# Table of Contents

**SQL Sentry** ........................................................................................................................................ 3
  **SQL Sentry Portal** ......................................................................................................................... 3
    Using SQL Sentry Portal & SentryOne Monitor ................................................................. 3
    SQL Sentry Portal & SentryOne Monitor Health ............................................................ 8
    SQL Sentry Portal & SentryOne Monitor Dashboards .................................................. 13
    SQL Sentry Portal & SentryOne Monitor Top SQL ......................................................... 24
    SQL Sentry Portal & SentryOne Monitor Alerts .......................................................... 36

**SentryOne Monitor** ........................................................................................................................... 44
  SentryOne Monitor Release Notes ....................................................................................... 44
  Getting Started with SentryOne Monitor ........................................................................... 47
  Monitor & Portal Custom Dashboards ................................................................................. 55
Overview & Navigation

- SentryOne Monitor is accessed through a browser at monitor.sentryone.com.
- SentryOne Portal is a self-hosted feature for SQL Sentry which has a similar UI experience. See the Getting Started with SentryOne Portal article for information on accessing it.

Default Home View

When you first log into SentryOne Monitor, the home view displays the overall health of the monitored environment.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Navigation bar</td>
<td>Options for products, date selector, providing feedback, accessing documentation, and user profile information. Full details are provided under the Navigation Bar section below.</td>
</tr>
<tr>
<td>2. Sidebar</td>
<td>The sidebar has areas for overview options (Health Overview &amp; Alerts) and the monitored Targets. Full details are provided under the Sidebar section below.</td>
</tr>
</tbody>
</table>
| 3. Health Overview | Health Overview is the default view when opening SentryOne Monitor.  
  - Health Score shows a summary of overall health scores for the monitored environment.  
  - Alerts by Severity breaks them into Critical, High, Medium, or Low (with a count for the number of alerts in each severity category).  
  - Alerts by Tag breaks them into categories such as Network, CPU, Memory, Disk, or Other (with a count for the number of alerts in each tag). |
### Alerts

A table of the most recent alerts that have been triggered across the monitored environment.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Alerts</td>
<td>A table of the most recent alerts that have been triggered across the monitored environment.</td>
</tr>
</tbody>
</table>

### Navigation Bar

This is the default view for the navigation bar:

- **Sidebar toggle**: The bars (or hamburger) button toggles the visibility of the left sidebar (the overview options and targets information).
- **SentryOne logo**: Selecting the SentryOne logo returns you to the home (*Health Overview*) view.
- **Product drop-down**: The SentryOne product drop-down allows you to switch between Monitor, Test, and Document.
  - **Note**: Each product has their own licensing requirements.
- **Date Selector**: This is disabled on the default home view due to the nature of the health score calculation. On other views, such as Dashboards, it allows you to select a time frame for which to display data on the view.
  - **Note**: Use to pause live data, to jump to current data, and to view live data where applicable.
- **Feedback**: Select the feedback button to submit ideas about any issues, features you’d like to see, or suggested changes.
- **Documentation**: Select the documentation button to open the Monitor documentation in SentryOne Docs.
- **Profile**: Select the user profile button to change your password or log out of Monitor.

### Sidebar

The sidebar has five main areas:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1. Overview Options    | • **Health** displays information about events and their severity or tag for a variety of views, such as all server, targets, or groups.  
                             • **Alerts** displays a log of alerts that have been triggered in your environment as well as some configuration options for email. |
| 2. Targets Filtering Options | Start typing a target name in the filter to get to a filtered list of targets or a specific target. Use the **Show Hierarchy** switch to toggle the list of targets between an alphabetically ordered flat list and a list grouped by sites. |
| 3. Targets             | Targets provides a list of all **targets** in your monitored environment.     |
| 4. Unwatched Servers   | Unwatched servers are targets that are not currently being monitored by SentryOne. See the **Unwatched Servers** section below for more details. |
| 5. Version             | The current version of SentryOne Monitor published to monitor.sentryone.com (or installed for SentryOne Portal). See the **Release Notes** article for more details. |

### Unwatched Servers

**Applies to the following products and features:** The on-premises **SentryOne Portal** feature for SQL Sentry. See the **Getting Started with SentryOne Portal** article for more details.

In SentryOne Portal, the bottom of the sidebar contains a group for **Unwatched Servers** (similar to **Inventory view**). Unwatched servers are not currently being monitored by SentryOne. Use the **SentryOne client** to watch a **target** (you may need to **purchase additional licenses**).
Note: A banner appears an individual unwatched targets across the SentryOne Portal views.

Feature Menu

When you select a target from the sidebar, the target **Health** view opens by default. From a target view like Health, the **Navigation Bar** toggles to a feature menu.

The following options are available for individual targets:

1. Health
2. Dashboards
   - Dashboards has a drop-down menu to make additional selections. **Performance Analysis** is the default dashboard view. Any dashboards listed below **Performance Analysis** are **Custom Dashboards**. An option to add a new custom dashboard is also available from this menu.
3. Top SQL
4. Blocking (**SentryOne Portal** only)
5. Deadlocks (**SentryOne Portal** only)

Distributed Databases

**Applies to the following products and features**: The on-premises **SentryOne Portal** feature for SQL Sentry. See the [Getting Started with SentryOne Portal](#) article for more details.

If you are using SentryOne Portal for SQL Sentry and have multiple SentryOne databases, you can configure your portal to display information for all your SentryOne databases through a single service. See the
Distributed Databases article for more information.
Health Overview

Overview

The **Overview** tab is displayed by default. As indicated in the name, this view is showing the overall health, and 10 most recent alerts for all monitored targets in your environment.

Health Score Calculation

The **Health Score** (shown above) is calculated by incorporating open alerts and their associated severity. It uses a fixed 5-day window in the calculation, which is why the date selector is disabled for this view. A high health score (with a maximum of 100 possible) is an indicator of good health.

The points per severity level are:

- Low - 1 point
- Medium - 3 points
- High - 6 points
- Critical - 9 points
Note: The weight of an open alert diminishes with its age. The score being presented is 100 minus the sum of the weighted value of open alerts. Closed alerts are not factored into the score. Alerts can be closed in the Events Log in the SentryOne configuration client.

Alert Details
Select the (chevron-right icon) to the right of the count to display the individual details about each logged alert on any of the health views:

**EHO for Sites and Groups**

The Environment Health Overview (EHO) is available for sites and groups in SentryOne Portal. Use the Show Hierarchy switch to toggle the list of targets between a flat list and a list grouped by sites.

Expand a site to view the groups within it and select a group to view a group EHO.
**Note:** The Site and Group icons and labels note which type of overview you are viewing.

### All SQL Servers View

Select **All SQL Servers** from the sidebar to display the overall health view for all SQL Servers in your monitored environment.

![All SQL Servers View Diagram](image)

**Note:** **Uptime** is a measure of the percentage of time that SentryOne Monitor is able to connect to the server to collect data.

The **All SQL Servers** view provides options for viewing overall health through **Alerts by Severity, Alerts by Tag, and Wait Time / Session**. These views are similar to the target health views described below, but they include the overview information for all monitored SQL Servers.

### Target Health View

When selecting a target, the **Health** view is the default view. This screen displays wedges for **Alerts by Severity, Alerts by Tag, and Wait Time / Session**. The default view is **Alerts by Severity**.

#### Alerts by Severity

In the example below, the **Alerts by Severity** widget is displaying **1 Critical** and **316 Medium** severity alerts. In the **Alerts** table, you can see 1 alert with a severity of **critical**, and 3 rows of medium alerts (78, 237, and 1) totaling 316. This is how the number in the color-coded alert is calculated.
Alerts by Tag example

Note: Selecting the blue hyperlinked target name on the left provides an option to switch to the Health, Dashboard, or Top SQL, Blocking, or Deadlocks view for that target.

Alerts by Tag

The Alerts by Tag view works by incorporating alerts to calculate the scores and associate them to tags on the alerts (Network, CPU, Memory, Disk, and Other). The score for these views is based on the severity of all open alerts within the selected time period. A high health score (with a maximum of 100 possible) is an indicator of good health.
Wait Time / Session

Wait Time / Session relies on wait stats data collected by SentryOne. These wait stats are broken down by major resource category (Network, CPU, Memory, Disk, Other). In this view, the health score worsens as waits increase. A low ms/session score is an indicator of good health.

Each category is calculated independently. The displayed value for a category is calculated as the waits for the category divided by the user sessions. The values for waits and user sessions are based on the most recent values present in the selected time frame. Each value is then rounded and displayed as the category’s value. The overall score is the sum of the five rounded categories.

The Other category is for other important wait types that either affect performance in more than one major category, or can’t be directly attributed to any category with absolute certainty, such as backups and parallelism respectively.

Related Targets

As seen in the image above, there is a list of Related Targets (1 SQL Server and 1 Windows Server in this example). When there are multiple instances on a SQL Server, they will be listed as a related target, as will the Windows Server where the SQL Server is installed.
Applies to the following products and features: The SentryOne Monitor product and the on-premises SentryOne Portal feature for SQL Sentry.

The Dashboards view displays the performance charts for monitored targets. Select a target from the sidebar, then select Dashboards from the feature navigation menu at the top.

Performance Analysis Dashboard

Select (chevron-down) to expand the list of available dashboards. The Performance Analysis option from the dashboard drop-down menu is the default dashboard view.

Dashboard Navigation

Within the Dashboard view, the options and information displayed include:

1. A drop-down menu to switch between monitored targets.
2. A pause button (which flips to a play button) to toggle the view of live data.
3. A date selector.
4. Default Dashboard | Performance Analysis for the default dashboard or Custom Dashboard | for a Custom Dashboard.

Note: In the SentryOne Portal feature for SQL Sentry, versions 2021.1 and later, the Custom Dashboards have been replaced by Custom Charts and are not part of the Performance Analysis Dashboard section.
Dashboard Options

Date Selector

Use the date selector to use an available Range or define a Custom Range of time for data to view on the dashboard. Use the Days, Weeks, and Months options to select an entire day, week, or month range at once.

Chart Details

Hover over a point in a chart to view additional details.

Note: The chart details tool-tips vary by chart.
Filter Dots

Use the filter dots (e.g. Disk, Other, Memory, CPU, and Network in the image below) to customize what appears on the chart from the available options:

The same chart with Disk filtered out:

Zoom

Highlight an area on the chart for options to Zoom the Dashboard charts into that time selection.

Jump To

Highlight an area on the chart for options to jump to other feature views (e.g. Health, Dashboard, Top SQL, Blocking, or Deadlocks) for the selected time period. For example, use this troubleshoot performance issues by correlating a spike to a SQL query.
Some charts, such as SQL Server Activity and SQL Server Memory may have additional metrics available in the legend and filter dots. Use the arrows to scroll through them.

Dashboard Charts

Note: The available charts displayed vary by target type (e.g. SQL Server vs. Azure SQL database). For example, an Azure SQL Managed Instance target will have SQL Server charts, but not Windows charts displayed.

Windows

Network

The Network chart displays the total network traffic on the server as well as the network utilization on each of the adapter present on the monitored target.

CPU Usage

The CPU Usage chart displays the total CPU Usage for the server as well as information on context switching, user time, kernel time, and more. The total processor time percentage across all processors on the server. A sustained value greater than 80 percent generally indicates a CPU bottleneck.

System Memory

The System Memory chart displays information about the amount of memory being used by different processes on the server as well as page faults and page file usage.

SQL Server

The amount of physical memory used by each SQL Server. Important for determining whether available memory is being used effectively, and whether there’s memory contention between multiple instances on the same server.
File cache

The amount of physical memory currently allocated to the system file cache.

Other

The amount of physical memory used by all processes on the server other than SQL Server or SSAS.

Disk I/O

The Disk I/O chart displays the read and write latency for each of the physical disks on the server.

ms/Read

The average time in milliseconds each physical disk read is taking.

Disk latency is the only disk measurement for which there are generally accepted ranges that represent good and bad performance from a SQL Server perspective. Disk queue metrics, for example, are not accurate for many SAN systems, and there are also no universally agreed upon good and bad ranges for SQL Server. The following ranges can be used as a general guideline to determine whether disk latency is acceptable:

- Less than 10ms - Fast *
- Between 10ms - 20ms - Acceptable
- Between 20ms - 50ms - Slow
- Greater than 50ms - Critical

* For transaction log writes, between 0ms and 2ms is desirable.

ms/Write

The average time in milliseconds each physical disk write is taking.

SQL Server

SQL Server Activity

The SQL Server Activity chart displays information about what the SQL Server instance is doing.

Batches

The total number of select, insert, or delete statements per second, including those inside a stored procedure. The name is somewhat misleading since it doesn't represent batches (groups of multiple statements) in the traditional sense. It's one of the best measures of overall activity on a SQL Server.

Over 1000 Mb per second is generally considered moderate to high activity. A 100Mb network can reach saturation at around 3000 Mb per second.

Compiles

The total number of initial compiles and recompiles per second. The value should generally be < 10 percent of
batches per second. Higher values indicate plan reuse is low, and will generally correlate with high CPU, since plan compilation is a CPU intensive operation. It may also correlate with low cache hit ratios for object and/or SQL plans.

It can also be a strong indicator of memory pressure, since there may not be enough room to keep all plans in cache.

**Recompiles**

The number of recompiles per second. The value should generally be < 10 percent of initial compiles per second.

**Transactions**

The total number of transactions per second across all databases on the server. A transaction can be either a user-defined statement block surrounded by a BEGIN TRAN and END TRAN, or an individual DML statement (insert, update or delete).

Compare with batches per second. On systems with high DML you typically want to see a low ratio of transactions to batches. A low ratio indicates that the individual statements are being bundled together, and can result in dramatically higher throughput and reduced IO due to log flushes.

**Key Lookups**

The number of times per second that the query processor had to perform a key lookup, across all queries. Lookups occur when the index being used is non-covering, meaning it doesn't include all of the columns required by the query. For each row returned by the index operation, the query processor has to go back to either the clustered index to perform a key lookup, or the base table to perform a RID lookup in the case of a heap.

Lookups are a high overhead operation, especially when large number of rows are involved, because each lookup incurs a random I/O and additional processing. This often correlates with higher CPU usage and page reads. Lookups can be eliminated by using a covering index, adjusting joins to reduce the set so the lookup isn’t needed, or using multiple indexes (intersection).

**Forwarded Records**

The number of times per second that the query processor had to lookup forwarded records, across all queries. Forwarded records occur in tables with no clustered index (heaps) when rows become too large to fit on the page and have to be relocated. Over time, this can cause severe fragmentation and queries to incur much higher than normal I/Os, specifically random reads. This can correlate with high SQL Server page reads, and high SQL Server disk wait time, data file and physical disk latency if the disk system isn’t keeping up with the additional reads. On many systems it’s not unusual for this counter to stay at zero if all tables have a clustered index, any heaps aren’t fragmented, or they just aren’t accessed frequently.

**Backup/Restore MB**

The data rate in MB/sec for any backup operations taking place on the server.

**SQL Server Waits**

The **SQL Server Waits** chart displays information about the classes and categories of waits that occurred as
well as the duration of milliseconds that the waits were in effect during that time period.

Although there are hundreds of wait types, only the wait types that can be definitively attributed to one of the physical resource categories (Disk, Memory, CPU, and Network) are included in the calculations for this chart. The Other category is for a few other important wait types that can either affect performance in more than one major category, or cannot be directly attributed to any category with absolute certainty, such as backups and parallelism respectively.

SQL Server Waits is one of the most important charts on the dashboard, because it provides an instant profile of the SQL Server and where it's spending the most time waiting for physical resources. If SQL Server are consistently low, then what the other dashboard charts are showing is less important. For example, if CPU and SQL Server Activity: Batches look unusually high, but CPU waits are low, then the server hardware is effectively handling the load.

Total waits of less than 200ms is excellent. Between 200ms and 1000ms is average. Greater than 1000ms likely requires some attention to determine where the bottleneck lies. Over 5000ms may indicate severe bottle-necking.

The total wait time may be higher by virtue of a large number of processes (spids) active on the server, because wait time is summed across all processes, it isn’t a per process average. This can be especially applicable to the Other category, because several processes experiencing parallelism at the same time can cause it to spike to high levels.

Additional Information:
- SQL Server Best Practices Article
- What to do (or not do) about top wait stats

SQL Server Memory

The SQL Server Memory chart displays information about how the Server instance is using memory that has been allocated to it.

Buffer

The current size of the buffer cache (in MB). You want this to be as large as possible for maximum performance, and on a dedicated SQL Server it should consume most of the SQL Server memory and physical memory.

Plan (SQL)

The current size of the cache used for query plans (in MB). This includes ad-hoc, auto-parameterized, and prepared plans. A high value in proportion to the buffer cache may indicate query plans aren’t being effectively reused.

Plan (Objects)

The current size of the cache used for object plans (in MB). This includes stored procedures, functions, and triggers. A high value in proportion to the buffer cache may indicate query plans aren’t being effectively
Caching Mechanisms

Columnstore

The current size of the Columnstore index on the SQL Server (in MB). This includes both clustered and nonclustered columnstore indexes.

In-Mem OLTP

The current amount of memory (in MB) dedicated to In-Memory OLTP. This includes Memory-optimized tables, non-durable tables, and natively compiled T-SQL modules.

Other

The current size of the cache used for all other plans (in MB). This includes bound trees, extended stored procedures, temporary tables, and table variables. This cache size should be low in proportion to the other plan caches. If it goes over roughly 10 percent of the object or SQL plan size, further investigation may be needed.

PLE (sec)

The average lifespan of a data page. If this value is less than 600, it's an indicator of memory pressure. Ideally, it should be much higher than 600 if ample memory is available. In general, the larger the buffer cache size, the higher it should be. This is the best universal indicator of memory pressure.

Plan (SQL)

The ratio of hits to lookups for the query plan cache. This value should stay above 90 percent.

Plan (Object)

The ratio of hits to lookups for the object plan cache. This value should stay above 90 percent.

Pages: Reads

The average number of buffer data pages read from disk per second. Ideally, this value should be at or near zero most of the time. If it's above zero, it means that the data wasn't found in the buffer cache, and so it had to be retrieved from disk. If spikes in page reads correlate with high disk latency, the disk system may not be keeping up.

Querying newly created temp tables will also show up as page reads, as well as activity from internal tempdb objects. This includes hash joins, hash aggregates, sort, and query spool operations. This means that you can still see high paging from tempdb due to query activity, even though you aren't explicitly using temp tables.

When page reads and page writes correlate closely, it's a strong indicator that it's related to tempdb activity, because pages are being written to disk when the objects are created, then immediately read back in to memory for use by querying operations.
If lazy writes > zero and track closely with page reads, and page life expectancy < 600, it's a strong indicator of memory pressure, because data is being moved out of buffer to make room for new data coming in.

Lazy writes also cause page writes, but generally much less than tempdb activity. If you see high page reads, and relatively low lazy writes and page writes, it's likely memory pressure and not tempdb activity.

**Pages: Writes**

The average number of buffer data page writes to disk per second.

Page writes can be caused by checkpoints, lazy writes, and tempdb activity. To calculate the approximate amount of writes related to tempdb, for any given interval, subtract checkpoints and lazy writes from total page writes.

If high page writes correlate with high latency, the disk system may not be keeping up.

**Database I/O**

The **Database I/O** chart displays information about the read and write latency for the databases.

**ms/Read**

The average time in milliseconds each physical disk read is taking for a particular database file. The top 10 database files (data and transaction log) with the highest latency for the specified date range are shown.

**ms/Write**

The average time in milliseconds each physical disk write is taking for a particular database file.

**Log Flushes**

Log flushes occur with every DML operation, and are a normal part of SQL Server activity. It's important to note that log writes to physical disk from updates to buffer pages happen immediately upon transaction commit, whereas writes to physical disk from the changed buffer pages is delayed until the next checkpoint occurs. It’s critical that the physical disk system where the transaction log resides is fast enough to keep up with activity. If not, it can slow down all DML operations occurring in the database.

Ideally each busy transaction log should have its own dedicated disks, so that writes can happen sequentially, which will minimize latency. If log flushes are high and latency is high for a transaction log file, then the disk system is likely under-powered for the current load.

**Checkpoint Pages**

The average pages per second written to disk by the checkpoint process. Checkpoints flush all dirty buffer pages for a given database to disk and are a normal part of SQL Server operations. The frequency of checkpoints and volume of checkpoint pages is dictated directly by the Recovery Interval server option. SQL Server uses checkpoints to batch writes to disk, which is generally more efficient. However, if the volume of each checkpoint is too high and you see a correlation with high disk latency, it may indicate that the disk system isn't keeping up.

**Lazy Writes**
The average number of writes per second by the lazy writer. The lazy writer periodically scans the buffer and evicts pages that have low use counts in order to maintain a certain number of pages on the free list. Ideally, this value should be at or near zero most of the time. When there is no memory pressure, the lazy writer will generally leave data pages in memory, even those with low use counts. However, when pressure exists, the lazy writer will continually be working to make room for new data coming into the buffer.

An indicator of memory pressure is ongoing lazy writes > zero with page reads/writes > zero and page life expectancy < 600.

Azure SQL Database

Resource Usage

A DTU represents the power of the database engine as a blended measure of CPU, memory, and read and write rates. This measure helps you assess the relative power of the SQL Database performance levels. Each service tier, which sets pricing and usage limits for an Azure SQL Database, expresses the amount of resource limits as a number of DTUs. The more DTUs an Azure SQL Database is allocated the more resources the database will have to service the workload.

**Additional Information:** See the What is Azure SQL Database? article on Microsoft Docs for more information regarding DTUs, purchasing models, and service tiers.

Total DTU %

If your database is seeing high Total DTU percentage usage it may benefit from adjusting to the next highest service tier to improve performance. If you’re consistently seeing very low total DTU percentage usage you may save some money by scaling down to the next lower service tier.

Data I/O

This metric is the average Data I/O percentage based on the limit of the service tier. This is one of the metrics that makes up DTU.

Log I/O

This metric is the average log I/O percentage based on the limit of the service tier. This is one of the metrics that makes up DTU.

CPU %

This metric is the average CPU percentage based on the limit of the service tier. This is one of the metrics that makes up DTU.

Memory Usage

**Allocated Memory Usage**

Each service tier has a maximum amount of memory allowed for the Azure SQL Database to use. This metric provides the percentage of the allowed memory being used for the database.
It will be very common for this metric to be high. If much of the data your applications need is in memory it means better performance because the database doesn’t have to read from the physical disk to return the data.

**Database Size**

Each service tier has a maximum allowed size for the Azure SQL Database. This chart uses that tier to determine the total space available.

**Used Space**

The space used by the database in MB.

**Free Space**

Amount of space remaining (in MB) from the total space allowed for the tier.
Introduction

The Top SQL view displays a unified picture of collected SQL statements. It's designed to help you quickly identify queries, applications, logins, and more that are causing the most waits, using the most resources, taking the most time, and putting the most load on your SQL Server.

**Note:** Top SQL data is retained for 15 days in SentryOne Monitor. If you’re using the SentryOne Portal feature for SQL Sentry, this default value may be changed and is controlled by the Monitoring Service Settings.

Available Charts

The full viewing options for this card’s charts are:

- Waits
- Resources
- Queries
- By App
- By DB
- By Host
- By Login

Select the button in the upper right for additional options such as reset, show/hide axis labels, and show/hide axes.

Show/hide axes has the following options:

- Avg Duration (ms)
- CPU Time (ms)
- Exec Count
- Reads Logical
- Writes Logical
- Reads Physical
On the options with the ➤ (chevron-right), additional choices similar to the above are available by selecting the ➤ symbol. For example, *By App - Duration (ms)* and *Queries - Reads (P)* are available chart options.

**Waits**

The first card in **Top SQL** defaults to a **Waits** view. Waits displayed here are from the SQL Server instance level. For a better understanding of waits, see the SQL Server Waits Stats section of the **Dashboards** article and this blog post *What to do (or not do) about top wait stats*.

Hover over an area on the chart to view additional details about the waits:

**Resources**

View resource usage from Top SQL, based on query and procedure stats:

Hover over a point on the chart to view additional details about the resources being used:
Queries

Select the ▶ (chevron-right) to view charts for:

- CPU
- Duration
- Exec Count
- Reads (L) - The default selection
- Reads (P)
- Writes (L)

The information displayed here is from query stats, procedure stats, and trace data.

Hover over an area in the chart to view more details about the queries:

By App

Select the ▶ (chevron-right) to view charts for:

- CPU
- Duration
- Exec Count
- Reads (L) - The default selection
- Writes (L)

The information displayed here is from trace data.

Hover over an area on the chart to view more details about the applications:

**By DB**

Select the ➔ (chevron-right) to view charts for:

- CPU
- Duration
- Exec Count
- Reads (L) - The default selection
- Reads (P)

Hover over an area on the chart to view more details about the databases:
By Host

Select the ➤ (chevron-right) to view charts for:

- CPU
- Duration
- Exec Count
- Reads (L) - The default selection
- Writes (P)

The information displayed here is from trace data.

Hover over an area on the chart to view more details about hosts:

By Login

Select the ➤ (chevron-right) to view charts for:

- CPU
- Duration
- Exec Count
- Reads (L) - The default selection
- Writes (P)

The information displayed here is from trace data.

Hover over an area on the chart to view more details about logins:

Totals

The Totals table displays the text data and associated information such as database, duration, count, and CPU for the collected SQL statements (including procedure stats, query stats, and completed queries).

Note: By default, the Totals grid displays the top 8 queries by logical reads (descending). For all grids, the arrow with the circle around it highlights by which column the data is sorted; ascending or descending.

Trace Events

Select the (chevron-right) under the Events column to display details for any collected Trace Events such
Note: In the upper right of the Top SQL page, there is a details switch. The switch is off by default. Select the switch to turn on details ( ), which flips all totals and statements tables to the detailed Trace Events and Trace Events Statements view on the page.

The Statements card displays additional information about Totals where applicable, including plan diagrams, text data, parameters, and plan XML.

Trace Events Statements

Select the (chevron-right) under the Events column to display statement details to any collected Trace Events such as SP:StmtCompleted.

Plan Diagram

Use the full screen button in the upper left to expand a larger plan, or use the Download Plan button to download the entire .sqlplan file.
Note:

- Sometimes the plan XML may be populated, but there's not a statement that can be matched for the plan diagram. This can be caused by things like nested procedures or individual statements falling outside of the collection thresholds on their own. In this case, the plan diagram will display a message such as "The selected statement was not found in the plan XML. Download the full plan to view in SentryOne Plan Explorer" and provide a Download Plan XML button.

- If the Query History chart shows a disabled point (i.e. a gray triangle), then the plan diagram will display a message to indicate that there are no plans available (e.g. "There is no data to display").

Text Data

Use the Text Data tab to view a formatted and syntax color-coded copy of the statement.

<table>
<thead>
<tr>
<th>PLAN DIAGRAM</th>
<th>TEXT DATA</th>
<th>PARAMETERS</th>
<th>PLAN XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT TOP 1 * FROM (SELECT TableID = i.object_id, TableName = s.name + '.', o.name, InverseName = ISNULL(i.name, $), IndexType = i.type, IndexID = i.index_id, IsPartitioned = CONVERT(BIT, CASE WHEN dsc.type IS NOT NULL THEN 1 ELSE 0 END), i.is_hypothetical, i.is_unique, i.is_primary_key, i.is_nullable, i.is_disabled, i.show_page_locks, fileGroupName = fg.name, DataSpaceType = COMILSCF(dsc.type, fg.type))</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameters

Use the Parameters tab to view compiled values for statement parameters.

<table>
<thead>
<tr>
<th>PLAN DIAGRAM</th>
<th>TEXT DATA</th>
<th>PARAMETERS</th>
<th>PLAN XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>(@AverageExecution)</td>
<td>(0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(@CalculateStatisticsSince)</td>
<td>'1753-01-01 00:00:00.000'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(@Catagory)</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(@CreatedByByname)</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(@CurrentInstanceId)</td>
<td>'1753-01-01 00:00:00.000'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use the **Plan XML** tab to view or copy the ShowPlanXML output.

Note: The Copy button is only available when you are using HTTPS (requires an SSL certificate for your SentryOne Portal installation).

### Query History

Select a query or statement, then view the **Query History** for it. **Query History** displays a graphical representation of the selected query over a specified range of time. **Query History** provides information about the query execution plans, if and when they were changed, and how they impacted different resources.

### Query Event

Each triangle represents a **Query Event**.

Note: The triangle colors represent execution plans. Triangles of the same color are using the same plan. If there is excessive plan drift (beyond 25 plans), then the 25 colors will start to be reused in the same order. Select a specific event to the **Plan #** represented.
Query Stats Sample

Each dot represents a **Query Stats Sample** or a Proc Stats Sample.

Proc Stats Sample

Each dot represents a **Proc Stats Sample** or a Query Stats Sample.

**Note:** The Mode for the Proc Stats Sample is set to **Procedure**.

Additional Options

Use the options below the chart to adjust the **Grouping**, **Show**, **Metric**, **Mode**, or **Dates** slider window.

- **Grouping**
  - None
  - Hour
  - Day
  - Week
- **Show**
- Actual/Average
- Totals
- Metric
  - Duration
  - CPU
  - IO
- Mode
  - Procedure
  - Statement

**Note:** In Procedure mode, the chart reflects changes in the procedure stats (plan_handle), whereas Statement mode displays the changes in query stats (query_plan_hash).

Example with **Grouping** by Hour, Show Totals, Metric CPU, and Mode Statement:

An additional example with multiple plans represented:

**Additional Information:**
- Multiple Plans for an "Identical" Query blog post by Aaron Bertrand on SQLPerformance
- Different Plans for "Identical" Servers blog post by Aaron Bertrand on SQL Performance
- Analyzing "death by a thousand cuts" workloads blog post by Erin Stellato on SQLPerformance
- How useful are query_hash and query_plan_hash for troubleshooting? blog post by Jonathan Kehayias on SQLskills
Overview

Applies to the following products and features: The SentryOne Monitor product and the on-premises SentryOne Portal feature for SQL Sentry.

Note: Only the Log view is available for SentryOne Portal.

SentryOne Monitor delivers a workable set of alerts that have been carefully chosen by experienced Microsoft data platform professionals for their relevance to most database monitoring situations.

From the Alerts page, you can configure alerts to send email notifications to yourself or other members of your organization. See the Details & Emails section in this article for step-by-step instructions.

Log

Note: The on-premises SentryOne Portal Alerts view only displays the Log. All alert configurations are still managed via the SentryOne client.
**Note:** The default view on the Alerts page displays the 5 most recent alerts logged across your environment *(Start Time ⬇).* You can sort the table by any of the available columns.

**Note:** The following columns are all read-only or hyperlinks.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Target** | The covered area of the watched target that triggered the alert.  
*Note:* If the scope of the target is at the server or instance level (i.e. Q-REGRESSION and not Q-REGRESSION: SQL Server Agent Jobs), then it will be a hyperlink to display Health, Dashboards, or Top SQL for that target. |
| **Type** | The alert type, such as SQL Server, a deadlock (Deadlocks: Deadlock), etc. |
| **Name** | The name of the alert. |
| **State** | The state of the alert (e.g. active or completed). An active alert has a start time, but no end time, as the situation is still meeting the conditions of the alert. |
| **Severity** | The severity may be high, medium, or low.  
*Note:* The severity is blank unless the alert is sourced from an advisory condition alert category. |
| **Start Time** | The time the alert started evaluating to true. |
| **End Time** | The time the alert stopped evaluating to true. |
| **Duration** | The amount of time that the alert was true.  
*Note:* The smallest value displayed is in seconds. If an alert was active for 500ms, it would display a duration of < 1s. |

Select the Details hyperlink to view a popup window containing the details logged for the alert:
The available details vary depending on the condition, and may contain information such as the step of a failed SQL Server Agent job and the error behind the failure.

On an alert such as High CPU, which looks for CPU greater than 90, the performance counter value collected at the time of the alert evaluation is included (e.g. Performance Counter: Processor Information: % Processor Time, Total [97.4264] > [90] *TRUE*).

### Configuration

Applies to the following products and features: The SentryOne Monitor product.

The default view on the Configuration page displays the available alerts in SentryOne Monitor. From this tab you can select the Name to view information about the significance of the alert or configure email notifications. Select the Status to access the Evaluation Status details.
Note: The following columns are all read-only or hyperlinks.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The descriptive name of the alert, such as <em>Data File Shrink</em> or <em>High CPU</em>. Select the name hyperlink to open the Alert window where you can view additional information about the meaning of the alert, as well as set up email notifications. See the Details &amp; Emails section in this article for more information.</td>
</tr>
<tr>
<td>Type</td>
<td>The coverage area (e.g. <em>Global</em>, <em>Top Commands</em>, etc.), target type (e.g. <em>SQL Server</em>, <em>Azure SQL Database</em>, <em>Windows Computer</em>, etc.), or SentryOne component (e.g. <em>Monitoring Service</em>) affected by the alert.</td>
</tr>
<tr>
<td>Tags</td>
<td>These are used to group the alerts by common goals (e.g. <em>Auditing</em>, <em>Performance</em>, <em>CPU</em>, <em>Memory</em>, etc.)</td>
</tr>
<tr>
<td>Email</td>
<td>Displays <em>On</em> if any email alerts are configured or <em>Off</em> if there are none.</td>
</tr>
<tr>
<td>Actions</td>
<td>Select the Status hyperlink to display the Evaluation Status table.</td>
</tr>
</tbody>
</table>

Alert Details & Email Notifications

* Applies to the following products and features: The SentryOne Monitor product.

Select the hyperlinked Name from the Configuration table on the Alerts page to view the Alert window:
The alert window displays a **Severity** at the top. Only alerts with a **Category** of **Advisory Conditions** have an associated severity level. The severity levels may be **High**, **Medium**, or **Low**.

**About**

The **About** section displays information about the alert’s purpose or meaning, and hints or links for how to troubleshoot or investigate (where applicable).

- **Show "About" by default**: When this box is selected, the **About** section is expanded on the **Alerts** window, when it is not selected, the **About** section is collapsed by default.

**Note:** The **Show "About" by default** option is a global setting for all **Alert** windows.

**Email Notifications**

You have the option to send alert emails when an alert’s condition evaluates to **True**.

**Important:** SentryOne will send a maximum of 50 alert emails per day. This maximum is for all emails, across all alerts. It is recommended that you only configure email notifications for critical alerts that require a fast response so that you keep your overall emails to a minimum, and do not miss emails for more important alerts.

Select **Send alerts for [Condition Name] to me at [email@address.com]** to receive email notifications for the associated alert, at the default email address for your user name. A toast notification pops up to confirm the change. Deselect this option to disable email notifications for the associated alert, for your user name.
Use the **Send alerts for [Condition Name] to these email addresses**: option to enter additional email addresses that should receive alert emails.

1. Enter an email address, or multiple email addresses (separated by a comma).
2. Select **Add**.
3. Once you have the email addresses entered, use the **Back to Conditions** button to save the changes.

Select the **X** after an additional email address to stop it from receiving alert emails for the associated alert.

**Important:** Changes to email addresses (adding or deleting) are not saved until you select the **Back to Alerts** button.

**Note:** You can send alert emails to yourself, others, or a combination of these emails.

**Evaluation Status**
The **Evaluation Status** table is accessed by selecting a row from the **Actions** column in the **Configuration** table.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object Name</strong></td>
<td>The name of the target associated with the evaluation.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The hyperlink provides option to jump to other views for the target (e.g. health or dashboard).</td>
</tr>
<tr>
<td><strong>Last Result</strong></td>
<td>Displays the outcome of the most recent evaluation against the alert. The result will be True or False.</td>
</tr>
<tr>
<td></td>
<td>Example: The <strong>High CPU</strong> alert checks for sustained CPU usage over 90%.</td>
</tr>
<tr>
<td></td>
<td>- True means that the last time this alert was evaluated that the monitored target’s CPU was over 90% for at least a minute.</td>
</tr>
<tr>
<td></td>
<td>- False indicates that it did not sustain a value over 90% for at least a minute.</td>
</tr>
<tr>
<td><strong>Duration</strong></td>
<td>The amount of time it took to evaluate the result of the alert.</td>
</tr>
<tr>
<td><strong>Last Evaluation Time</strong></td>
<td>The last time the evaluation completed.</td>
</tr>
<tr>
<td><strong>Last Error</strong></td>
<td>If there was an error during the evaluation attempt, it displays information here.</td>
</tr>
<tr>
<td><strong>Evaluation Type</strong></td>
<td>Displays whether the evaluation was executed as Scheduled or Manual. Each alert in SentryOne Monitor is set to evaluate at a frequency specific to the alert.</td>
</tr>
<tr>
<td></td>
<td><em>Scheduled</em> indicates that the evaluation was performed on schedule by the SentryOne Monitor.</td>
</tr>
</tbody>
</table>
monitoring service. For example, the High CPU alert is scheduled to evaluate every 30 seconds, while the SQL Server Paged to Disk alert is scheduled to evaluate every 5 minutes.

Manual means that a user chose to evaluate the alert ad-hoc through the configuration client. This is an infrequent Evaluation Type in SentryOne Monitor as users rarely need to access the configuration client.

### Available Alerts

 Applies to the following products and features: The [SentryOne Monitor](#) product.

The following alerts are available with SentryOne Monitor:

- % Free Space - Data/Log File(s)
- % Free Space - Disk
- Amazon RDS for SQL Server: Offline
- Amazon RDS for SQL Server: Online
- Azure SQL Database: Offline
- Azure SQL Database: Online
- High Avg Wait Time per User Session
- High CPU
- High CPU for non-SQL Server process
- High Pending Disk IO Count
- SQL Availability Group Failover
- SQL File Auto-growth Exceeds Free Space
- SQL Server Agent Job: Failure
- SQL Server Memory Exhaustion
- SQL Server Process Paged to Disk
- SQL Server Process Physical Memory Low
- SQL Server Process Virtual Memory Low
- SQL Server: Blocking SQL
- SQL Server: Deadlock
- SQL Server: Offline
- SQL Server: Online
- Suspect Pages - Active Corrupt Pages
- Windows: Offline
- Windows: Online
For more information on the additional steps SolarWinds is taking to improve our security posture and policies, please reference [this blog](#) by SolarWinds President and CEO Sudhakar Ramakrishna.

**Applies to:** These release notes are for SentryOne Monitor. For SentryOne Portal and SQL Sentry products, see the SentryOne release notes.

**v2020**

**Version 2020.21 [October 21, 2020]**

**Notes**
- Various code optimizations and security patches

**Version 2020.20 [October 7, 2020]**

**Notes**
- **Top SQL:** Query History Chart: Show last plan represented on the Query History chart (ID 63380)
- **Top SQL:** Display plan XML regardless of matching statements (ID 65426)
- **Top SQL:** Plan XML should provide download of *.sqlplan rather than copy to clipboard (ID 65523)
- **Top SQL:** Single-statement plan matching should consider children nodes (ID 65608)
- **Health:** EHO for Sites/Groups (ID 63798)

**Version 2020.17 [August 24, 2020]**

**Notes**
- **Top SQL:** Text Data: Show text data for both the Totals and Statements grids (ID 64388)
- **Navigation:** Restyling and Behavior Changes (ID 64568)
- **Navigation:** Hierarchy View With Sites and Groups (ID 64571)
- **Dashboards:** SQL Server Memory Chart not rendering PLE line correctly (ID 65239)

**Version 2020.15 [July 29, 2020]**

**Notes**
• Change Dashboard menu option from Filter to Zoom (ID 64896)
• Add multi-chart range selection to Dashboard charts (ID 64885)
• Disable Dashboard cloning option when no Custom Dashboards exist (ID 64707)
• Resolved correlation error during SentryOne Monitor login (ID 62451)

Version 2020.5 [April 10, 2020]

Notes
• Various bug fixes

Version 2020.2 [February 27, 2020]

Notes
• Various usability improvements and bug fixes

Monitor GA 2019.9 [October 31, 2019]

Notes
• SentryOne Monitor now supports Azure SQL DB monitoring and dashboards
• New feature: Email alerts can be configured in SentryOne Monitor
• Multi-user support: Additional users can be added to SentryOne Monitor using the Platform Security Portal
• Various usability improvements and bug fixes

Monitor Preview 2019.4 [August 14, 2019]

Features
• Fix dashboard vertical and horizontal scrollbars (ID 56221)
• Implement date context awareness for event log grid (ID 55432)
• Add time context to Dashboard and Top SQL (ID 53381)
• Top SQL: Plan Diagram: Add Download XML Link (ID 54774)

Fixes
• Overview: Event grid not being filtered by score (55529)

Monitor Preview [July 30, 2019]
Features

- Custom Dashboards added

Fixes

- Top SQL: Statements Grid: Trace Events show no data (ID 52669)
- Top SQL: Resources chart has frequent gaps (ID 53562)
- Fixed Events by Tag Wedget CPU label wrapping (ID 52764)
- Reduce Dashboard DTU Usage (ID 53581)
- Top SQL: No data shown in Statements grid (ID 53564)
- Events by Waits Time/Session not showing wedgets (ID 52753)
- Lower default trace threshold (ID 53383)
- Client: Move Alerts (ID 53331)
- Client: Remove core-based option from Watch > Target menu (ID 53329)
- Client: Update desktop shortcut icon text (ID 53335)
- Client: Update install title (ID 53349)
- Auto-select Monitor connection when client is launched (ID 53343)
- Client: Remove save session prompt (ID 53332)
- Alerting Email: Wrong URL is displayed (ID 53266)
- Client: Enable include logins and hot names by default (ID 53268)
- Client: Remove target types that cannot be watched (ID 53262)
- Client: Hide Open button in watch status window (ID 53269)
- Client: Align instance name textbox in add target window (ID 53338)
- Managed SQL instance targets are not displayed in the Cloud Sync tab in the config client (ID 52750)
- On-Premises Configuration Client: Settings Pane: SMTP Server is not read-only in global settings (ID 52796)
Overview

Existing Accounts & Passwords

You can access or edit your existing account information at my.sentryone.com.

⚠️ Warning: If you already have a SentryOne Cloud Sync or my.sentryone.com account, you must ensure that your passwords are consistent across all SentryOne account types. Use your existing password when setting up your SentryOne Monitor password.

Local Installation

The following components will be installed on a local machine and must all be on the same machine:

- SQL Server Express version of the SentryOne database specific to Monitor
- SentryOne monitoring service
- SentryOne configuration client

⚠️ Warning: The machine where these components are installed will be running the monitoring service which must be able to run 24/7 in order to avoid interruptions in monitoring your environment. Ensure that this is a machine that will always be running (i.e. not your laptop).

⚠️ Important: If the Microsoft Visual C++ Redistributable Package for Visual Studio 2017 is installed without 2015 on the local machine running the SentryOne components, the SQL Server installer fails when attempting to install 2015 as a dependency. In this scenario, you must uninstall the 2017 package before running the installer for the configuration client.

Targets

You can monitor up to 10 instances of SQL Server, Azure SQL Database, Azure SQL Managed Instance, or Amazon RDS. The monitored SQL Servers may be running on your local network or on virtual machines in AWS.
and Azure.

**Recommended Browsers**
- Chrome
- Microsoft Edge (the version based on Chromium)

**Security**
- SentryOne Monitor utilizes standard SSL Port 443 to establish a connection.
- SentryOne Monitor does not send Windows or SQL Server password information to the cloud.
- See the Organization Settings article for information about adding new users or editing existing ones for SentryOne Monitor.
- For security requirements related to the SentryOne monitoring service account, see the Monitoring Service Security article.

**Additional Information:** Visit the SentryOne Cloud Security section of the product FAQs page for additional information.

**Status**
You can check on the operational status, incident history, and uptime statistics of Monitor and other SentryOne cloud-based products at status.sentryone.com.

**Service Agreement**
The Cloud Services Agreement may be accessed at: https://www.sentryone.com/cloud-service-agreement.

**Instructions**

1. **Get SentryOne Monitor**
Log in with your my.sentryone.com account credentials. If you do not have an account, you will be asked to set up a username and password during the purchase process. This information will be used when you configure SentryOne Monitor and access via your browser.

2. **Download the Installer**

   **Note:** The installer will install the following components on the local machine:
   - SQL Server Express version of the SentryOne database specific to Monitor
   - SentryOne monitoring service
**SentryOne configuration client**

Download the installer and execute the file to install the on-premises components of SentryOne Monitor.

| Step                      | Example
|---------------------------|--------------------------------------------------|
| Execute the installer    | ![SentryOne Platform Setup](image)

**Note:** See the Monitoring Service Security article for details on the security requirements for the service account login.

| Complete setup            | ![SentryOne 19.2 Setup](image)

**Note:** See the Monitoring Service Security article for details on the security requirements for the service account login.

3. Use Configuration Client
1. Open the on-premises configuration client (if not launched after setup).
2. Enter the username and password created during the purchase process.
3. Configure your environment (i.e. add targets). See the **Configuration Client** tab for additional information.

<table>
<thead>
<tr>
<th>Step</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign in</td>
<td><img src="image" alt="SentryOne Sign in" /></td>
</tr>
</tbody>
</table>

Looks like this is your first time...
First a few quick questions to help you get started.
Add Targets

Note: See the Monitoring Service Security article for details on the security requirements for the service account login to watch a target.

Important: All configuration must be done through the configuration client on the local machine where it is installed.

4. Access SentryOne Monitor

Use a supported browser to view performance and monitoring information, including features such as the SentryOne dashboard, server health status, alerts, and Top SQL.

Warning: Do not attempt to log into SentryOne Monitor until you have completed the configuration client setup from step 3.

1. Go to https://monitor.sentryone.com/
2. Enter the username and password created during the purchase process.

Success: You are now ready to use SentryOne Monitor!

Additional Information: See the Using SentryOne Monitor article for more details.
Configuration Client

The **Configuration Client** is used to perform tasks such as adding or deleting targets.

**Add Target**

1. **Note:** Azure SQL Database targets support the following authentication methods:
   - SQL Server
   - Active Directory - Password
   - Active Directory - Integrated

2. To add a new target, right-click on **Default Site** in the **Navigator**, then select **Add** and a target type:

3. Enter the target information:
3. The monitoring service will connect and start monitoring the target in your Monitor environment:

Delete Target

⚠️ Warning: All history data collected for a target will be permanently deleted when you delete the target.

1. To delete a target, you must first stop watching all of the instances associated with it. Right click the instance, go to Stop Watching, then select Instance.
2. Next right-click the target, then select **Delete Target**. You need to confirm that you want to delete the target.
Monitor & Portal Custom Dashboards

This feature has been deprecated in the on-premises SentryOne Portal feature for SQL Sentry in versions 2021.1 and later. It is replaced by Custom Charts.

Applies to the following products and features: The SentryOne Monitor product and the on-premises SentryOne Portal feature for SQL Sentry (versions earlier than 2021.1).

SentryOne Monitor offers the ability to create custom dashboards. With custom dashboards, you can include the performance metric widgets that you want, and you can have performance charts for different targets on the same dashboard.

Create Custom Dashboard

1. To create a custom dashboard, select Add New from the bottom of the Dashboards drop-down menu:

2. After selecting Add New, the Create a New Dashboard dialog appears where you will enter the name of the new custom dashboard or select the name of existing one to clone. When you are done entering the Dashboard Name, select Create Dashboard.

Note: You must have existing custom dashboards to have the option to clone one using a Dashboard Template.
3. Once the dashboard appears, use the edit button in the upper right to customize the dashboard:

4. Select the performance charts from the list of Available Widgets to add them to your custom dashboard:

5. Once you've added the desired widgets, select Save Dashboard:
Success: You’ve created your first custom dashboard!

Note: This custom dashboard is available across other targets.
Configure Widget

Widgets may be configured to have a **Custom Date Range** or a **Custom Target** (which allows for multiple targets per dashboard).

**Custom Date Range**

A **Custom Date Range** allows you to configure a widget to display a defined number of minutes, hours, or days.

To set a Custom Date Range:

1. Select 🖼 edit on a custom dashboard.
2. Select ⚙ configure on a specific widget.
3. Toggle the Use Dashboard Time Range switch to false.
4. Use the Enter Number textbox and Select Units drop-down to enter an amount of time.
5. Select Update on the Configure Widget screen.

In this example, the top CPU Usage chart has the default Dashboard Time Range, and the bottom chart is...
displaying a **Custom Time Range** of 5 minutes.

![CPU Usage Chart](image)

Multiple Targets per Dashboard

1. If you would like to have different targets on the same dashboard, use the **Configure** option on a widget in the custom dashboard to open the **Configure Widget** options. Under **Configure Widget** options, toggle **Use Dashboard Target** to **off**, then click **Select Target**.

![Configure Widget](image)

2. Use **Select Target** to open the target selection window and choose a target to associate with the performance chart widget. You can type into the filter area or use  (chevron-down) to select a target from the list of devices.

![Select Target](image)
3. Once you’ve selected a device, you'll see it populated under **Custom Target**. Select **Update** to set the chart widget to the **Custom Target**:

4. Configure the widgets until you have the performance chart and target combination desired for your custom dashboard and save your changes using **Save Dashboard**. The example below shows a custom dashboard with CPU Usage for a custom target and Disk I/O for the primary target of the dashboard:
The example below shows a custom dashboard with three SQL Server Activity chart widgets, all configured to display the values for a different target:

SQL Server vs. Azure SQL DB Widgets

When adding widgets, you’ll see SQL Server and Azure SQL DB labels under the widget name. If you add an Azure SQL DB widget to a custom dashboard that has an Azure SQL Database as the default target, then the chart widget will use that target.

Add an Azure SQL DB widget to a dashboard that has an Azure SQL Database as the default target:
The widget loads for the default target without any additional configuration:

However, if you start from a different target type as the base (SQL Server in this case), you’ll need to configure the target to an Azure SQL DB instance to populate the chart data.

Add an Azure SQL DB widget to a dashboard with a different target type from the default target. The widget is added without data to display. The default target is a SQL Server which doesn’t have Azure SQL DB data to match the widget:
Select ⚙ edit on the widget to select a different target for the chart. Configure per the Custom Target instructions under Multiple Targets per Dashboard.

Note: The list of targets and instances provided in the Select Target list will be specific to targets of the same widget type.

Clone Dashboard

Once you have a custom dashboard created, you may be able to save time by using it as a template for
additional custom dashboards.

To clone a dashboard:

1. Select **Add New** from the **Dashboards** menu.

2. Enter a **Dashboard Name**.

3. Select a **Dashboard Template** from the drop-down menu to select the name of the dashboard you want to clone.

4. Select **Create Dashboard**.

5. The cloned dashboard appears and you can use the **edit** button to configure the options for your cloned dashboard.

Delete Dashboard

While viewing a custom dashboard, select the **edit** button to open **Edit Dashboard** options and select **Delete Dashboard** at the bottom to delete the dashboard.
Edit Dashboard

Available Widgets

- Database I/O (Azure SQL DB)
- Database Size (Azure SQL DB)
- Memory Usage (Azure SQL DB)
- Resource Usage (Azure SQL DB)
- SQL Server Activity (Azure SQL DB)
- SQL Server Waits (Azure SQL DB)
- CPU Usage
- Database I/O (SQL Server)
- Disk I/O
- Network

Delete Dashboard

Dashboard deletion cannot be undone.