log Message Details

This resource displays the SQL event log messages while providing a search window and a filter. The filter allows you to choose which types of log messages you want to display. The data can be sorted by clicking the headers of each column. This resource shows only *Error* and *Warning* events specific to the monitored SQL instance.



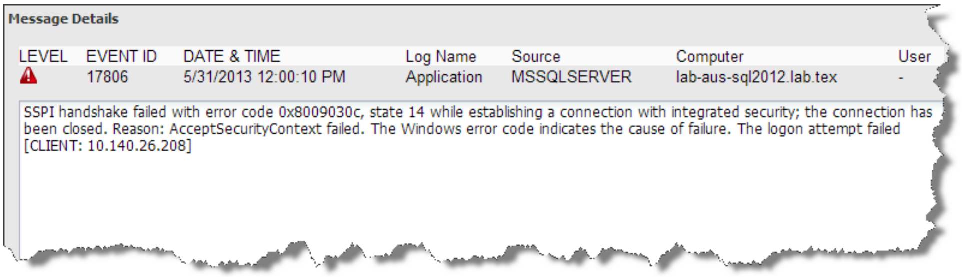
SQL Event Log Message Details

Filter: All

LEVEL	EVENT ID	DATE & TIME	MESSAGE
⚠	17806	5/31/2013 11:50:10 AM	SSPI handshake failed with error code 0x8009030c, s...
⚠	17806	5/31/2013 11:50:08 AM	SSPI handshake failed with error code 0x8009030c, s...
⚠	17806	5/31/2013 11:50:08 AM	SSPI handshake failed with error code 0x8009030c, s...
⚠	17806	5/31/2013 11:50:07 AM	SSPI handshake failed with error code 0x8009030c, s...
⚠	17806	5/31/2013 11:50:07 AM	SSPI handshake failed with error code 0x8009030c, s...

Page 1 of 76 Displaying events 1 - 5 of 378

Clicking any event message in the list will open a new window, displaying the entire message, as shown below:



Message Details

LEVEL	EVENT ID	DATE & TIME	Log Name	Source	Computer	User
⚠	17806	5/31/2013 12:00:10 PM	Application	MSSQLSERVER	lab-aus-sql2012.lab.tex	-

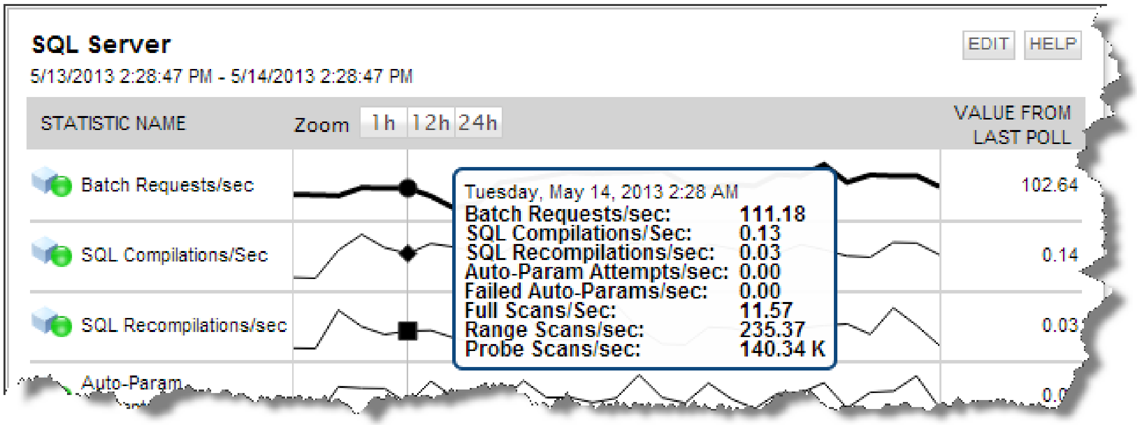
SSPI handshake failed with error code 0x8009030c, state 14 while establishing a connection with integrated security; the connection has been closed. Reason: AcceptSecurityContext failed. The Windows error code indicates the cause of failure. The logon attempt failed [CLIENT: 10.140.26.208]

SQL Server

This resource shows the status of SQL server counters that warrant a view independent of grouped resources.

The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of the current statistic crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of the current statistic crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



SQL Server

5/13/2013 2:28:47 PM - 5/14/2013 2:28:47 PM

STATISTIC NAME	VALUE FROM LAST POLL
Batch Requests/sec	102.64
SQL Compilations/Sec	0.14
SQL Recompilations/sec	0.03
Auto-Param	0.0

```
Tuesday, May 14, 2013 2:28 AM
Batch Requests/sec:          111.18
SQL Compilations/Sec:        0.13
SQL Recompilations/sec:      0.03
Auto-Param Attempts/sec:     0.00
Failed Auto-Params/sec:      0.00
Full Scans/Sec:              11.57
Range Scans/sec:             235.37
Probe Scans/sec:             140.34 K
```

Batch Requests/Sec	**Definition:** This performance counter returns the number of Batch Requests that SQL Server receives per second. **Information:** The values this monitor returns generally follows in step as to how busy your server's CPUs are. From a network bottleneck approach, a typical 100Mbs network card is only able to handle about 3,000 batch requests per second. **Possible problems:** Generally speaking, over 1,000 batch requests per second indicates a very busy SQL Server. If this is the case, you may soon experience a CPU bottleneck, if you are not already. Of course, this is a relative number, and the more powerful your hardware, the more batch requests per second your SQL Server can handle. Sometimes low Batch Requests/Sec can be misleading. If there were a SQL Statements/sec counter, this would be a more accurate measure of the

	amount of SQL Server activity. For example, an application may call only a few stored procedures, yet each stored procedure does a great deal of work. In this case, we will see a low number for Batch Requests/sec, but each stored procedure (one batch) will execute many SQL statements that drive up CPU and other resources. As a result, many counter thresholds based on the number of Batch Requests/sec will seem to identify issues because the batch requests on such a server are unusually low for the level of activity on the server. **Remediation:** Check your SQL server and verify system resources are being used efficiently. Applications other than SQL may be using unnecessary system resources. Try and recover memory by closing unnecessary applications. Installing additional memory and upgrading your hardware should solve this problem.
SQL Compilations/Sec	**Definition:** This performance counter returns the number of times per second that SQL Server compilations have occurred. **Information:** This value should be as low as possible. **Possible problems:** If you see a high value, say above 100, then this can be an indication that there are a great deal of ad hoc queries that are running which may cause increased CPU usage. **Remediation:** Re-write the running ad hoc queries as stored procedures or use the following command: sp_ executeSQL.

SQL Recompilations/Sec	**Definition:** This performance counter returns the number of SQL statement recompiles that are triggered per second. **Information:** When an execution plan is invalidated due to some significant event, SQL Server will re-compile it. Re-compiles, like compiles, are expensive operations so you want to minimize the number of re-compiles. Ideally you want to keep this counter less than 10% of the number of Compilations/Sec. In other words, keep this value as low a possible. In SQL Server 2000, when SQL Server recompiles a stored procedure, the entire stored procedure is recompiled, not just the statement that triggered the recompilation. In SQL Server 2008 and SQL Server 2005, the behavior is changed to statement-level recompilation of stored procedures. When SQL Server 2008 or SQL Server 2005 recompiles stored procedures, only the statement that caused the recompilation is compiled, not the entire procedure. This uses less CPU bandwidth and results in less contention on lock resources such as Compile locks. **Possible problems:** Returned values that are high can indicate deadlocks and compile locks that are not compatible with any locking type. **Remediation:** Recompilation can happen for various reasons, such as: Schema changed; Statistics changed; Deferred compile; Set option changed; Temporary table changed; Stored procedure created with the Recompile query hint or using the Option (Recompile). • If the recompile occurred because a Set option changed, use SQL Server Profiler to determine which Set option changed. Avoid

	changing Set options within stored procedures. It is better to set them at the connection level. Ensure that Set options are not changed during the lifetime of the connection. • Recompilation thresholds for temporary tables are lower than for normal tables. If the recompiles on a temporary table are due to statistics changes, you can change the temporary tables to table variables. A change in the cardinality of a table variable does not cause a recompilation. The drawback of this approach is that the query optimizer does not keep track of a table variable's cardinality because statistics are not created or maintained on table variables. This can result in non-optimal query plans. Test the different options and choose the best one. • Another option is to use the Keep Plan query hint. This sets the threshold of temporary tables to be the same as that of permanent tables. The EventSubclass column indicates that "Statistics Changed" for an operation on a temporary table. • To avoid recompilations that are due to changes in statistics (for example, when the plan becomes suboptimal due to change in the data statistics), specify the KeepFixed Plan query hint. With this option in effect, recompilations can only happen because of correctness-related reasons (for example, when the underlying table structure has changed and the plan no longer applies) and not due to statistics. An example might be when a recompilation occurs if the schema of a table that is referenced by a statement changes, or if a table is marked with the sp_recompile stored procedure. • Turning off the automatic updates of statistics for indexes and statistics that are defined on a table

	or indexed view prevents recompiles that are due to statistics changes on that object. Note, however, that turning off the Auto-Stats feature by using this method is usually not a good idea. This is because the query optimizer is no longer sensitive to data changes in those objects and sub-optimal query plans might result. Use this method only as a last resort after exhausting all other alternatives. • Batches should have qualified object names (for example, dbo.Table1) to avoid recompilation and to avoid ambiguity between objects. • To avoid recompiles that are due to deferred compiles, do not interleave DML and DDL or create the DDL from conditional constructs such as If statements. • Run Database Engine Tuning Advisor (DTA) to see if any indexing changes improve the compile time and the execution time of the query. • Check to see if the stored procedure was created with the With Recompile option or if the Recompile query hint was used. If a procedure was created with the With Recompile option, in SQL Server 2005, you may be able to take advantage of the statement level Recompile hint if a particular statement within that procedure needs to be recompiled. This would avoid the necessity of recompiling the whole procedure each time it executes, while at the same time allowing the individual statement to be compiled.
Auto-Param Attempts/Sec:	**Definition:** This monitor returns the number of auto-parameterization attempts per second. **Information:** The total for this monitor should be the sum of the

	failed, safe, and unsafe auto-parameterizations. Auto-parameterization occurs when an instance of SQL Server tries to parameterize a Transact-SQL request by replacing some literals with parameters so that reuse of the resulting cached execution plan across multiple similar-looking requests is possible. Note that auto-parameterizations are also known as simple parameterizations in newer versions of SQL Server. This counter does not include forced parameterizations. **Possible problems:** Bad T-SQL coding practices can increase recompilation frequency and adversely affect SQL Server's performance. Such situations can be debugged and corrected in many cases.
Longest Transaction Running Time	**Definition:** The performance counter displays the length of time, in seconds, the transaction that has been running the longest, has been active. **Information:** Transactions that run longer than others use more resources. They can be used to track down procedures and calls that are taking longer than expected by identifying the specific transaction(s). **Possible problems:** Long running transactions can prevent truncation of transaction logs. This can cause the transaction log files to grow until they consume all available physical disk space shutting down access to the database. **Remediation:** Check the functionality of the query and/or redesign the long running transaction.
Failed Auto-	**Definition:**

Params/Sec	This monitor returns the number of failed auto-parameterization attempts per second. **Information:** The value returned by this monitor should be low. Note that auto-parameterizations are also known as simple parameterizations in later versions of SQL Server. Preferred values should be near zero. **Possible problems:** Bad T-SQL coding practices can increase recompilation frequency and adversely affect SQL Server's performance. Such situations can be debugged and corrected in many cases.
Full Scans/Sec	**Definition:** This performance counter returns the number of Full Scans on base tables or indexes. **Information:** This is the number of unrestricted full scans per second. These can be either base-table or full-index scans. **Possible problems:** Values greater than 1 or 2 indicate table/Index page scans are occurring. If the CPU is running high, you should investigate the cause as related to this counter measurement. You can rule this out as a problem if the full scans are on small tables. **Remediation:** Following are a few of the main causes of high Full Scans/sec: • Missing indexes • Too many rows requested; Queries with missing indexes or too many rows requested will have a large number of logical reads and an increased CPU time.

	• Scans are IO-intensive and should run within your databases minimally. Identify the tables that have a large number of scans against them. Review the fill factor you have set up on your indexes and minimize it where appropriate.
Range Scans/Sec	**Definition:** This performance counter returns the number of Qualified Range Scans through indexes per second. **Information:** This monitor returns the number of qualified range scans through indexes per second.
Plan Re-Use	**Definition:** A query plan is used to execute a query. **Information:** Plan re-use is desirable for OLTP workloads because re-creating the same plan (for similar or identical transactions) is a waste of CPU resources. **Possible problems:** Zero cost plans will not be cached (not re-used) in SQL 2005 SP2. Applications that use zero cost plans will have a lower plan re-use but this is not a performance issue. **Remediation:** Review your plan re-use design. Tune your plan re-use design as it is described in the following article: http://sqlmag.com/sql-server/fine-tuning-plan-reuse.
Probe Scans/Sec	**Definition:** This performance counter returns the number of Probe Scans, per second, that are used to find at most, one single qualified row in an index or base table directly. **Information:** There are no thresholds associated with this

	performance counter.
Recompilations/ Compilations	**Definition:** This performance counter shows the ratio of SQL Recompilations to SQL Compilations. **Information:** SQL Recompilations should be less than 10% of SQL Compilations. **Possible problems:** Returned values that are high can indicate more temporary tables in use. **Remediation:** Change stored procedures to not change schemas, Use table variables instead of temporary tables
Compilations/ Recompilations/Sec	**Definition:** Monitoring the number of query compilations and recompilations and the number of batches received by an instance of SQL Server gives you an indication of how quickly SQL Server is processing user queries and how effectively the query optimizer is processing the queries. **Information:** Compilation is a significant part of a query's turnaround time. In order to save the compilation cost, the Database Engine saves the compiled query plan in a query cache. The objective of the cache is to reduce compilation by storing compiled queries for later reuse, therefore ending the requirement to recompile queries when later executed. However, each unique query must be compiled at least one time. Compilations/sec divided by 10 minus recompilations/sec. **Possible problems:** Query recompilations can be caused by the following

	factors: - Schema changes, including base schema changes such as adding columns or indexes to a table, or statistics schema changes such as inserting or deleting a significant number of rows from a table. - Environment (Set statement) changes. Changes in session settings such as Ansi_Padding or Ansi_Nulls can cause a query to be recompiled.

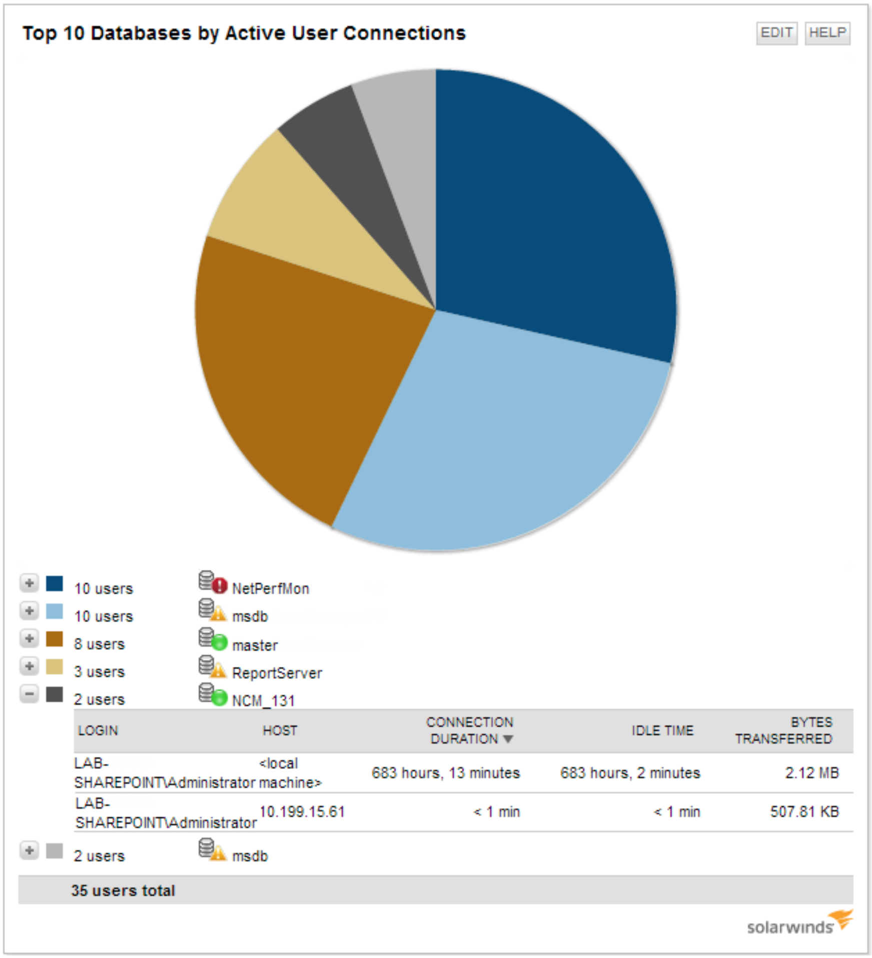
Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Top 10 Databases by Active User Connections

This resource graphically shows the number of users per database for the top 10 most active databases. The colors of the chart correspond to the labels in the legend, which indicate the most active databases.



Top 10 Databases by Active User Connections

+	🟦	10 users	NetPerfMon
+	🟦	10 users	msdb
+	🟫	8 users	master
+	🟨	3 users	ReportServer
−	⬛	2 users	NCM_131

LOGIN	HOST	CONNECTION DURATION ▼	IDLE TIME	BYTES TRANSFERRED
LAB-SHAREPOINT\Administrator	<local machine>	683 hours, 13 minutes	683 hours, 2 minutes	2.12 MB
LAB-SHAREPOINT\Administrator	10.199.15.61	< 1 min	< 1 min	507.81 KB

+	⬜	2 users	msdb

35 users total

The performance counter, **Top Active Sessions with Statistics**, is used to provide metrics for this resource and cannot be edited.

Expanding the tree by clicking **[+]** will provide details about the users, host, length of connection and idle time, as well as the number of bytes transferred during that time.

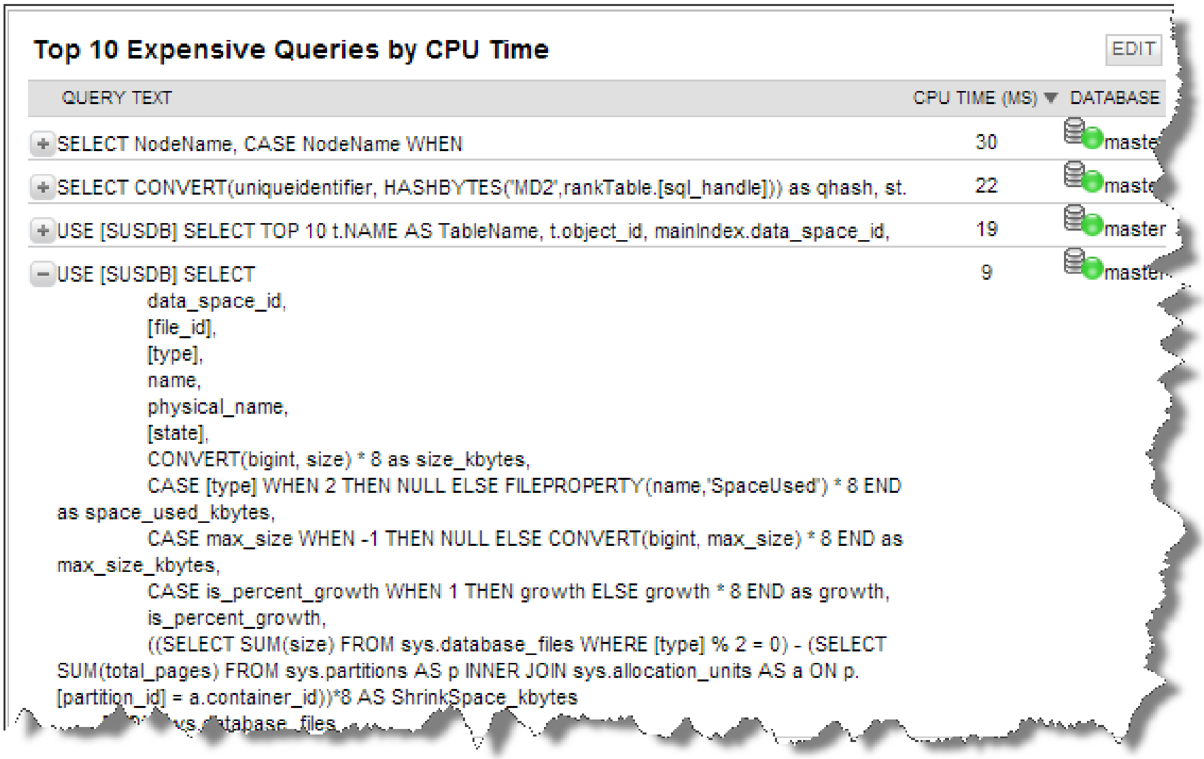
Top 10 Most Expensive Queries by CPU Time

This resource displays the top 10 most expensive queries sorted by the length of time the CPU spent processing the listed query.

When viewing this resource from the *Summary* tab on the_*Application Details* page, queries are listed for the entire SQL Server instance. Conversely, when this same resource appears on the *Database Details* view, queries are only listed for that specific database.

The data can be sorted by clicking the headers of each column. Additionally, this resource can optionally show *Top Queries by I/O* by clicking **Edit** in this resource and making the necessary adjustments.

Note: The resource is filtered based on the settings found on the *Queries* tab in *Details* view in the **Most Expensive Queries** resource.



Top 10 Expensive Queries by CPU Time

QUERY TEXT	CPU TIME (MS) ▼	DATABASE
+ SELECT NodeName, CASE NodeName WHEN	30	maste
+ SELECT CONVERT(uniqueidentifier, HASHBYTES('MD2',rankTable.[sql_handle])) as qhash, st.	22	maste
+ USE [SUSDB] SELECT TOP 10 t.NAME AS TableName, t.object_id, mainIndex.data_space_id,	19	master
− USE [SUSDB] SELECT	9	master

```
USE [SUSDB] SELECT
            data_space_id,
            [file_id],
            [type],
            name,
            physical_name,
            [state],
            CONVERT(bigint, size) * 8 as size_kbytes,
            CASE [type] WHEN 2 THEN NULL ELSE FILEPROPERTY(name,'SpaceUsed') * 8 END
as space_used_kbytes,
            CASE max_size WHEN -1 THEN NULL ELSE CONVERT(bigint, max_size) * 8 END as
max_size_kbytes,
            CASE is_percent_growth WHEN 1 THEN growth ELSE growth * 8 END as growth,
            is_percent_growth,
            ((SELECT SUM(size) FROM sys.database_files WHERE [type] % 2 = 0) - (SELECT
SUM(total_pages) FROM sys.partitions AS p INNER JOIN sys.allocation_units AS a ON p.
[partition_id] = a.container_id))*8 AS ShrinkSpace_kbytes
      sys.database_files
```

- **Query Text:** The text shown is a preview of the full query. To view an entire query in the list, click **[+]** next to the query listed to expand and view the query.
  Note: Clicking any listed query will take you to the *Most Expensive Queries* resource on the *Queries* tab, providing greater detail about the query.
- **CPU Time:** Display the time, in milliseconds, the listed query took to complete.
- **Database:** Displays the database with which the listed query is associated. The icon in this column shows the current state of the listed database.

The performance counter, **Top Expensive Queries**, is used to provide metrics for this resource and cannot be edited.

AppInsight for SQL Details View (Queries Tab)

The *Queries* tab provides detailed insight into the most expensive queries running on your SQL instance.

The performance counter, **Top Expensive Queries**, is used to provide metrics for this resource and cannot be edited.

Most Expensive Queries

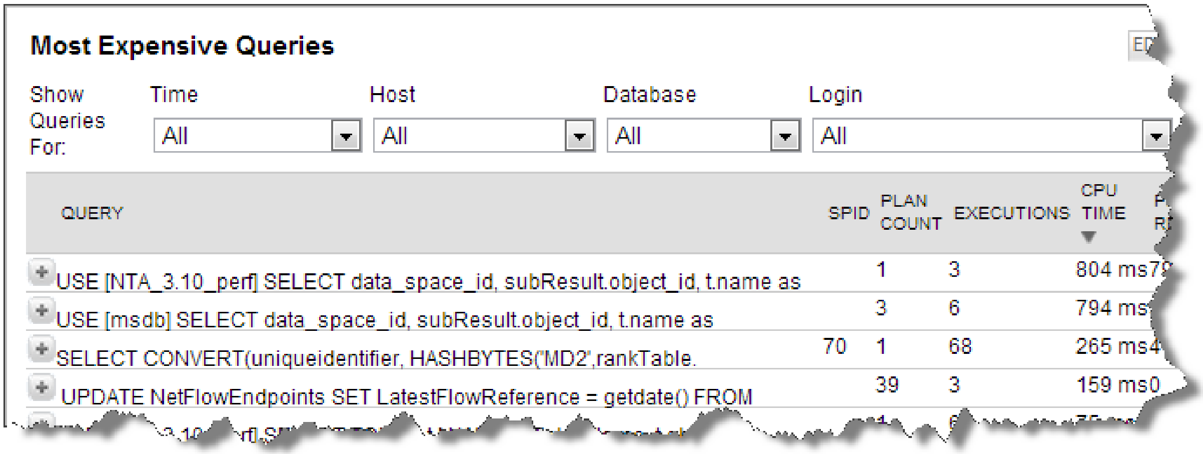
This resource provides detailed insight into the most expensive queries running on your SQL database. Queries for this view can be filtered by *Time*, *Host*, *Database*, and *Login* using the filter dropdown menus at the top of the window.

Note: The filters used here will also be applied to the **Top 10 Most Expensive Queries by CPU Time** resource found on the *Summary* tab in the *Details* view.

To view an entire query in the list, hover the mouse over a query, or click **[+]** next to the query listed to expand and view the query.

The data can be sorted by clicking the headers of each column.

Note: This resource will be empty when polling of queries is disabled.



Most Expensive Queries

Show Queries For: Time: All | Host: All | Database: All | Login: All

QUERY	SPID	PLAN COUNT	EXECUTIONS	CPU TIME
USE [NTA_3.10_perf] SELECT data_space_id, subResult.object_id, t.name as		1	3	804 ms79
USE [msdb] SELECT data_space_id, subResult.object_id, t.name as		3	6	794 ms
SELECT CONVERT(uniqueidentifier, HASHBYTES('MD2',rankTable.	70	1	68	265 ms4
UPDATE NetFlowEndpoints SET LatestFlowReference = getdate() FROM		39	3	159 ms0

- **Query Text:** The text shown is a preview of the full query. To view an entire query in the list, click **[+]** next to the query listed to expand and view the query.
- **SPID:** Server Process ID of the listed query.
- **Plan Count:** Displays the number of cached query plans for the listed query. A large number might indicate a need for explicit query parameterization.
- **Executions:** Displays the number of times the listed query was executed.

- **CPU Time:** Displays the amount of time, in milliseconds, that the listed query took to complete.
- **Phys. Read:** This field returns the number of physical reads. Physical reads are reads that require reading directly from disk. Physical reads are substantially slower than reading from the cache or from RAM.
- **Logical Reads:** Displays the number of times data was read from the cache.
- **Logical Writes:** Displays the number of times data was written to the cache.
- **Average Duration:** This displays the length of time it took, in milliseconds, to execute the listed query averaged by the number of executions.
- **Bytes Transferred:** Number of bytes transferred during the query.
- **Login:** Displays the current login information for the listed database.
- **Host:** Displays the host for the listed database.
- **Database:** Displays the current name for the listed database. Clicking any database in the list will take you to the *Database Details* page.

Database Details View

The *Database Detail*s view provides a great deal of information about the current database and can be accessed by clicking any listed database.

To navigate to the Database Details view:

1. From the web console, navigate to the *AppInsight for SQL Details* view.
2. From the *All Databases* resource, click any database in the list.

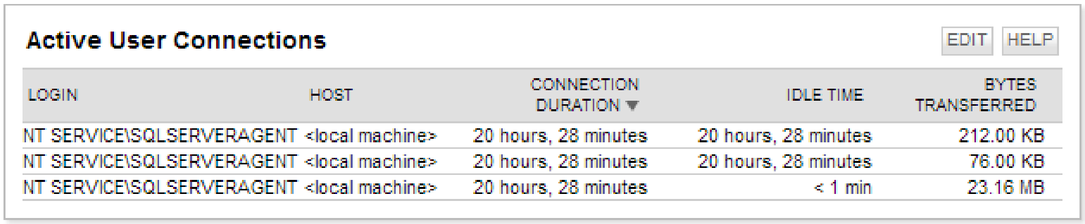
The available resources unique to this view include:

- Active User Connections
- Connections
- Database Availability
- Database Details
- Database Disk I/O
- Database Size by File
- Log Flushes
- Processes and Services
- Storage
- Top 10 Clustered Indexes by Fragmentation
- Top 10 Most Expensive Queries by CPU Time
- Top 10 Nonclustered Indexes by Fragmentation
- Top 10 Tables by Size

- Transaction Log Disk I/O
- Transaction Log Size by File

Active User Connections

This resource provides detailed information about the active user connections associated with the database selected. Data can be sorted by clicking the headers of each column.



Active User Connections

LOGIN	HOST	CONNECTION DURATION ▼	IDLE TIME	BYTES TRANSFERRED
NT SERVICE\SQLSERVERAGENT	<local machine>	20 hours, 28 minutes	20 hours, 28 minutes	212.00 KB
NT SERVICE\SQLSERVERAGENT	<local machine>	20 hours, 28 minutes	20 hours, 28 minutes	76.00 KB
NT SERVICE\SQLSERVERAGENT	<local machine>	20 hours, 28 minutes	< 1 min	23.16 MB

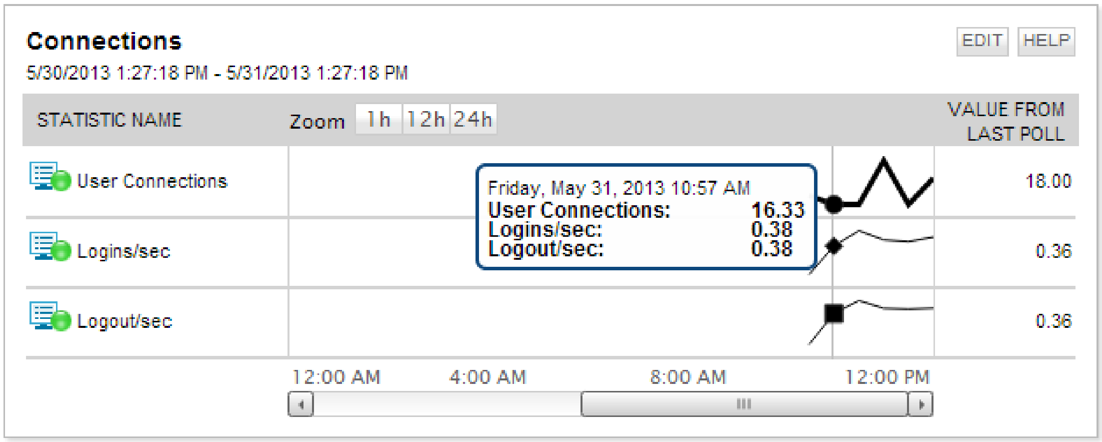
- **Login:** This displays the login information for the listed user connection.
- **Host:** This displays the type of host associated with the listed user connection.
- **Connection Duration:** This displays the length of time the listed user connection has been connected.
- **Idle Time:** This displays the length of time the listed user connection has been idle.
- **Bytes Transferred:** This displays the number of bytes transferred for the listed user connection.

The performance counter, **Top Active Sessions with Statistics**, is used to provide metrics for this resource and cannot be edited.

Connections

This resource shows the status of your current database connections. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Connections

5/30/2013 1:27:18 PM - 5/31/2013 1:27:18 PM

STATISTIC NAME	VALUE FROM LAST POLL
User Connections	18.00
Logins/sec	0.36
Logout/sec	0.36

Friday, May 31, 2013 10:57 AM
User Connections: 16.33
Logins/sec: 0.38
Logout/sec: 0.38

User Connections	**Definition:** The User Connections performance counter identifies the number of different users that are connected to your SQL Server at the time the sample was taken. **Information:** You need to watch this performance counter over time to understand your baseline user connection numbers. Since each user connection consumes some memory space, a high number of user connections can impact throughput and cause a performance slow-down. Once you have an idea of your high and low thresholds during normal usage of your system, you can then look for times when this counter exceeds these high and low marks. **Possible problems:** If the returned value of this performance counter goes down and the load on the system remains stable, you might have a bottleneck that is not allowing your server to handle the normal load. Keep in mind that this counter value might go down just because less people are using your SQL Server instance. If you see this number jump by 500% from your baseline, you may be seeing a slowdown of your server activity. **Remediation:** You may want to boost the SQL Server configuration setting,

	Maximum Worker Threads to a figure higher than the default setting of 255. The setting for Maximum Worker Threads should be higher than the maximum number of user connections your server ever reaches.
Logins/Sec	**Definition:** This performance counter returns the total number of logins started, per second, and does not include pooled connections. **Information:** Opening and closing connections is an expensive process. A pooled connection is one which is kept open by an application for other requests to re-use. **Possible problems:** If the returned value is high, this can indicate that the application is not correctly using connection pooling. **Remediation:** Review the Connection Polling configuration.
Logout/Sec	**Definition:** This performance counter returns the total number of logout operations started, per second. **Information:** Opening and closing connections is an expensive process. When applications do not use a connection pool, each request needs to establish its own connection before the query can be executed. It then has to close it. A pooled connection is one which is kept open by an application for other requests to re-use. **Possible problems:** If the returned value is high, this can indicate that the application is not correctly using connection pooling. **Remediation:** Review the Connection Polling configuration.

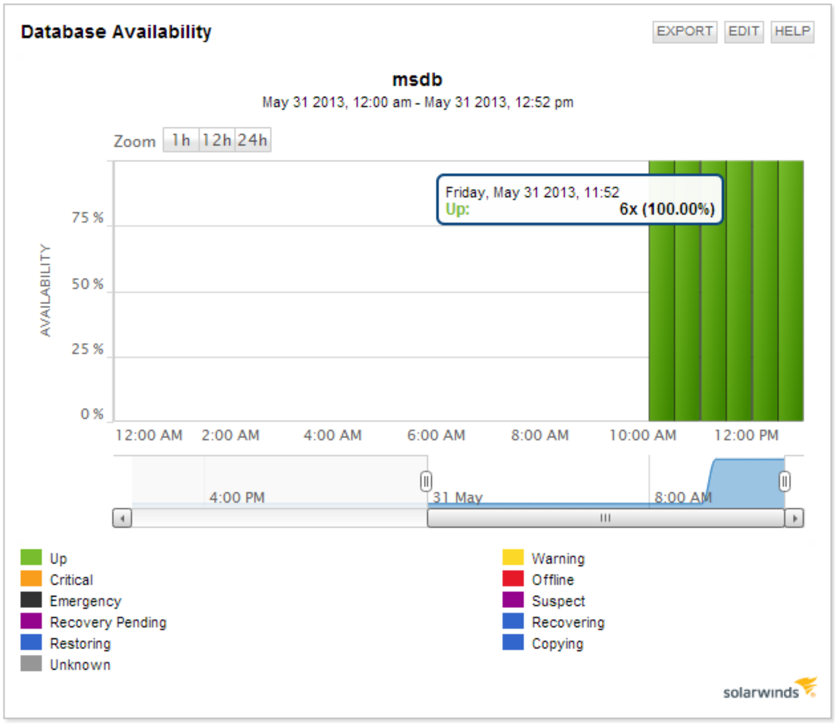
Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Database Availability

This resource graphically shows the various states of availability for the current database in a percentage format. This chart is made up of three sections: the legend, the main chart, and the lower chart. The colors of the main chart correspond to the labels in the legend.



Clicking the **Edit** button will allow you to change the default zoom range, the amount of historical data loaded, and set the sample interval time period, as well as reveal the Advanced section. Expanding the Advanced section will allow you to change the defaults of the titles from macros to anything you want.

Zooming

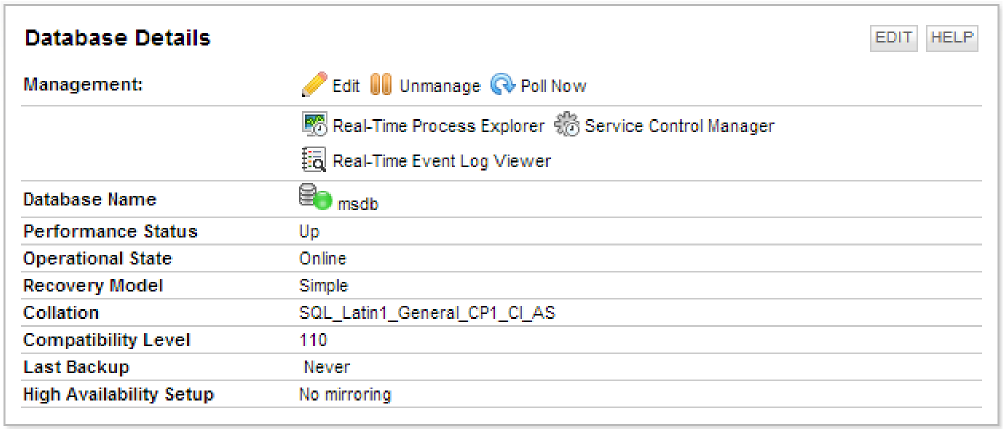
You can have the chart show a pre-determined, hourly time period of data by clicking on any one of the three *Zoom* buttons; **1h**, **12h**, or **24h**. Alternatively, you can have the chart show a specific time range by dragging the mouse over a small area of the chart.

Once you have a time period selected, the lower chart will zoom to the starting and ending values of the selected time period. With the lower chart you can zoom in further by fine tuning the view with the sliders. The main view of the chart will display the selected time period between the two sliders of the lower chart.

Moving the mouse over the main chart will reveal a tooltip with more detailed information about the specific time period the mouse is hovering over.

Database Details

This resource provides tools and details concerning the current database.



Database Details

Management:	Edit Unmanage Poll Now Real-Time Process Explorer Service Control Manager Real-Time Event Log Viewer
Database Name	msdb
Performance Status	Up
Operational State	Online
Recovery Model	Simple
Collation	SQL_Latin1_General_CP1_CI_AS
Compatibility Level	110
Last Backup	Never
High Availability Setup	No mirroring

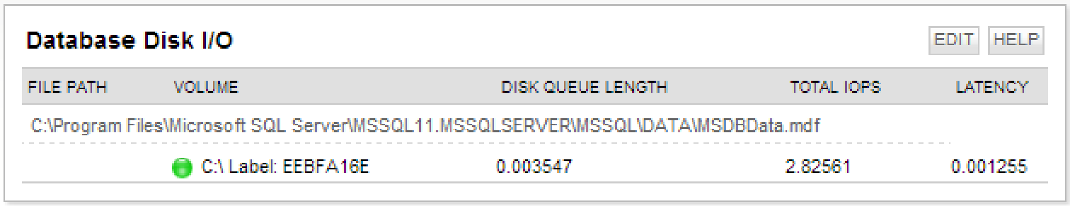
- **Management:** This provides tools allowing you to more easily manage and troubleshoot the current application.
- **Database Name:** Displays the name of the current database.
- **Performance Status:** Displays the status of the current database.
- **Operational State:** Displays whether or not the database is currently online.
- **Recovery Model:** Will display the type of recovery model employed on the listed database. This can be either, Simple, Full, or Bulk-Logged. For more information, refer to the following MSDN article: http://msdn.microsoft.com/en-us/library/ms175987(v=sql.105).aspx

- **Collation:** Displays the method of sorting used by the database. (In the above image, Latin1 means the server treats strings using Charset Latin 1, i.e. ASCII. CI means Case Insensitive. AS means Accent Sensitive ('ü' does not equal 'u')).
- **Compatibility Level:** Displays the compatibility level of the current database. The value displayed in this field correlates to the bulleted table below:
  - 60 = SQL Server 6.0
  - 65 = SQL Server 6.5
  - 70 = SQL Server 7.0
  - 80 = SQL Server 2000
  - 90 = SQL Server 2005
  - 100 = SQL Server 2008
  - 110 = SQL Server 2012
- **Last Backup:** Displays the date and time of the last successful backup for the current database.
- **High Availability Group:** Displays the database mirroring status and details of the mirrored database, if available.

Database Disk I/O

This resource displays the status, as well as I/O metrics, for your current database file. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any volume in the *Volume* column will take you to the *Volume Details* page for that counter.

The data can be sorted by clicking the headers of each column.



FILE PATH	VOLUME	DISK QUEUE LENGTH	TOTAL IOPS	LATENCY
C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\MSDBData.mdf				
	● C:\ Label: EEBFA16E	0.003547	2.82561	0.001255

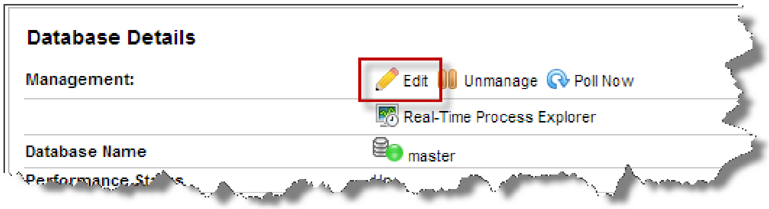
- **File Path:** Displays the full path of the listed database.
- **Volume:** Displays the name of the volume of the current database. Clicking a listed volume will take you to the Volume Details page.
- **Disk Queue Length:** This is the number of both read and write requests that were queued for the selected disk during the sample interval.
- **Total IOPS:** Displays the current total read and write I/O operations occurring for the database logical volume. High I/O may be normal for some databases, but excessively high I/O will lead to higher queue lengths, meaning

there are more requests for disk reads or writes than the storage subsystem can handle simultaneously.
- **Latency:** Displays a measurement of disk latency occurring on the disk where the database resides. High latency values indicate sluggish database performance

The performance counter, **Database Files**, is used to provide metrics for this resource. It is used in combination with volume polling. This counter is not listed in the resource; however, the thresholds can be changed.

To change the thresholds for the Database Files performance counter:

1. From the *Database Details* view, locate the **Database Files** resource and click **Edit** in the *Management* field.



Database Details

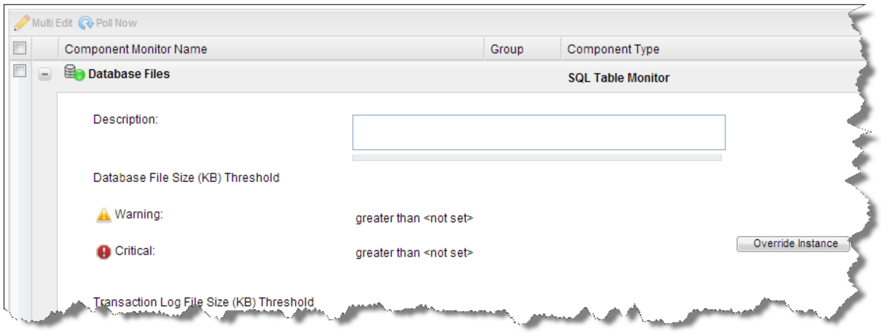
Management: Edit Unmanage Poll Now

Real-Time Process Explorer

Database Name: master

Performance Status

2. Locate the **Database Files** performance counter and click **[+]** to expand the counter and reveal the counter details.



	Component Monitor Name	Group	Component Type
□ ⊟	Database Files		SQL Table Monitor

Description:

Database File Size (KB) Threshold

Warning: greater than <not set>

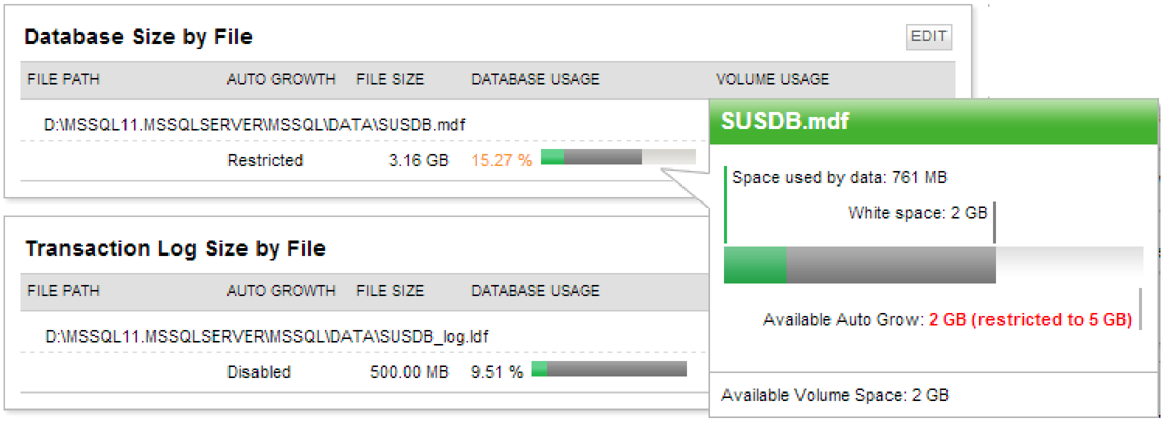
Critical: greater than <not set>

Transaction Log File Size (KB) Threshold

3. Click **Override Template** and adjust the thresholds to suit your needs.
4. When done, click **Submit**.

Database Size by File

This resource lists the current databases sorted by file size. Moving the mouse over the Database Usage percent value will reveal a tooltip with more detailed information about that database. Clicking the value will take you to the *Custom Chart* resource page. If the *Warning* or *Critical* thresholds is breached, the color of the bar chart will change from green to yellow or red, respectively.



Database Size by File

EDIT

FILE PATH	AUTO GROWTH	FILE SIZE	DATABASE USAGE	VOLUME USAGE
D:\MSSQL11.MSSQLSERVER\MSSQL\DATA\SUSDB.mdf				
	Restricted	3.16 GB	15.27 %	

SUSDB.mdf

Space used by data: 761 MB

White space: 2 GB

Available Auto Grow: 2 GB (restricted to 5 GB)

Available Volume Space: 2 GB

Transaction Log Size by File

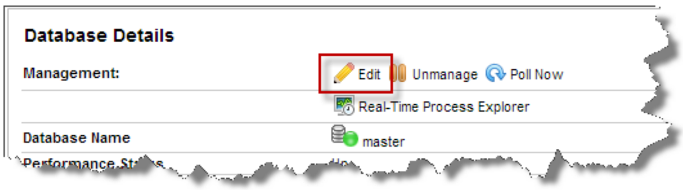
FILE PATH	AUTO GROWTH	FILE SIZE	DATABASE USAGE
D:\MSSQL11.MSSQLSERVER\MSSQL\DATA\SUSDB_log.ldf			
	Disabled	500.00 MB	9.51 %

- **File Path:** Displays the full path of the listed database.
- **Auto-Growth:** Displays the Auto-growth status of the listed database.
- **File Size:** Displays the size of the listed database in megabytes.
- **Database Usage:** Depending on how your database is configured, this value is showing you how close you are to running out of space. If your database is configured as a fixed size, auto-grow, or a restricted auto-grow, the percentages relate to what is going to cause your database to run out of space.
- **Volume Usage:** The value returned is the database file usage in relation to the space remaining on the volume.

Database Files, is used to provide metrics for this resource. This counter is not listed in the resource; however, the thresholds can be changed.

To change the thresholds for the Database Files performance counter:

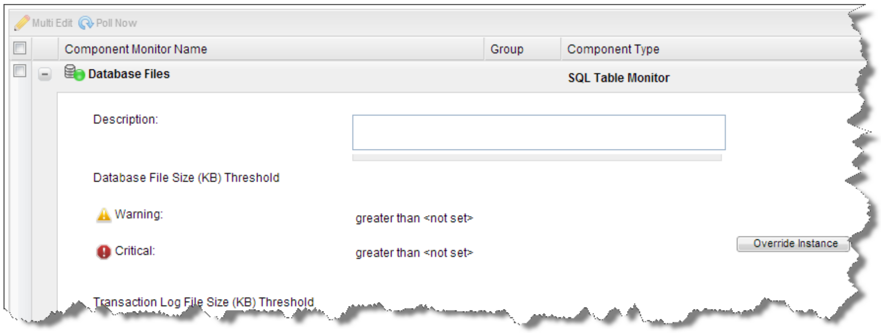
1. From the *Database Details* view, locate the **Database Files** resource and click **Edit** in the *Management* field.



Database Details

Management:	Edit Unmanage Poll Now Real-Time Process Explorer
Database Name	master
Performance Status	

2. Locate the **Database Files** performance counter and click **[+]** to expand the counter and reveal the counter details.



Component Monitor Name	Group	Component Type
Database Files		SQL Table Monitor

Description:

Database File Size (KB) Threshold

Warning: greater than <not set>

Critical: greater than <not set>

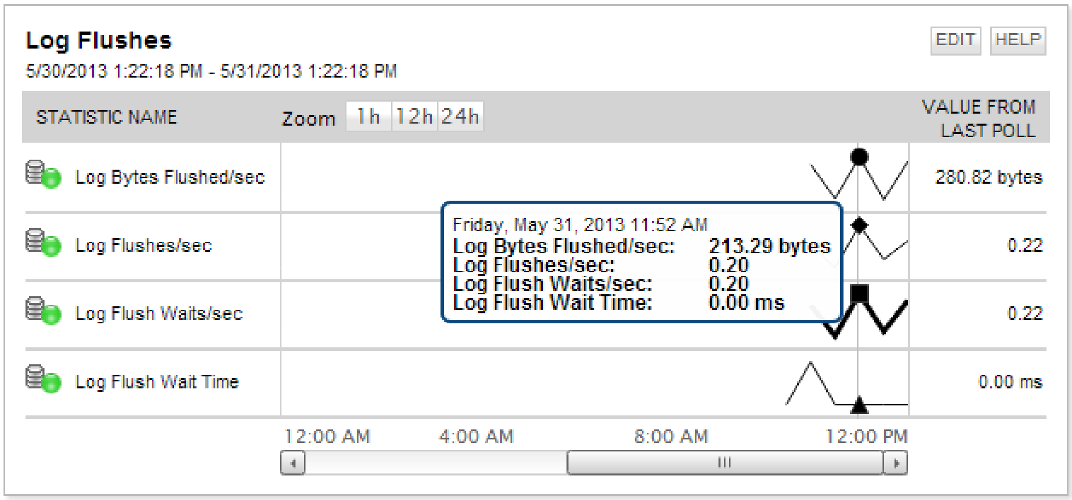
Transaction Log File Size (KB) Threshold

3. Click **Override Template** and adjust the thresholds to suit your needs.
4. When done, click **Submit**.

Log Flushes

This resource contains monitors specific to the current status of log flushes for the current database. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Log Flushes

5/30/2013 1:22:18 PM - 5/31/2013 1:22:18 PM

STATISTIC NAME	VALUE FROM LAST POLL
Log Bytes Flushed/sec	280.82 bytes
Log Flushes/sec	0.22
Log Flush Waits/sec	0.22
Log Flush Wait Time	0.00 ms

Friday, May 31, 2013 11:52 AM

Log Bytes Flushed/sec:	213.29 bytes
Log Flushes/sec:	0.20
Log Flush Waits/sec:	0.20
Log Flush Wait Time:	0.00 ms

Log Bytes Flushed/Sec	**Definition:** This performance counter shows the average log bytes flushed per second for the current database since the last data point. **Information:** The value returned helps identify trends of the transaction log. A log flush happens when data is written from the log cache to the transaction log on disk, each time a transaction happens. **Note:** This is not available in SQL 2012. **Possible problems:** The rate at which log records are written to the disk. This is the log generation rate of the application. It plays a very important role in determining database mirroring performance. This counter is in the Databases performance object. Log Bytes Flushed/sec can indicate many small transactions leading to high mechanical disk activity. **Remediation:** Look at the statements associated with the log flush and determine if the number of log flushes can be reduced. When a single transaction is used, the log records for the transaction can be bundled and a single, larger write can be used to flush the gathered log records. The mechanical intervention is significantly reduced. It is not recommended that you increase your transaction scope. Long-running transactions can lead to excessive and unwanted blocking as well as increased overhead.
Log Flushes/Sec	**Definition:** This performance counter returns the number of log flushes per second, averaged since the last collection time. **Information:** A log flush happens when a transaction is committed and data is written from the log cache to the transaction log file. The log cache is a location in memory where SQL Server stores data to be written to the log file. This is used to roll back a transaction before it is committed. Once complete, the log cache is flushed

	to the physical log file on the disk. Generally speaking, log flushes per second should parallel the number of transactions per second. **Possible problems:** If the returned value is higher than expected, check your use of explicit transactions in your queries. **Remediation:** Explicitly define the start and end of your transactions. This should reduce the number of log flushes, and reduce the impact on I/O. Also check the Log Bytes Flushed/Sec monitor.
Log Flush Waits/Sec	**Definition:** This performance counter returns the number of commits, per second, waiting for the log flush. **Information:** This value should be as low as possible. **Possible problems:** A high number of Log Flush Waits can indicate that it is taking longer than normal to flush the transaction log cache. This will slow the performance of your SQL Server. **Remediation:** Check the value of the Disk avg. Write time monitor. If the returned value is greater than 5ms, then this can indicate that there is an I/O bottleneck. Move your transaction log file to a disk drive separate from your data files. This should increase performance since there is no access competition on the same disk. Consider upgrading to RAID 10 for transaction log storage. Also, adding spindles to your RAID array should increase performance.
Log Flush Wait Time	**Definition:** This performance counter returns the total wait time (in milliseconds) to flush the log to the transaction log file. **Information:**

	On an AlwaysOn secondary database, this value indicates the wait time for log records to be hardened to disk.
Percent Log Used	**Definition:** This performance counter returns the percentage of space in the log that is in use. **Information:** This is the size of the transaction log that actively contains data in relation to the total physical size on disk of the log file. **Possible problems:** If the log file is 100% used, it will attempt to increase its size. If there is not sufficient disk space to accommodate the growth, the database will stop functioning. **Remediation:** Perform a transaction log backup to truncate the log.

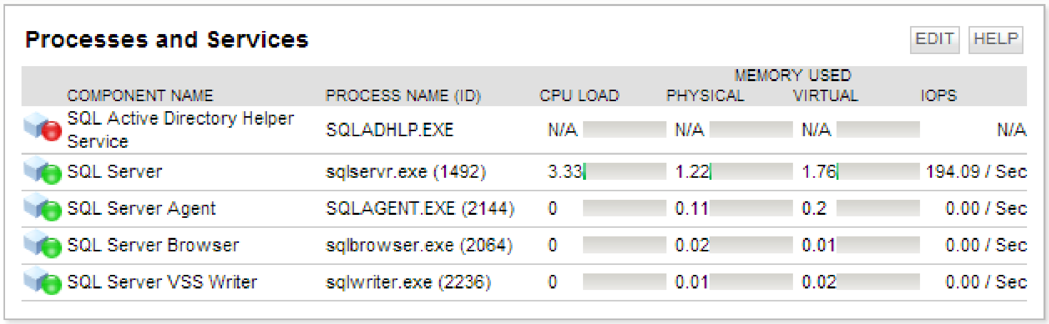
Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Processes and Services

This resource provides detailed information about the active processes and services associated with the database selected. Hovering over a component will give you detailed information about that component.



Processes and Services

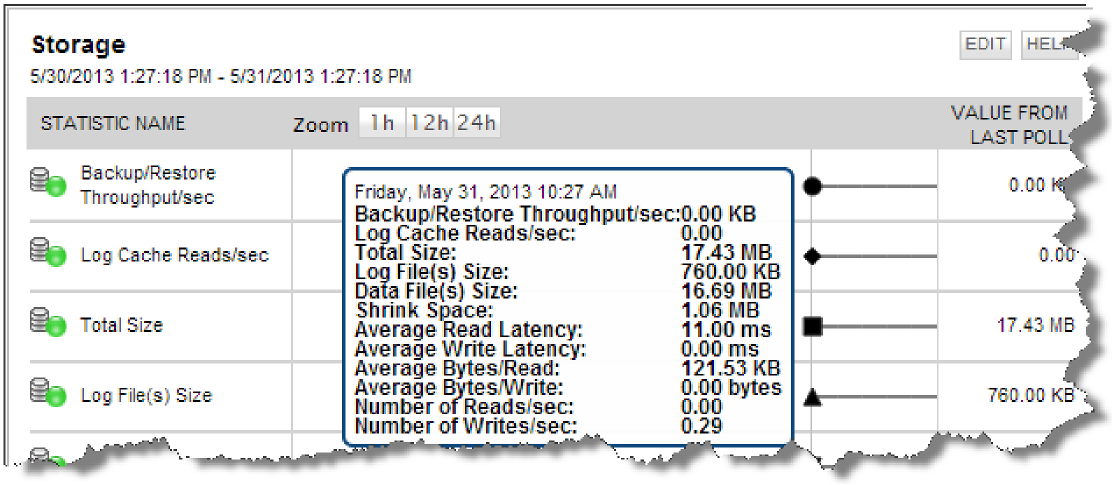
COMPONENT NAME	PROCESS NAME (ID)	CPU LOAD	MEMORY USED		IOPS
			PHYSICAL	VIRTUAL	
SQL Active Directory Helper Service	SQLADHLP.EXE	N/A	N/A	N/A	N/A
SQL Server	sqlservr.exe (1492)	3.33	1.22	1.76	194.09 / Sec
SQL Server Agent	SQLAGENT.EXE (2144)	0	0.11	0.2	0.00 / Sec
SQL Server Browser	sqlbrowser.exe (2064)	0	0.02	0.01	0.00 / Sec
SQL Server VSS Writer	sqlwriter.exe (2236)	0	0.01	0.02	0.00 / Sec

- **Component Name:** Name of the listed component.
- **Process Name:** Name of the listed process.
- **CPU Load:** Percentage of the CPU that the listed process/service is currently using.
- **Memory Used:** Memory currently being used by the listed process/service.
  - **Physical:** RAM currently being used by the listed process/service.
  - **Virtual:** Virtual memory currently being used by the listed process/service.
- **IOPS:** Current amount of Input/Output data being processed per second.

Storage

This resource contains monitors specific to the current storage status of the current database. The icons of the listed performance counters in the Statistic Name column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Storage

5/30/2013 1:27:18 PM - 5/31/2013 1:27:18 PM

STATISTIC NAME	VALUE FROM LAST POLL
Backup/Restore Throughput/sec	0.00 K
Log Cache Reads/sec	0.00
Total Size	17.43 MB
Log File(s) Size	760.00 KB

Friday, May 31, 2013 10:27 AM

Backup/Restore Throughput/sec:	0.00 KB
Log Cache Reads/sec:	0.00
Total Size:	17.43 MB
Log File(s) Size:	760.00 KB
Data File(s) Size:	16.69 MB
Shrink Space:	1.06 MB
Average Read Latency:	11.00 ms
Average Write Latency:	0.00 ms
Average Bytes/Read:	121.53 KB
Average Bytes/Write:	0.00 bytes
Number of Reads/sec:	0.00
Number of Writes/sec:	0.29

Backup/Restore Throughput/Sec	**Definition:** This performance counter shows the Read/Write throughput for backup and restore operations of a database per second. **Information:** You can measure how the performance of the database backup operation changes when more backup devices are used in parallel, or when faster devices are used. Throughput of a database backup or restore operation allows you to determine the progress and performance of your backup and restore operations. There are no thresholds associated with this performance counter.
Log Cache Reads/Sec	**Definition:** This performance counter returns the amount of Reads performed per second through the Log Manager cache.
Total Size	**Definition:** This performance counter returns the total size of the database including white space. Possible problems: Running out of storage space. Remediation: Shrink the database if free space is running low.

Log File(s) Size	**Definition:** This performance counter shows the cumulative size (in kilobytes) of all the transaction log files in the database. **Information:** Every transaction in the database is logged to the Transaction log. This is useful during recovery in the event of data loss. Information about the log file is shown under the Files section of the Database Overview page in SQL Server Management Studio. The size of this file is based on the logging level set for the database. By default, a new database uses Full Recovery Mode or the recovery mode that the model database is set up as. Transactions in the log file are only removed when a complete backup or transaction log backup is initiated. This enables pinpoint recovery for critical applications. Be aware that if the transaction log is not backed up on a regular basis, the log will grow until your disk is completely full. The transaction log should be adjusted to a reasonable size. This will depend on the number of transactions you expect, and how often you perform backups. Set the correct Autogrowth properties: The default of 10% Autogrowth for data and log files should be enough for low use databases. Conversely, a 500 MB Autogrowth rate may be better suited for a busy database, preventing a large I/O impact caused by normal Autogrowth operations. Possible problems: If the returned value occasionally drops to zero, this can indicate a problem with the following file: sys.master_files. This file is what the SQL Monitor uses to collect data. **Note:** With Simple Recovery Model, fast writing to the transaction log will trigger Autogrowth. **Remediation:** If you are performing a insert operation, you should consider switching the recovery model to Bulk Logged for the insert.

	• If you do not need pinpoint recovery, it is recommended you switch to Simple Recovery model. • If your database is configured with the Full or Bulk Logged recovery model, back up the log on a regular basis so it can be truncated to free up log space. **Note:** This action removes inactive virtual log files, but does not reduce the file size.
Data File(s) Size	**Definition:** This performance counter shows the cumulative size (in kilobytes) of all the data files in the database including any automatic growth. **Information:** Monitoring this counter is useful for determining the correct size of tempdb. The value returned from this monitor reports the size of your database (not including the Transaction log). Information about distinct primary data files and secondary data files is shown under the Files section of the Database Overview page in SQL Server Management Studio. Having a file size increase on demand is expensive and will degrade performance significantly. Autogrowth should only be used to let a database increase in size if you are running out of disk space. Autogrowth should not be used to manage your MDF size increases. Data files should be sized properly when they are initially created to meet the expected growth. This also helps avoid fragmentation, which leads to better database performance. **Possible problems:** Be wary of large and unusual increases to the size of your data files. Generally speaking, your database should be sized to minimize Autogrowth. An increase in size is expensive concerning I/O. Additionally, this will fragment your data and log files. If the returned value occasionally drops to zero, this can indicate a problem with the following file: sys.master_files. This file is what the SQL Monitor uses

	to collect data. **Remediation:** - Ensure that Full Recovery Mode is used for your databases and that you regularly backup everything, including the transaction log. - Manually add space as data increases in size. It is recommended that you have approximately one year of space available for data growth. If this is not possible, move the database to a higher capacity drive or simply upgrade the drive. - **Note:** Compressing files causes fragmentation and is therefore not recommended.
Shrink Space	**Definition:** This is essentially the white space in the database that can be reclaimed by compacting/shrinking the database. **Information:** Shrinking data files recovers space by moving pages of data from the end of the file to unoccupied space closer to the front of the file. When enough free space is created at the end of the file, data pages at end of the file can de-allocated and returned to the file system. **Possible problems:** Data that is moved to shrink a file can be scattered to any available location in the file. This causes index fragmentation and can slow the performance of queries that search a range of the index. **Remediation:** To eliminate the fragmentation, consider rebuilding the indexes on the file after shrinking.
Average Bytes/Read	**Definition:** Perfmon captures the total number of bytes sent to the disk (write) and retrieved from the disk (read) over a period of one second. **Information:**

	The Disk Read Bytes/sec and the Disk Write Bytes/sec performance counters break down the results displaying only read bytes or only write bytes, respectively.
Average Bytes/Write	**Definition:** Perfmon captures the total number of bytes sent to the disk (write) and retrieved from the disk (read) over a period of one second. **Information:** The Disk Read Bytes/sec and the Disk Write Bytes/sec performance counters break down the results displaying only read bytes or only write bytes, respectively.
Average Read Latency	**Definition:** This performance counter indicates the response time of the disk – how long it takes for a read request to get acknowledged by the disk. **Information:** The average read latency should be less than 20ms for database files.
Average Write Latency	**Definition:** Avg. Disk sec/Write is the average time, in seconds, to write data to the disk. **Information:** This analysis determines if any of the physical disks are responding slowly. **Possible problems:** If the response times are greater than .015 (15 milliseconds), then the disk subsystem is keeping up with demand, but does not have much overhead left. If the response times are greater than .025 (25 milliseconds), then noticeable slow-downs and performance issues affecting users may be occurring.
Forwarded Records/Sec	**Definition:** This performance counter returns the number of records per

second fetched through forwarded record pointers.

Information:
Heaps have one interesting feature - forwarded records. If a record needs to be updated and the updated record size is greater than the current record size, and there is no space on the page to fit the new record in, then we have two options:

1. Move the record to a new page and change all the non-clustered index records that point to it to point to the new location of the record.
2. Move the record to a new page and leave a forwarding record in the original location to point to the new location.

Possible problems:
This is one drawback of using heaps - all the extra space that is wasted with the Forwarding/Forwarded Records. Another drawback is that when scanning through the heap, forwarding records have to be followed immediately (as opposed to ignoring them and just reading the forwarded records when they are encountered). This is to vastly reduce the possibility of read anomalies such as non-repeatable reads or missed rows if a row moves before the scan point during a scan.

Remediation:
Once a data table (heap) includes forward pointers, there is only one way to get rid of them : Table Reorg. There are a few options to do this:

- The simplest one would be to create a clustered index on the data table and drop it again.
- To avoid forward pointers entirely, create a clustered index from the beginning. A clustered index keeps the data rows in its leaf node level. Therefore the data is always sorted according to the index keys and forward pointers will not be used. This is like a continuous online Reorg in this regard.

Log Cache Reads/Sec	**Definition:** This performance counter displays the number of Reads performed per second through the Log Manager cache. **Information:** There are no thresholds associated with this performance counter.
Log Cache Hit Ratio	**Definition:** This performance counter returns the percentage of Log cache reads satisfied from the Log cache. **Information:** The log cache is used to hold information that will be written to the transaction log. There are no thresholds associated with this performance counter.
Number of Reads/Sec	**Definition:** This performance counter returns the number of physical database page reads issued per second. **Information:** This statistic displays the total number of physical page reads across all databases. 80 to 90 per second is normal. Anything that is above 90 indicates indexing or memory constraint. **Possible problems:** Because physical I/O is expensive, you may be able to minimize the cost, either by using a larger data cache, intelligent indexes, and more efficient queries, or by changing the database design. **Remediation:** Attempt to tune the application so that fewer I/O operations are required. For example, perhaps I/O would be reduced if there were appropriate indexes or if the database design were de-normalized. If the applications cannot be tuned, you will need to acquire disk devices with more capacity. Compare to the Memory: Pages/sec performance counter

	to see if there is paging while the SQL Server:Buffer Manager\Page reads/sec is high. Note: Before adjusting the fill factor, at a database level, compare the SQL Server:Buffer Manager\Page reads/sec performance counter to the SQL Server:Buffer Manager\Page writes/sec counter, and use the fill factor option only if writes are a substantial fraction of reads (greater than 30 percent).
Number of Writes/Sec	**Definition:** This performance counter returns the number of data pages written per second. **Information:** Page writes per second are the total number of physical writes, including checkpoint writes, lazy writer writes, and large block writes done during index creation or bulk copy operations. **Possible problems:** If Page Writes/sec is high, this might indicate a SQL Server performance issue. **Remediation:** If these values are high, you can try to reduce the values by tuning your application or database to reduce I/O operations such as index coverage, better indexes, normalization, increasing the I/O capacity of the hardware, or by adding memory.
Total Size: Percent Change 24 Hrs	**Definition:** This value is the same as the value returned by the performance counter, Total Size but calculated as a percentage of change over the last 24 hours. **Information:** This metric allows users to be alerted when the database experiences sudden growth. **Possible problems:** A database that suddenly changes in size by a high percentage could be indicative of a problem that should be investigated.

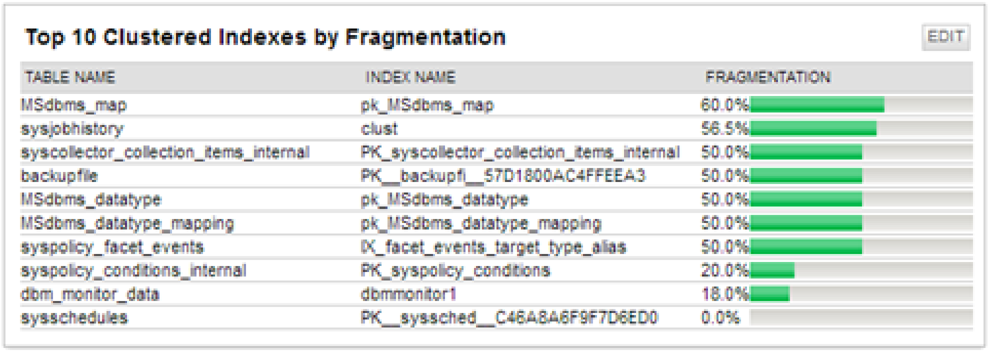
Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Top 10 Clustered Indexes by Fragmentation

A clustered index determines the physical order of data in a table. This resource displays the clustered indexes with the highest level of fragmentation. If the *Warning* or *Critical* threshold is breached, the color of the bar chart will change from green to yellow or red, respectively.



Top 10 Clustered Indexes by Fragmentation

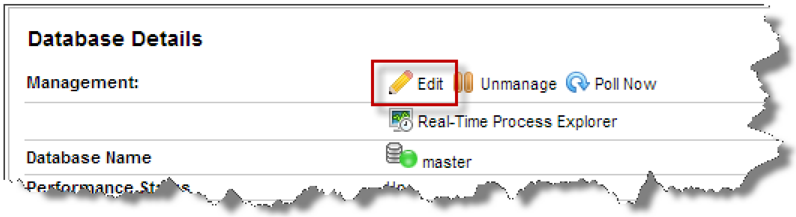
TABLE NAME	INDEX NAME	FRAGMENTATION
MSdbms_map	pk_MSdbms_map	60.0%
sysjobhistory	clust	56.5%
syscollector_collection_items_internal	PK_syscollector_collection_items_internal	50.0%
backupfile	PK__backupfi__57D1800AC4FFEEA3	50.0%
MSdbms_datatype	pk_MSdbms_datatype	50.0%
MSdbms_datatype_mapping	pk_MSdbms_datatype_mapping	50.0%
syspolicy_facet_events	IX_facet_events_target_type_alias	50.0%
syspolicy_conditions_internal	PK_syspolicy_conditions	20.0%
dbm_monitor_data	dbmmonitor1	18.0%
sysschedules	PK__syssched__C46A8A6F9F7D6ED0	0.0%

- **Table Name:** Displays the name of the listed table.
- **Index Name:** Displays the name listed index.
- **Fragmentation:** Numerically and graphically displays the percentage of fragmentation of the listed index.

The performance counter, **Top Indexes for Database**, is used to provide metrics for this resource. This counter is not listed in the resource; however, the thresholds can be changed.

To change the thresholds for the Top Indexes for Database performance counter:

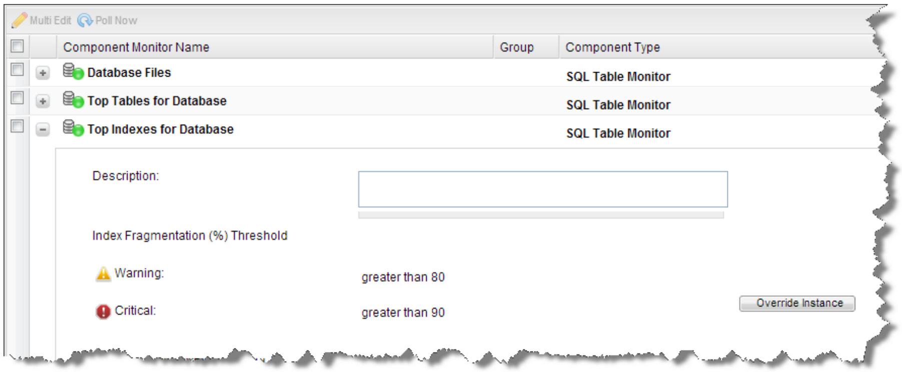
1. From the *Database Details* view, locate the **Database Details** resource and click **Edit** in the *Management* field.



Database Details

Management:	Edit Unmanage Poll Now Real-Time Process Explorer
Database Name	master
Performance Status	

2. Locate the **Top Indexes for Database** performance counter and click **[+]** to expand the counter and reveal the counter details.



		Component Monitor Name	Group	Component Type
□	+	**Database Files**		**SQL Table Monitor**
□	+	**Top Tables for Database**		**SQL Table Monitor**
□	−	**Top Indexes for Database**		**SQL Table Monitor**

Description:

Index Fragmentation (%) Threshold

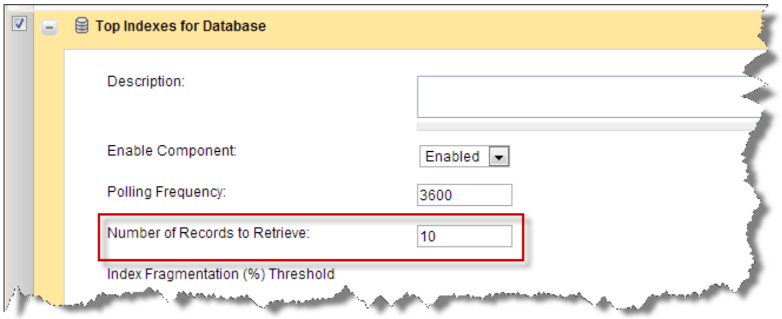
Warning:	greater than 80
Critical:	greater than 90

3. Click **Override Template** and adjust the thresholds to suit your needs.
4. When done, click **Submit**.

To change the number of tables displayed in this resource, values must be changed in two locations:

1. In the resource, click **Edit**.
2. Change the value for the field: **Maximum Number of Records to Display**, and then click **Submit**.
3. Edit the application or the template that the application is based on. To edit the template which will affect applications based on it:
   a. From the web console, navigate to **Settings > SAM Settings > Manage Templates**.
   b. Check the *AppInsight for SQL* application and click Edit in the toolbar.
   c. Expand the **Top Indexes for Database** component monitor by clicking **[+]**.

d. Change the value for the field, **Number of Records to Retrieve** and then click **Submit**.



☑ Top Indexes for Database

Description:

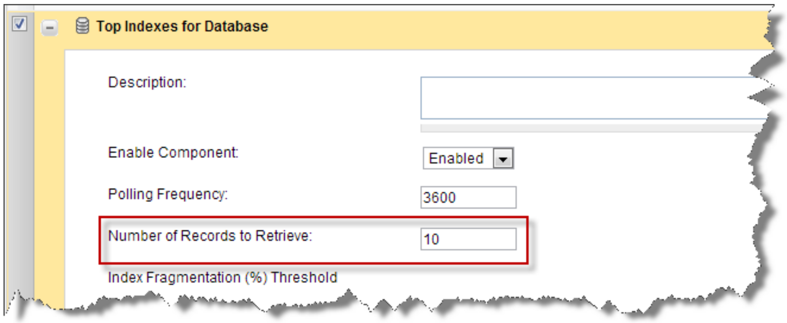
Enable Component: Enabled

Polling Frequency: 3600

Number of Records to Retrieve: 10

Index Fragmentation (%) Threshold

4. To edit the individual application:
   a. From the web console, click the *Applications* tab.
   b. From the **All Applications** resource, click an *AppInsight for SQL application* to select it.
   c. From the Application Details resource, click Edit Application Monitor.
   d. Expand the **Top Indexes for Database** component monitor by clicking **[+]**.
   e. Click **Override Template** corresponding to the row, **Number of Records to Retrieve**.
   f. Change the value for the field, **Number of Records to Retrieve** and then click **Submit**.



☑ Top Indexes for Database

Description:

Enable Component: Enabled

Polling Frequency: 3600

Number of Records to Retrieve: 10

Index Fragmentation (%) Threshold

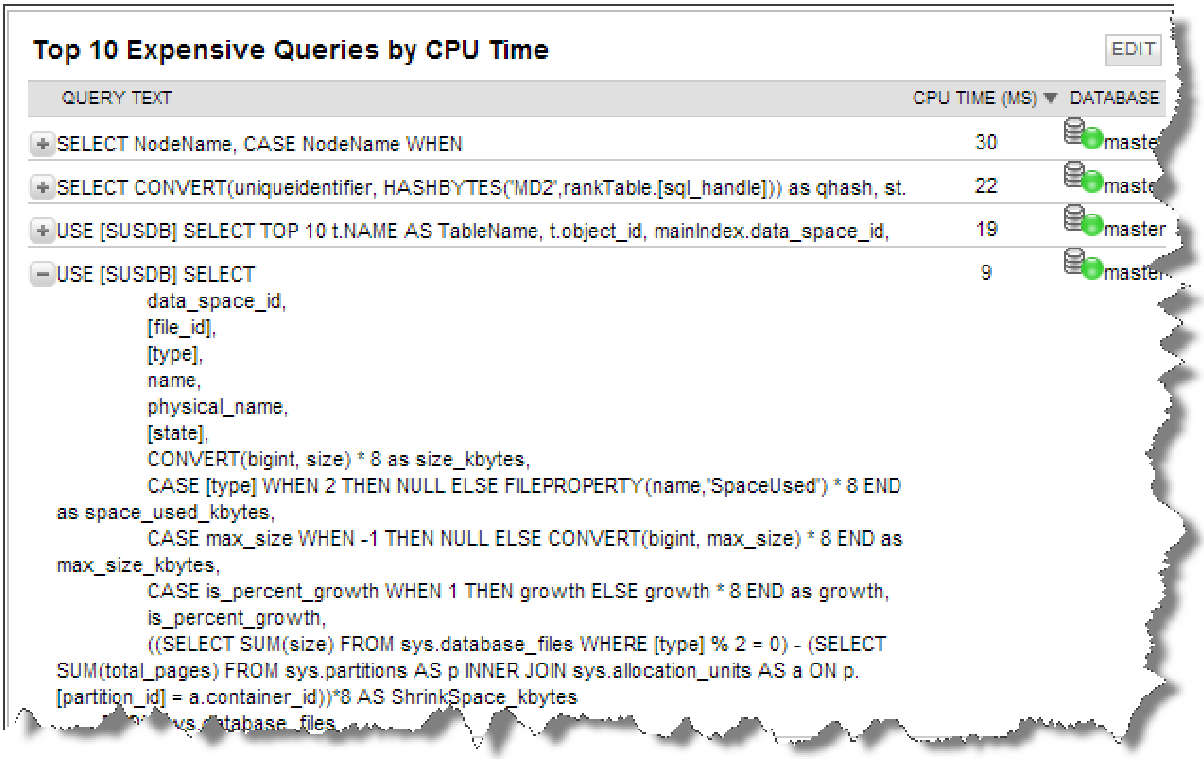
Top 10 Most Expensive Queries by CPU Time

This resource displays the top 10 most expensive queries sorted by the length of time the CPU spent processing the listed query.

When viewing this resource from the *Summary* tab on the_*Application Details* page, queries are listed for the entire SQL Server instance. Conversely, when this same resource appears on the *Database Details* view, queries are only listed for that specific database.

The data can be sorted by clicking the headers of each column. Additionally, this resource can optionally show *Top Queries by I/O* by clicking **Edit** in this resource and making the necessary adjustments.

Note: The resource is filtered based on the settings found on the *Queries* tab in *Details* view in the **Most Expensive Queries** resource.



Top 10 Expensive Queries by CPU Time

QUERY TEXT	CPU TIME (MS) ▼	DATABASE
+ SELECT NodeName, CASE NodeName WHEN	30	maste
+ SELECT CONVERT(uniqueidentifier, HASHBYTES('MD2',rankTable.[sql_handle])) as qhash, st.	22	maste
+ USE [SUSDB] SELECT TOP 10 t.NAME AS TableName, t.object_id, mainIndex.data_space_id,	19	master
− USE [SUSDB] SELECT	9	master

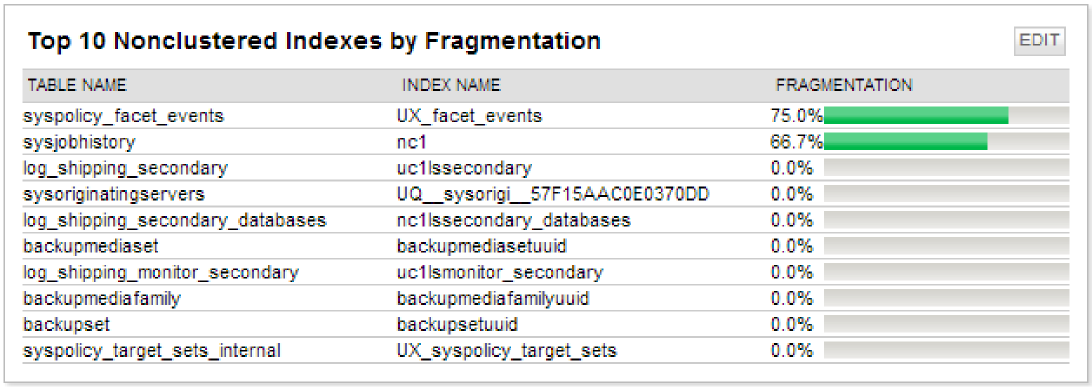
```
USE [SUSDB] SELECT
            data_space_id,
            [file_id],
            [type],
            name,
            physical_name,
            [state],
            CONVERT(bigint, size) * 8 as size_kbytes,
            CASE [type] WHEN 2 THEN NULL ELSE FILEPROPERTY(name,'SpaceUsed') * 8 END
as space_used_kbytes,
            CASE max_size WHEN -1 THEN NULL ELSE CONVERT(bigint, max_size) * 8 END as
max_size_kbytes,
            CASE is_percent_growth WHEN 1 THEN growth ELSE growth * 8 END as growth,
            is_percent_growth,
            ((SELECT SUM(size) FROM sys.database_files WHERE [type] % 2 = 0) - (SELECT
SUM(total_pages) FROM sys.partitions AS p INNER JOIN sys.allocation_units AS a ON p.
[partition_id] = a.container_id))*8 AS ShrinkSpace_kbytes
            sys.database_files
```

- **Query Text:** The text shown is a preview of the full query. To view an entire query in the list, click **[+]** next to the query listed to expand and view the query.
  Note: Clicking any listed query will take you to the *Most Expensive Queries* resource on the *Queries* tab, providing greater detail about the query.
- **CPU Time:** Display the time, in milliseconds, the listed query took to complete.
- **Database:** Displays the database with which the listed query is associated. The icon in this column shows the current state of the listed database.

The performance counter, **Top Expensive Queries**, is used to provide metrics for this resource and cannot be edited.

Top 10 Nonclustered Indexes by Fragmentation

A non-clustered index is one in which the data rows are not sorted and stored in order based on their non-clustered keys. This resource displays the non-clustered indexes with the highest level of fragmentation. If the *Warning* or *Critical* thresholds is breached, the color of the bar chart will change from green to yellow or red, respectively.



Top 10 Nonclustered Indexes by Fragmentation

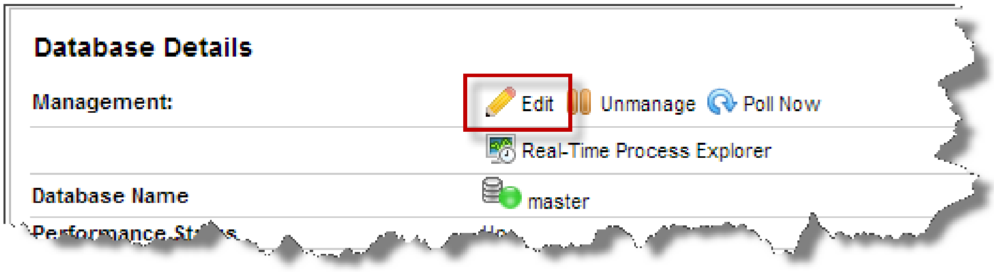
TABLE NAME	INDEX NAME	FRAGMENTATION
syspolicy_facet_events	UX_facet_events	75.0%
sysjobhistory	nc1	66.7%
log_shipping_secondary	uc1lssecondary	0.0%
sysoriginatingservers	UQ__sysorigi__57F15AAC0E0370DD	0.0%
log_shipping_secondary_databases	nc1lssecondary_databases	0.0%
backupmediaset	backupmediasetuuid	0.0%
log_shipping_monitor_secondary	uc1lsmonitor_secondary	0.0%
backupmediafamily	backupmediafamilyuuid	0.0%
backupset	backupsetuuid	0.0%
syspolicy_target_sets_internal	UX_syspolicy_target_sets	0.0%

- **Table Name:** Displays the name of the listed table.
- **Index Name:** Displays the name listed index.
- **Fragmentation:** Numerically and graphically displays the percentage of fragmentation of the listed index.

The performance counter, **Top Indexes for Database**, is used to provide metrics for this resource. This counter is not listed in the resource; however, the thresholds can be changed.

<u>To change the thresholds for the Top Indexes for Database performance counter:</u>

1. From the *Database Details* view, locate the **Database Details** resource and click **Edit** in the *Management* field.



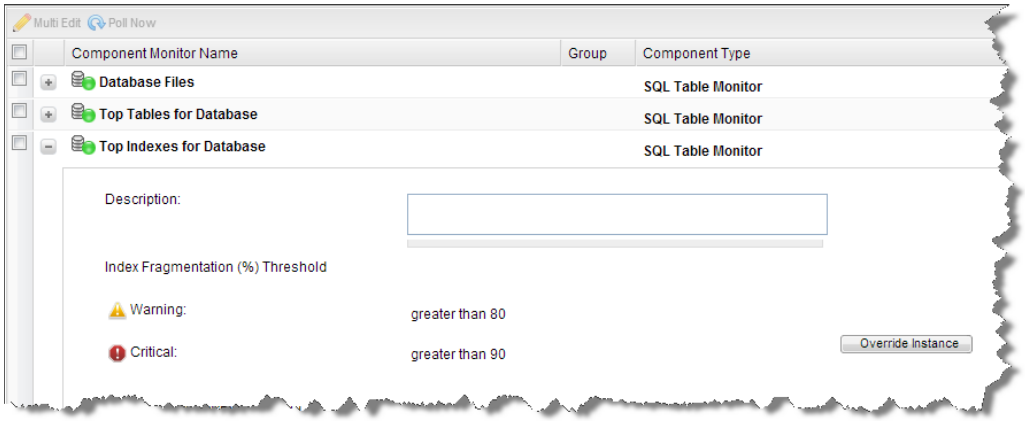
Database Details

Management: Edit Unmanage Poll Now

Real-Time Process Explorer

Database Name master

2. Locate the **Top Indexes for Database** performance counter and click **[+]** to expand the counter and reveal the counter details.



Multi Edit Poll Now

□		Component Monitor Name	Group	Component Type
□	+	Database Files		SQL Table Monitor
□	+	Top Tables for Database		SQL Table Monitor
□	−	Top Indexes for Database		SQL Table Monitor

Description:

Index Fragmentation (%) Threshold

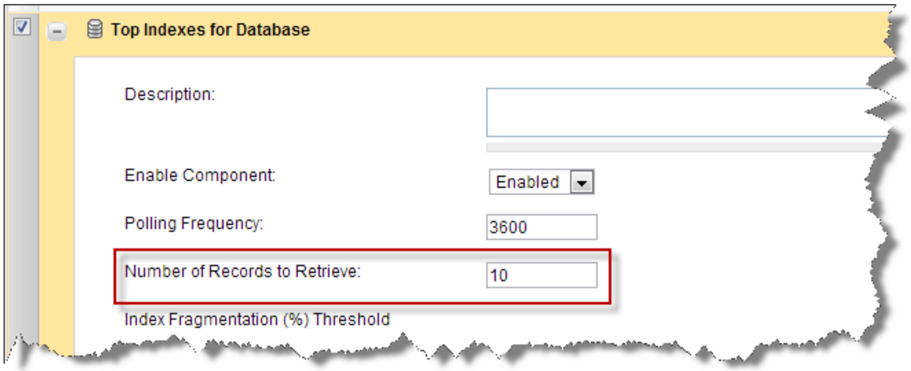
Warning: greater than 80

Critical: greater than 90

3. Click **Override Template** and adjust the thresholds to suit your needs.
4. When done, click **Submit**.

To change the number of tables displayed in this resource, values must be changed in two locations:

1. In the resource, click **Edit**.
2. Change the value for the field: **Maximum Number of Records to Display**, and then click **Submit**.
3. Edit the application or the template that the application is based on. To edit the template which will affect applications based on it:
   a. From the web console, navigate to **Settings > SAM Settings > Manage Templates**.
   b. Check the *AppInsight for SQL* application and click Edit in the toolbar.
   c. Expand the **Top Indexes for Database** component monitor by clicking **[+]**.
   d. Change the value for the field, **Number of Records to Retrieve** and then click **Submit**.



☑ − Top Indexes for Database

Description:

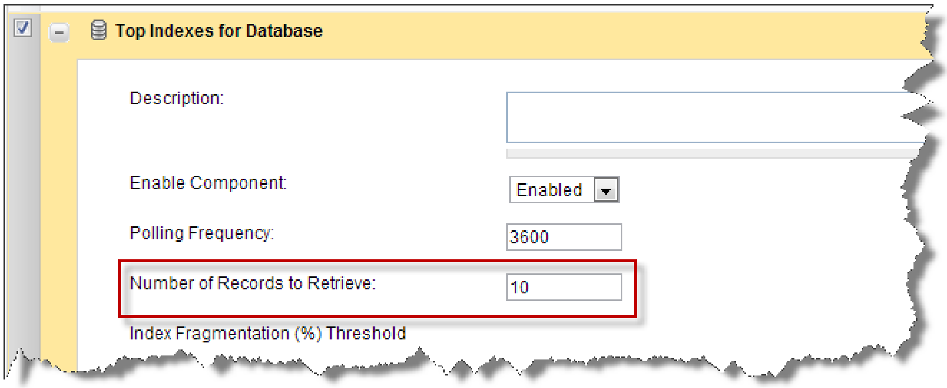
Enable Component: Enabled

Polling Frequency: 3600

Number of Records to Retrieve: 10

Index Fragmentation (%) Threshold

4. To edit the individual application:
   a. From the web console, click the *Applications* tab.
   b. From the **All Applications** resource, click an *AppInsight for SQL application* to select it.
   c. From the Application Details resource, click Edit Application Monitor.
   d. Expand the **Top Indexes for Database** component monitor by clicking **[+]**.
   e. Click **Override Template** corresponding to the row, **Number of Records to Retrieve**.
   f. Change the value for the field, **Number of Records to Retrieve** and then click **Submit**.



☑ ⊟ Top Indexes for Database

Description:

Enable Component: Enabled

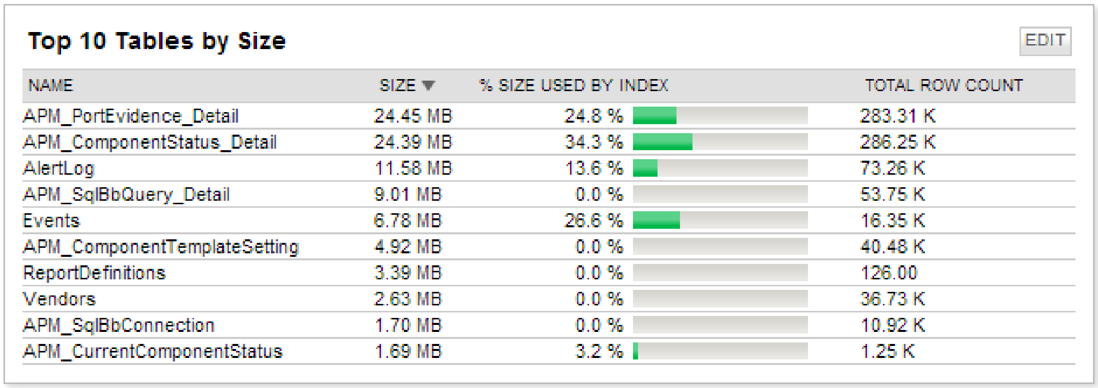
Polling Frequency: 3600

Number of Records to Retrieve: 10

Index Fragmentation (%) Threshold

Top 10 Tables by Size

This resource displays the current database tables in order based on size. If the *Warning* or *Critical* thresholds is breached, the color of the bar chart will change from green to yellow or red, respectively.



Top 10 Tables by Size EDIT

NAME	SIZE ▼	% SIZE USED BY INDEX	TOTAL ROW COUNT
APM_PortEvidence_Detail	24.45 MB	24.8 %	283.31 K
APM_ComponentStatus_Detail	24.39 MB	34.3 %	286.25 K
AlertLog	11.58 MB	13.6 %	73.26 K
APM_SqlBbQuery_Detail	9.01 MB	0.0 %	53.75 K
Events	6.78 MB	26.6 %	16.35 K
APM_ComponentTemplateSetting	4.92 MB	0.0 %	40.48 K
ReportDefinitions	3.39 MB	0.0 %	126.00
Vendors	2.63 MB	0.0 %	36.73 K
APM_SqlBbConnection	1.70 MB	0.0 %	10.92 K
APM_CurrentComponentStatus	1.69 MB	3.2 %	1.25 K

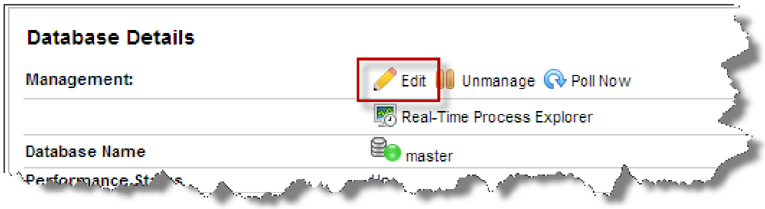
- **Name:** Displays the name of the listed table of the current database.
- **Size:** Displays the size of the listed table of the current database.

- **% Used by Index:** Displays the percentage of the table space used by the index.
- **Total Row Count:** Displays the total number of rows in the listed table of the current database.

The performance counter, **Top Tables for Database**, is used to provide metrics for this resource. This counter is not listed in the resource; however, the thresholds can be changed.

To change the thresholds for the Top Tables for Database performance counter:

1. From the *Database Details* view, locate the **Database Details** resource and click **Edit** in the *Management* field.



Database Details

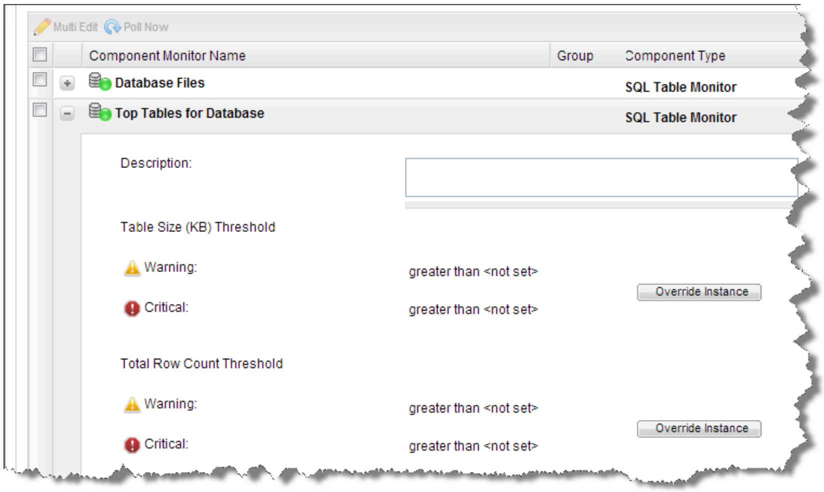
Management: Edit Unmanage Poll Now

Real-Time Process Explorer

Database Name: master

Performance Status

2. Locate the **Top Tables for Database** performance counter and click **[+]** to expand the counter and reveal the counter details.



□	Component Monitor Name	Group	Component Type
□	[+] Database Files		SQL Table Monitor
□	[−] Top Tables for Database		SQL Table Monitor

Description:

Table Size (KB) Threshold

Warning: greater than <not set>

Critical: greater than <not set>

Override Instance

Total Row Count Threshold

Warning: greater than <not set>

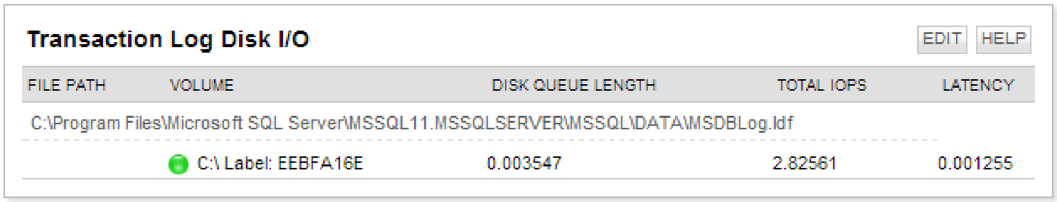
Critical: greater than <not set>

Override Instance

3. Click **Override Template** and adjust the thresholds to suit your needs.
4. When done, click **Submit**.

Transaction Log Disk I/O

This resource lists the current status of the transaction log file. The icons of the *Volume* column will change to reflect the listed volume's current state. Clicking the volume will take you to the *Volume Details* page.



Transaction Log Disk I/O

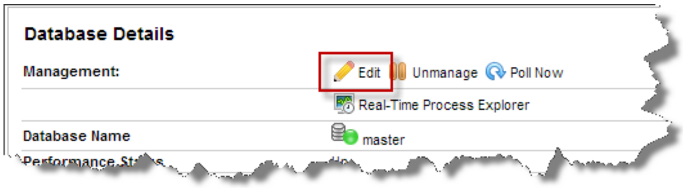
FILE PATH	VOLUME	DISK QUEUE LENGTH	TOTAL IOPS	LATENCY
C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\MSDBLog.ldf				
	● C:\ Label: EEBFA16E	0.003547	2.82561	0.001255

- **File Path:** Displays the full path where the transaction log file resides.
- **Volume:** Displays the volume where the transaction log file resides. Clicking on the volume name will take you to the Volume Details View.
- **Disk Queue Length:** Displays the current disk queue length for the listed volume as seen on the Volume Details view. Queue lengths that are high for a sustained period of time are indicative of disk I/O performance issues that can negatively impact the database.
- **Total IOPS:** Displays the current total read and write IO operations occurring on the drive where the database or transaction log file resides. High I/O may be normal for some databases, but excessively high I/O will lead to higher queue lengths, meaning there are more requests for disk reads or writes than the storage subsystem can handle.
- **Latency:** Displays a measurement of disk latency occurring on the disk where the database or transaction log file resides. High volume latency values indicate sluggish database performance.

The performance counter, **Database Files**, is used to provide metrics for this resource. It is used in combination with volume polling. This counter is not listed in the resource; however, the thresholds can be changed.

To change the thresholds for the Database Files performance counter:

1. From the *Database Details* view, locate the **Database Files** resource and click **Edit** in the *Management* field.



Database Details

Management:

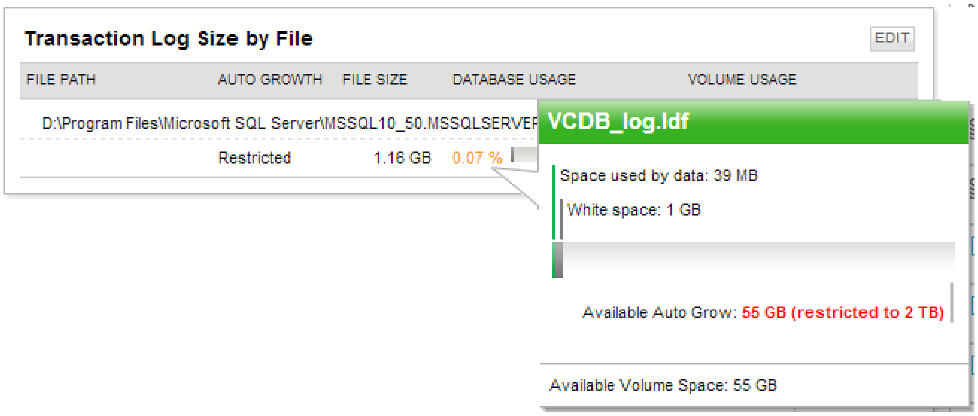
Database Name: master

Performance St..s

2. Locate the **Database Files** performance counter and click **[+]** to expand the counter and reveal the counter details.
3. Click **Override Template** and adjust the thresholds to suit your needs.
4. When done, click **Submit**.

Transaction Log Size by File

This resource lists the current status of transaction log files as sorted by size. Hovering the mouse over a listed percentage will reveal a tooltip with detailed information about the item listed. Clicking the percentage will take you to the *Database Details* page. If the *Warning* or *Critical* thresholds is breached, the color of the bar chart will change from green to yellow or red, respectively.



Transaction Log Size by File

FILE PATH	AUTO GROWTH	FILE SIZE	DATABASE USAGE	VOLUME USAGE
D:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVEF				
	Restricted	1.16 GB	0.07 %	

VCDB_log.ldf

Space used by data: 39 MB

White space: 1 GB

Available Auto Grow: 55 GB (restricted to 2 TB)

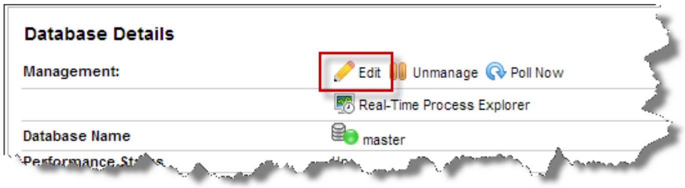
Available Volume Space: 55 GB

- **File Path:** Displays the complete file path of the listed transaction log.
- **Auto-Growth:** Displays the current Auto-growth setting.
- **File Size:** Displays the current size of the listed transaction log.
- **Database Usage:** This resource displays the amount of space consumed as a percentage of whatever restrictions are in place for the file. For example:
  - If the transaction log is allowed to grow unrestricted, then it is limited only by the amount of free space on the volume.
  - If the transaction log is allowed to grow to a maximum size, the percentage is calculated based on this restricted size limit.
  - If the transaction log has a fixed size, the percentage used is calculated based on the total size of the transaction log file and the amount of remaining white space.
- **Volume Usage:** Displays the amount of space the listed file is consuming on the volume where it is located.

The performance counter, **Database Files**, is used to provide metrics for this resource. This counter is not listed in the resource; however, the thresholds can be changed.

To change the thresholds for the Database Files performance counter:

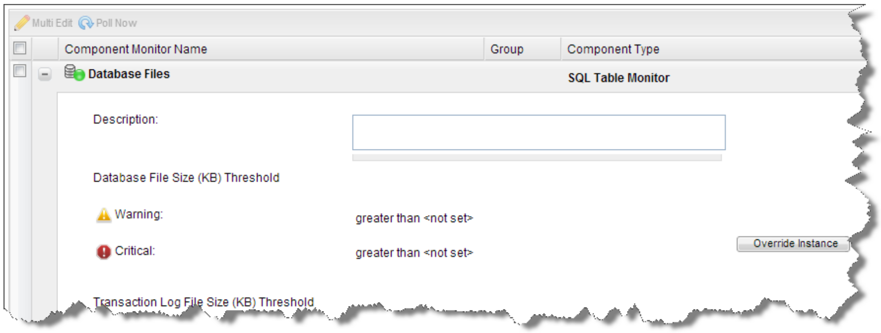
1. From the *Database Details* view, locate the **Database Files** resource and click **Edit** in the *Management* field.



Database Details

Management:	Edit Unmanage Poll Now
	Real-Time Process Explorer
Database Name	master
Performance St...s	

2. Locate the **Database Files** performance counter and click **[+]** to expand the counter and reveal the counter details.



□	Component Monitor Name	Group	Component Type
□	⊟ **Database Files**		**SQL Table Monitor**

Description:

Database File Size (KB) Threshold

Warning: greater than <not set>

Critical: greater than <not set>

Transaction Log File Size (KB) Threshold

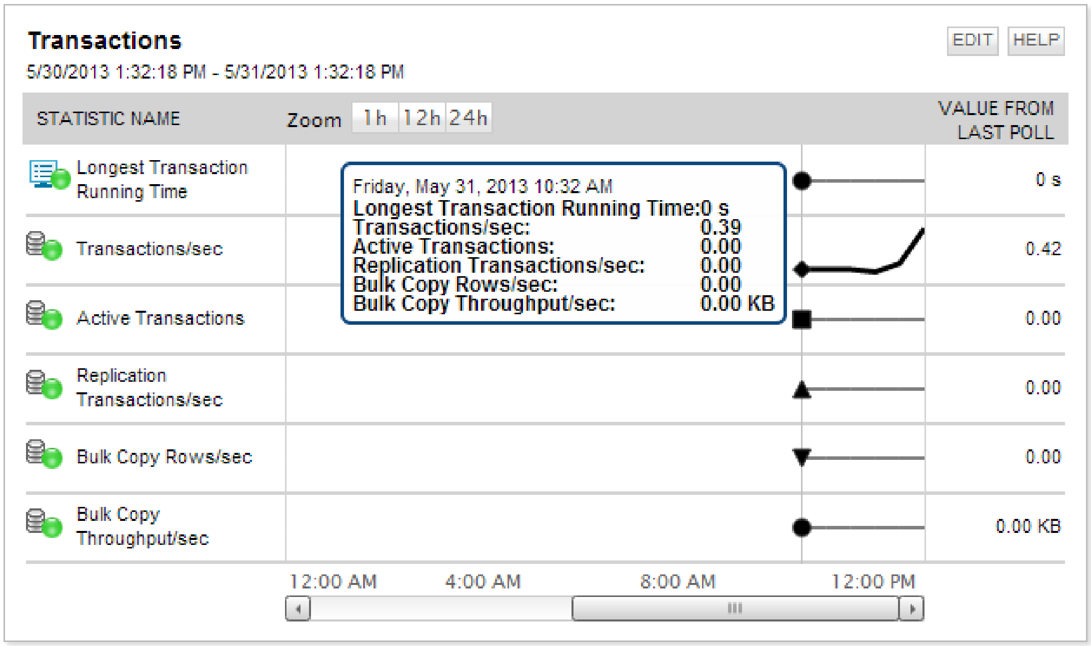
3. Click **Override Template** and adjust the thresholds to suit your needs.
4. When done, click **Submit**.

Transactions

A transaction is a unit of work that is performed against a database. In other words, if you are creating a record, for example, then you are performing transaction on the table. It is important to control transactions to ensure data integrity and to handle database errors.

This resource contains monitors specific to the current status of transactions. The icons of the listed performance counters in the *Statistic Name* column will change to reflect the listed counter's current state. Clicking any performance counter in the *Statistic Name* column will take you to the *Performance Counter Details* page for that counter.

If the value of a listed monitor crosses the *Warning* threshold, the chart for the listed monitor will display a yellow background. If the value of a listed monitor crosses the *Critical* threshold, the chart for the listed monitor will display a red background. Hovering over any time period within the chart will display detailed information about that time period in the tooltip. Hovering over a counter will give you detailed information about that counter.



Transactions

5/30/2013 1:32:18 PM - 5/31/2013 1:32:18 PM

STATISTIC NAME	VALUE FROM LAST POLL
Longest Transaction Running Time	0 s
Transactions/sec	0.42
Active Transactions	0.00
Replication Transactions/sec	0.00
Bulk Copy Rows/sec	0.00
Bulk Copy Throughput/sec	0.00 KB

```
Friday, May 31, 2013 10:32 AM
Longest Transaction Running Time:0 s
Transactions/sec:                0.39
Active Transactions:             0.00
Replication Transactions/sec:    0.00
Bulk Copy Rows/sec:              0.00
Bulk Copy Throughput/sec:        0.00 KB
```

Longest Transaction Running Time	**Definition:** This performance counter displays the length of time, in seconds, the transaction that has been running the longest, has been active. **Information:** Transactions that run longer than others use more resources. They can be used to track down procedures and calls that are taking longer than expected by identifying the specific transaction(s). **Possible problems:** Long running transactions can prevent truncation of transaction logs. This can cause the transaction log files to grow until they consume all available physical disk space shutting down access to the database.

	Remediation: Check the functionality of the query and/or redesign the long running transaction.
Transactions/ Sec	**Definition:** This performance counter returns the number of transactions started for the database, per second. **Information:** Transaction rate is affected by general system performance and resource constraints, such as I/O, number of users, cache size, and the complexity of requests. **Possible problems:** A high rate of transactions can indicate that some transactions are not completing. **Remediation:** Using stored procedures in transactions speeds transaction processing because SQL code in stored procedures runs locally on the server. The transactions can finish much faster without the network traffic required by the transactions.
Active Transactions	**Definition:** This performance counter returns the number of active transactions for the database. **Information:** During an active transaction, no other transaction can modify the data held by the active transaction. While you are modifying data, an exclusive lock is held on that data. Conversely, you cannot read data held by another transaction because another transaction holds an exclusive lock on the data that prevents it from being read. The returned value for this monitor should be less than 200. **Possible problems:** A high number of active transactions can indicate that certain

	transactions are never completing. **Remediation:** Keeping a low level of active transactions assists in keeping requested data accessible.
Repl. Trans. Rate **(Replication Transactions Rate)**	**Definition:** This performance counter returns the number of transactions read out of the transaction log and sent to the distribution database **Information:** SQL Server Overall Replication Performance Analysis provides a graph of several SQL Server overall replication performance counters for the specified server during the specified time period. Performance counter data is averaged by the hour. The graph displays the following replication performance counters: • SQLServer:Databases / Replication Pending Xacts (SQL Server; Scale=1\100) • SQLServer:Databases / Replication Transaction Rate (SQL Server; Scale=1\100) • SQLServer Replication Published DB / Replicated Transactions Per Second
Bulk Copy Rows/sec	**Definition:** This monitor returns the number of rows bulk copied per second. **Information:** Measure the number of rows coming into the database. Optimization yields a higher number. See The Data Loading Performance Guide http://technet.microsoft.com/en-us/library/dd425070(v=SQL.100).aspx **Possible problems:** BCP to bulk load data over the network, having the correct throughput configured is crucial.

Bulk Copy Throughput/ Sec	**Definition:** This monitor returns the amount of data bulk copied (in kilobytes) per second. **Possible problems:** Overall throughput will mostly be limited by your I/O subsystem.

Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart. The time and date displayed at the top of the resource shows the amount of data currently loaded into the chart. This can be changed by clicking **Edit** from within the resource.

Note: The *Value from Last Poll* column shows the most recent data available for the listed statistic.

Performance Counter Details View

This view is equivalent to the Component Details view in a typical SAM application. This view can be accessed by clicking any one of the performance counters found in resources from the *Summary* tab and *Database Details* view. The default resources for this view are:

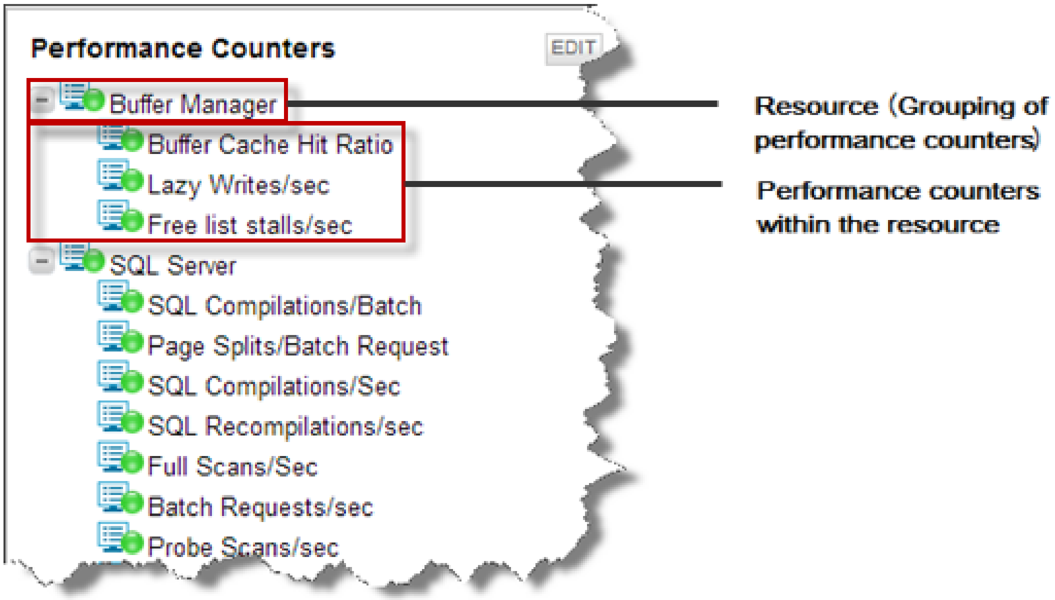
- Performance Counters
- Performance Counter Details
- Performance Counter History

Performance Counters

This resource displays all the available performance counters associated with the current application or database.

The list of performance counters displayed will change depending on how you enter this view. For example, if you enter this view by drilling down from the database level, just the counters associated with that database will be displayed. If you enter this view by drilling down from the application view, then only the performance counters associated with the application will be shown.

Clicking any performance counter in the list will display detailed information about that performance counter in the *Performance Counter Details* resource.



Performance Counters EDIT

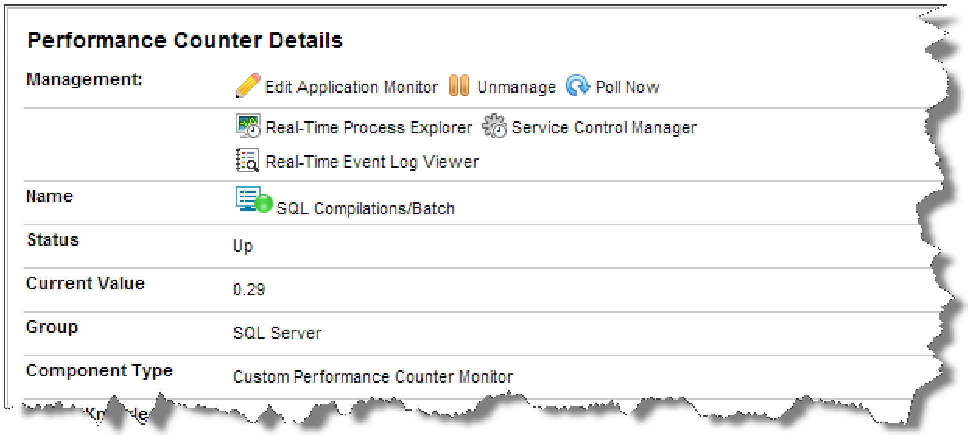
- Buffer Manager
  - Buffer Cache Hit Ratio
  - Lazy Writes/sec
  - Free list stalls/sec
- SQL Server
  - SQL Compilations/Batch
  - Page Splits/Batch Request
  - SQL Compilations/Sec
  - SQL Recompilations/sec
  - Full Scans/Sec
  - Batch Requests/sec
  - Probe Scans/sec

Resource (Grouping of performance counters)

Performance counters within the resource

Performance Counter Details

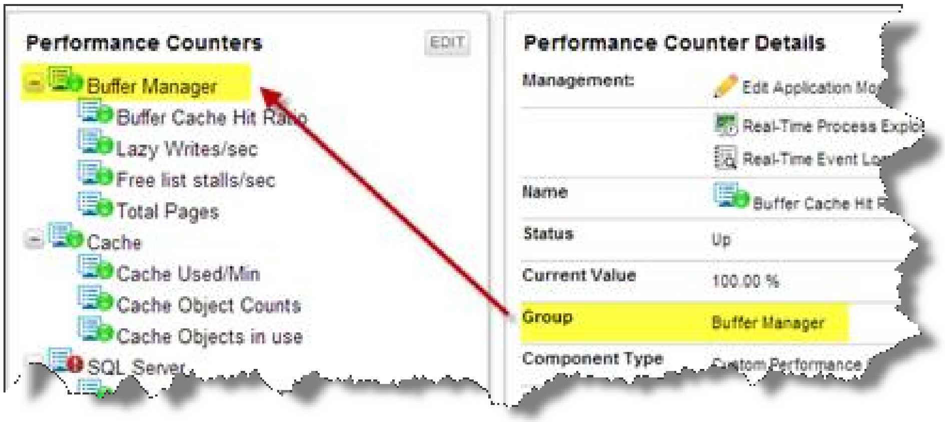
This resource provides tools and detailed information about the current performance counter.



Performance Counter Details

Management:	Edit Application Monitor Unmanage Poll Now
	Real-Time Process Explorer Service Control Manager Real-Time Event Log Viewer
Name	SQL Compilations/Batch
Status	Up
Current Value	0.29
Group	SQL Server
Component Type	Custom Performance Counter Monitor

- **Management:** This section lists the tools that are available.
- **Name:** Name of the current performance counter.
- **Status:** Status of the current performance counter.
- **Current Value:** The value of the current performance counter.
- **Group:** Displays the group, or resource, this performance counter belongs to as it would be viewed on the *Application Details* view. For example:



Performance Counters EDIT

- Buffer Manager
  - Buffer Cache Hit Ratio
  - Lazy Writes/sec
  - Free list stalls/sec
  - Total Pages
- Cache
  - Cache Used/Min
  - Cache Object Counts
  - Cache Objects in use
- SQL Server

Performance Counter Details

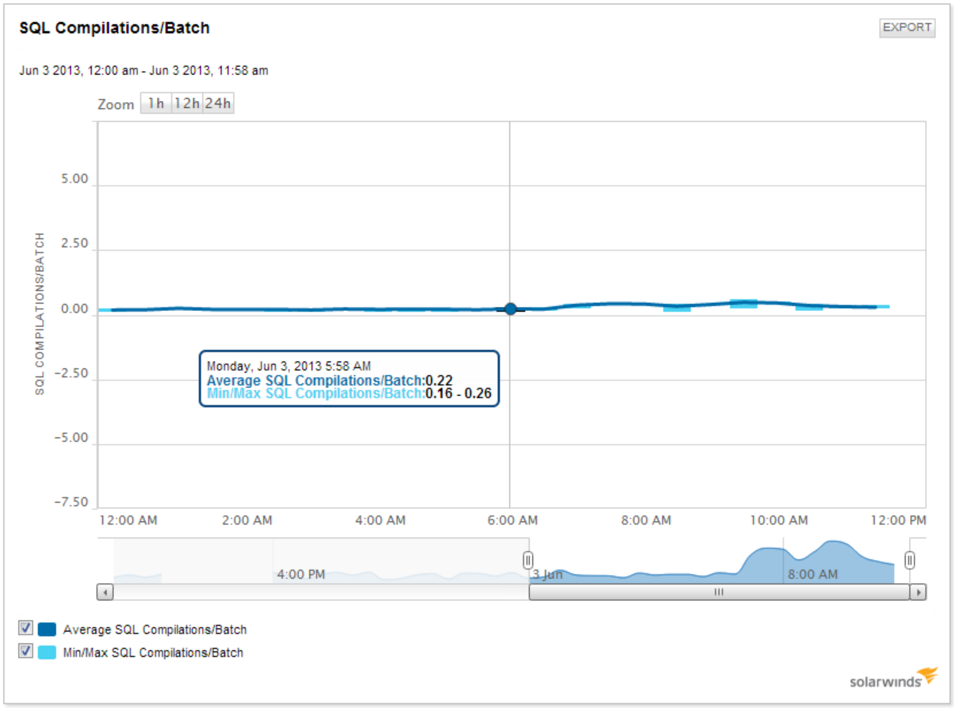
Management:	Edit Application Mo Real-Time Process Explo Real-Time Event Lo
Name	Buffer Cache Hit R
Status	Up
Current Value	100.00 %
Group	Buffer Manager
Component Type	Custom Performance

- **Component Type:** The type of the current performance counter.
- **Expert Knowledge:** This section provides information about the performance counter, including possible problems that may occur as a result of the returned value being abnormal. Remediation information is also provided, where applicable, to help you resolve typical issues that may be occurring if abnormal values are returned.

Performance Counter History

This resource graphically details the value history of the current performance counter. Hovering over any time period within the chart will display detailed information about that time period in the tooltip.

The colors of the main chart correspond to the labels in the legend. You can choose which data to display by using the checkboxes in the legend.


SQL Compilations/Batch
EXPORT
Jun 3 2013, 12:00 am - Jun 3 2013, 11:58 am
Zoom
1h
12h
24h
SQL COMPILATIONS/BATCH
5.00
2.50
0.00
-2.50
-5.00
-7.50
Monday, Jun 3, 2013 5:58 AM
Average SQL Compilations/Batch:0.22
Min/Max SQL Compilations/Batch:0.16 - 0.26
12:00 AM
2:00 AM
4:00 AM
6:00 AM
8:00 AM
10:00 AM
12:00 PM
4:00 PM
3 Jun
8:00 AM
☑ Average SQL Compilations/Batch
☑ Min/Max SQL Compilations/Batch
solarwinds

Zooming

You can have the chart show a pre-determined time period of data by clicking on any one of the three **Zoom** buttons in the head of the *Zoom* column. Alternatively, you can have the chart show a specific date range by dragging the mouse over an area of the chart or manipulating the sliders of the lower chart.