# Table of Contents

**SentryOne Cloud** ............................................................................................................................................................................. 3  
Organization Settings ........................................................................................................................................................................... 3  
**SentryOne Monitor & SentryOne Portal** ............................................................................................................................ 8  
SentryOne Monitor Release Notes .................................................................................................................................................. 8  
Getting Started with SentryOne Monitor .................................................................................................................................. 10  
Using SentryOne Monitor ................................................................................................................................................................... 18  
Monitor Health .................................................................................................................................................................................... 21  
Monitor Dashboards ............................................................................................................................................................................ 26  
Monitor Custom Dashboards ............................................................................................................................................................... 34  
Monitor Top SQL .................................................................................................................................................................................. 43  
Monitor Alerts ....................................................................................................................................................................................... 54  
Monitor Blocking .................................................................................................................................................................................. 62  
Monitor Deadlocks ............................................................................................................................................................................... 68  
Monitor TempDB .................................................................................................................................................................................. 73  
**SentryOne Software** .................................................................................................................................................. 81  
**Installation** .................................................................................................................................................................................. 81  
SentryOne Portal Configuration ..................................................................................................................................................... 81  
**Getting Started** .................................................................................................................................................. 89  
Getting Started with SentryOne Portal .................................................................................................................................... 89
Organization Settings

Overview

The Organization Settings page is applicable to all SentryOne cloud-based products (Monitor, Document, and Test). It can be accessed at https://myorg.sentryone.com.

Organization Settings allows you to add new users to the organization so you can share the same Monitor, Document, or Test environment data, as well as assign user roles such as Admin.

Note: The Admin user role in this context is only for the Organization Settings. This is not related to SQL Server or Windows administrator privileges. A user with the Admin role has the ability to manage users for the organization as described below in the Manage Users section.

User Profile

Change Password

Select the User Profile button in the upper right corner to change your password or log out of Organization Settings.
Manage Users

**Note:**

- Only users with *Admin* as their **Assigned Role** can access the **Manage Users** screen and its features.
- The first user added to the account (at the time of account creation) is assigned the *Admin* role.
- If an organization has only one user, the user is assigned the *Admin* role by default.
  
  - The admin privileges cannot be removed from this sole account, nor can the user be removed if it remains the only one.

Select **Manage Users** to continue. From the **Manage Users** screen, you can **add**, **edit**, or **delete** users.
Note: Registered users are brought into the list of users by default. These legacy users may not have name information populated. That information can be edited as described below.

Add New Users

Select **New User** to add a new user to the organization.

1. Enter the **First Name**, **Last Name**, and **Email** address for the user.
2. The **Role** is set to **User** by default. Use the drop-down menu to select **Admin** if necessary.
3. The default **Status** is **Active**, but may be switched to **Inactive** on this screen. **Note:** **Inactive** is intended for situations when you need to temporarily disable users.
4. Select **Save** to complete the process.

Note: For **SentryOne Document**, users must be added to both **Organization Settings** and the **Activation Key** in [https://my.sentryone.com/](https://my.sentryone.com/).

1. Log into [https://my.sentryone.com/](https://my.sentryone.com/), then go to the bottom of the page to the **S1 TaskFactory/Workbench Licenses** section.
2. Select the **Activation Key** text for the license where you want to add a user.
3. On the **Manage License** view, select **View Contacts**.
4. Select Add Contact from the Contact Details view.

5. On the Add Contact form, enter the matching Name and Email used in Organization Settings, then select Add.

![Add Contact form](image)

**Success:** Your contact is now ready to use SentryOne Document!

### Edit Users

Use the Edit button on the Manager Users screen to enter the user's detail screens and make changes as needed.

### Delete Users

Use the Delete button on the Manager Users screen to delete the user from the organization.

There will be a Delete user? popup to confirm that you want to delete the user and a User removed from
the organization toast notification to confirm the change was successful.
SentryOne Monitor Release Notes
Last Modified on 23 April 2020

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
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

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 Platform Security Portal article for information about adding new users or editing existing ones for your SentryOne Monitor installation.
- 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

   1. Go to the SentryOne store to purchase SentryOne Monitor. 🏷️ Note: You can also request a free trial.
   2. 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
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

1. Once you’ve completed your SentryOne Monitor purchase, you will receive an email with a link to download the installer.

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

<table>
<thead>
<tr>
<th>Step</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute the installer</td>
<td><img src="image" alt="SentryOne Platform Setup" /></td>
</tr>
<tr>
<td>Complete setup</td>
<td><img src="image" alt="SentryOne 19.2 Setup" /></td>
</tr>
</tbody>
</table>

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="Sign in" /></td>
</tr>
</tbody>
</table>

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

_SentryOne™_
Monitor • Diagnose • Optimize

Continue
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

**Note:** Azure SQL Database targets support the following authentication methods:

- SQL Server
- Active Directory - Password
- Active Directory - Integrated

1. To add a new target, right-click on **Default Site** in the **Navigator**, then select **Add** and a target type:
2. 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.
Using SentryOne Monitor
Last Modified on 20 July 2020

Overview & Navigation

SentryOne Monitor is accessed through a browser at monitor.sentryone.com.

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 feature navigation (Health, Top SQL, Alerts, and Dashboards) and the monitored targets. Full details are provided under the Sidebar section below.</td>
</tr>
<tr>
<td>3. Health Score</td>
<td>Health Score shows a summary of overall health scores for the monitored environment.</td>
</tr>
<tr>
<td></td>
<td>- Alerts by Severity breaks them into Critical, High, Medium, or Low (with a count for the number of alerts in each severity category).</td>
</tr>
<tr>
<td></td>
<td>- 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).</td>
</tr>
<tr>
<td>4. Alerts</td>
<td>A table of the top 10 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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sidebar toggle</td>
<td>The bars (or hamburger) button toggles the visibility of the left sidebar (the feature navigation and targets information)</td>
</tr>
<tr>
<td>2. SentryOne logo</td>
<td>Selecting the SentryOne logo returns you to the home view.</td>
</tr>
<tr>
<td>3. Product drop-down</td>
<td>The SentryOne product drop-down allows you to switch between Monitor, Test, and Document. Each product has their own licensing requirements.</td>
</tr>
<tr>
<td>4. Date Selector</td>
<td>This is disabled on the default home view. On other views, such as Dashboards, it allows you to select a time frame for which to display data on the view:</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Date Selector" /></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Use to pause live data, to jump to current data, and to view live data where applicable.</td>
</tr>
<tr>
<td>5. Feedback</td>
<td>Select the feedback button to submit ideas about any issues, features you’d like to see, or suggested changes.</td>
</tr>
<tr>
<td>6. Documentation</td>
<td>Select the documentation button to open the Monitor documentation in SentryOne Docs.</td>
</tr>
<tr>
<td>7. Profile</td>
<td>Select the user profile button to change your password or log out of Monitor.</td>
</tr>
</tbody>
</table>

Sidebar

The sidebar has four main areas:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Health displays information about events and</td>
</tr>
</tbody>
</table>
1. Feature navigation

- Top SQL provides a unified view of collected SQL statements for the monitored environment.
- Alerts displays a log of alerts that have been triggered in your environment as well as some configuration options for email.
- Dashboards contains the performance charts for the monitored target or instance, and allows you to create customized dashboards.

2. Target filter (by name)
Start typing a target name in the filter to get to a filtered list of targets or a specific target.

3. Target filter (by type)
Choose All SQL Servers or All Windows Servers to get a health overview of those target types.

4. Targets
Targets provides a list of all targets in your monitored environment.

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

The Health view is the default view when opening SentryOne Monitor. This screen displays widgets for Alerts by Severity and Alerts by Tag on the default view for All Servers.

All Servers View

The All Servers view 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.

In the example below, the Alerts by Severity widget is displaying 2 critical and 186 high severity alerts. In the Alerts table (now sorted by Severity), you can see 2 alerts with a severity of critical, and 3 rows of high alerts (183, 2, and 1) totaling 186. This is how the number in the color-coded alert is calculated.
Note: Selecting the blue hyperlinked target name on the left provides an option to switch to the Health, Dashboard, or Top SQL view for that target.

Details

Select the ▶ (chevron-right icon) to the right of the count to display the individual details about each logged alert:

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

### Alerts by Severity

The **Alerts by Severity** view (shown above) works by incorporating alerts and their associated severity to calculate the scores. A high health score (with a maximum of 100 possible) is an indicator of good health. The score for these views is based on the severity of all open alerts within the selected time period. The severity points are as follows:

- 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.

A table of alerts is below the targets.

**Note:** Select the (chevron-up icon) on the right to collapse the **Targets** grid.
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.

---

**Target View**

Select a target from the sidebar to see the details for that specific target.

---

**Related Targets**

As seen in the image above, there is a list of Related Targets (3 SQL Servers 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.
Monitor Dashboards

Last Modified on 16 July 2020

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

The Dashboard view displays the performance charts for monitored targets.

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 navigation bar displays a pause button (which flips to a play button) to toggle the view of live data, and a 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:
Hover over a point in a chart to view additional details:

SQL Server Memory

The same chart with Disk filtered out:

Highlight an area on the chart for options to Zoom or Jump to Top SQL for that time period:
Dashboard Charts

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 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 bottlenecking.

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 reused.

Additional Information: 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.
Monitor Custom Dashboards

Last Modified on 16 April 2020

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

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 Dashboard dialog appears where you will enter the name of the custom dashboard. You have the option to start with a Blank Dashboard or Clone Existing. If you select Clone Existing, you must then select from an existing dashboard to clone. When you are done, select Create Dashboard.

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:

![Available Widgets](image)

5. Once you’ve added the desired widgets, select **Save Dashboard**:

![Save Dashboard](image)

**Success:** You’ve created your first custom dashboard!
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 **Save** on the **Configure Widget** screen.
6. Select **Save Dashboard** on the **Edit Dashboard** 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.
Multiple Targets per Dashboard

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

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 \(\text{	extgreater}\) (chevron-down) to select a target from the list of devices.
3. Once you’ve selected a device, you’ll see it populated under **Custom Target**. Select **Save** 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:
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 Clone Existing from the radio button options.
4. Use the Select Existing drop-down to select the name of the dashboard you want to clone.
5. Select Create Dashboard.
6. The cloned dashboard appears and you can use the edit button in the upper right to configure the options for your cloned dashboard.
Delete Dashboard

While viewing a custom dashboard, select the edit button from the upper right to open Edit Dashboard options and select Delete Dashboard at the bottom to delete the dashboard.
Monitor Top SQL

Last Modified on 23 July 2020

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

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 symbol, 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:

![Chart](chart.png)

**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 $\downarrow$). For all grids, the arrow with the circle around it highlights by which column the data is sorted; ascending $\uparrow$ or descending $\downarrow$.

### Trace Events

Select the $\rightarrow$ (chevron-right) under the Events column to display details for any collected Trace Events such as RPC:Completed.

**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 ( $\boldsymbol{\text{details}}$ ), which flips all totals and statements tables to the detailed Trace Events and Trace Events Statements view on the page.

### Statements

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 button on the lower right to download the entire `.sqlplan` file.

Text Data

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Compiled Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>@LastLevelBreakMinutes</td>
<td>120</td>
</tr>
<tr>
<td>@LevelBreakMinutes</td>
<td>240</td>
</tr>
<tr>
<td>@RollupEndTimestamp</td>
<td>125311580</td>
</tr>
<tr>
<td>@RollupStartTimestamp</td>
<td>125308800</td>
</tr>
</tbody>
</table>

Plan XML

Use the Plan XML tab to view or copy the ShowPlanXML output.

Query History

Select a query or statement above, 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**.

**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.

**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:

**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
Monitor Alerts

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 tab 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.
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.

The following columns are all read-only or hyperlinks.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target</strong></td>
<td>The covered area of the watched target that triggered the alert.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: 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.</td>
</tr>
<tr>
<td><strong>Closed</strong></td>
<td>An event may be open or closed and this status syncs from the configuration client where it may be set. A closed event may indicate that someone within your organization has responded to or investigated the alert.</td>
</tr>
<tr>
<td></td>
<td><strong>Open</strong> (switch to the left, gray)</td>
</tr>
<tr>
<td></td>
<td><strong>Closed</strong> (switch to the right, blue)</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The alert type, such as SQL Server, a deadlock (Deadlocks: Deadlock), etc.</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td>The name of the alert. Select the hyperlink to navigate to the Configuration tab.</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Severity</strong></td>
<td>The severity may be high, medium, or low.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: The severity is blank unless the alert is sourced from an advisory</td>
</tr>
</tbody>
</table>
The time the alert started evaluating to true.

The time the alert stopped evaluating to true.

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 (as shown in the image above).

On an alert such as High CPU, 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><strong>Name</strong></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 tab in this article for more information.</td>
</tr>
<tr>
<td><strong>Is Enabled</strong></td>
<td>Displays a switch indicating whether or not the alert is enabled to execute checks for the condition.</td>
</tr>
<tr>
<td></td>
<td><em>Disabled</em> (switch to the left, gray)</td>
</tr>
<tr>
<td><strong>Type</strong></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><strong>Tags</strong></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><strong>Actions</strong></td>
<td>Select the Status hyperlink to display the Evaluation Status table.</td>
</tr>
</tbody>
</table>
Alert About & 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 Status and Severity at the top:

- **Status**: Displays On or Off to indicate whether the alert is (High Compiles + High CPU, in this example) enabled to execute checks against the alert’s condition. This is the same as Is Enabled on the Configuration tab.
- **Severity**: Only alerts with a Category of Advisory Conditions may 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. Enter an email address, or multiple email addresses (separated by a comma), then select Add. 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

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

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

![Evaluation Status Table]

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

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Object Name** | The name of the target associated with the evaluation.  
**Note:** The hyperlink provides option to jump to the health, dashboard, or top SQL for the target. |
| **Last Result** | Displays the outcome of the most recent evaluation against the alert. The result will be **True** or **False**.  
Example:  
The *High CPU* alert checks for sustained CPU usage over 90%.  
- **True** means that the last time this alert was evaluated that the monitored target’s CPU was over 90% for at least a minute.  
- **False** indicates that it did not sustain a value over 90% for at least a minute. |
<table>
<thead>
<tr>
<th>Duration Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Evaluation Time</td>
<td>The last time the evaluation completed.</td>
</tr>
<tr>
<td>Last Error</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 <em>Scheduled</em> or <em>Manual</em>. Each alert in SentryOne Monitor is set to evaluate at a frequency specific to the alert. <em>Scheduled</em> indicates that the evaluation was performed on schedule by the monitoring service. For example, the <em>High CPU</em> alert is scheduled to evaluate every 30 seconds, while the <em>SQL Server Paged to Disk</em> alert is scheduled to evaluate every 5 minutes. <em>Manual</em> means that a user chose to evaluate the alert ad-hoc through the configuration client. This is an infrequent <em>Evaluation Type</em> in SentryOne Monitor as users rarely need to access the configuration client.</td>
</tr>
</tbody>
</table>

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
- High Avg Wait Time per User Session
- High CPU
- High CPU for non-SQL Server process
- 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: Offline
- SQL Server: Online
- Suspect Pages - Active Corrupt Pages
- Windows: Offline
- Windows: Online
Introduction

Blocking occurs in SQL Server when a session places a lock on a resource, and an additional session attempts to lock that same resource, but is unable to obtain the desired lock due to the existing lock from the first session. Another session can come along, attempting to lock that resource before the second has even had a chance to do so, creating a blocking chain. When these blocks become excessive and last longer, performance starts to decline and can cause a serious issue in your database’s performance.

Monitor Blocking maps the relationships between all blocking and blocked sessions (SPIDs) in a blocking chain, allowing you to pinpoint the cause and fix the blocking issue.

Blocking by

At the top of the screen is the Blocking by... chart, which allows you to select Application, Wait Resource, or Wait Type. Each chart displays the Total Blocks, Total Time (in seconds), and Average Time (in seconds).

Note: Hover over the charts to view more details.

Application

Use the Blocking by Application chart to discover which application is causing the most or longest blocks.

In the example below, we can see that the .Net SqlClient Data Provider has the longest average block time, but it’s the SQL Server Agent Job ‘Block A’ that has the highest number of blocks, and longest amount of total time with blocking.
Wait Resource

SQL Server can place locks on resources, at different levels, such as a table, page, or single row. The **Blocking by Wait Resource** chart shows you which resources are having the most serious blocks.

Additional Information: Transaction Locking and Row Versioning Guide on Microsoft Docs.

Wait Type

The **Blocking by Wait Type** chart shows you which wait types (e.g. LCK_M_IX, PAGELATCH_SH, etc.) are applicable to the blocks.

Additional Information:
- Introduction to Wait Statistics
- SQLskills Wait Types Library Now Shows SentryOne Data
- What to do (or not do) about top wait stats
- Why Wait Stats Alone Are Not Enough
- Troubleshooting SQL Server: Where are the Waits?

Head Blockers
The **Head Blockers** table displays the details behind the blocks.

The top row is the head of the blocking chain and contains the blocking statement. There will be an icon at the beginning of the row to indicate if the session has completed (check mark) or is currently still running (running person).

Select the ▶ arrow to expand the row and view the blocked sessions in the chain (labeled with the stop button icon). All blocked statements are nested underneath the blocking statement.

### Available Columns

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPID</strong></td>
<td>The session process ID of the associated blocked/blocking process.</td>
</tr>
<tr>
<td><strong>Start Time</strong></td>
<td>Start time of the request.</td>
</tr>
<tr>
<td><strong>Duration</strong></td>
<td>The length of time that the block exists.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Statement</td>
<td>The command text associated with the request.</td>
</tr>
<tr>
<td>Wait Time</td>
<td>Duration of wait time in milliseconds.</td>
</tr>
<tr>
<td>Wait Type</td>
<td>Name of the wait type.</td>
</tr>
<tr>
<td>Additional Information: See the <a href="#">sys.dm_os_wait_stats (Transact-SQL) MSDN article</a>.</td>
<td></td>
</tr>
<tr>
<td>Wait Resource</td>
<td>Name of the resource on which the request is currently waiting.</td>
</tr>
<tr>
<td>Application</td>
<td>The associated application.</td>
</tr>
<tr>
<td>Database</td>
<td>The associated database.</td>
</tr>
<tr>
<td>Login</td>
<td>The login name associated with the session.</td>
</tr>
<tr>
<td>Version</td>
<td>A unique record is shown for each version of a blocking chain, denoted by the Version column. A new row is added each time the blocking chain changes, meaning that a blocked SPID is either added or removed from the chain between polling intervals. For some blocks this may happen frequently, creating multiple rows, while others may not change at all for the duration of the block.</td>
</tr>
</tbody>
</table>

**Text Data**

The **Text Data** section displays the T-SQL for the highlighted statement in the **Head Blockers** table.

```
spid 239 text data

select * from #f
```

**Additional Information:** Back to Basics: The "Runaway" Query

**Filters**

Use the filter icon in the upper right to apply filters to the **Applications**, **Databases**, or **Hosts**.
Make selections on the filter screen, then select **Apply** to apply them, or **Reset** to undo your changes and return to the default view (everything selected, no filters).
Note:

- The filter does not affect the **Blocking by...** charts. It filters values for the **Head Blockers** table only (including the **Text Data**).
- The filter icon 🔄 will be highlighted in blue (.Ticks) if a filter is applied.
- At least one selection must be made from each filter category to apply a filter.
- The filter does not impact the collection of blocking data.
Deadlocks

The **Deadlocks** view provides details about deadlocks within your monitored environment. Use it to identify and fix deadlock issues on your monitored servers.

Note: Deadlocks for Amazon RDS are not supported in [SentryOne Monitor](https://www.sentryone.com) or the on-premises [SentryOne Portal](https://www.sentryone.com).

Deadlock Diagram

The deadlock diagram is built from the captured deadlock XML. The **victim**, **process**, and **resource** (e.g. **Object Lock**) nodes are represented, as well as any relationships that exist between them.

Note:

- Resize the deadlock diagram using the magnifying and minimizing glass icons, and reset it to the original if needed.
- Select different nodes on the diagram to change the information displayed on the screen.
- Use the expand button to open the deadlock diagram and XML in a full window.
- Drag and drop deadlock files into the deadlock diagram space to get a diagram and view additional information.
The deadlock **victim** is highlighted in red (shown as *Victim 60* in the example above). The *victim* is selected, and the **Node Details** and **Locks** associated with the victim are displayed to the right in the image. There are two resource nodes (shown as *Object Lock*, this could also be at a different level such as a *Page Lock*), and a process node (shown as *Process 116 [0]*). Select any of the nodes to display the associated **Node Details** and **Locks** (if applicable).

> **Note:** The numbers (1,2,3, and 4) and associated arrows that connect the nodes indicate the sequence of events that took place to create the deadlock.

### Deadlock Details

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPID [ecid]</td>
<td>The session process ID of the associated owner/waiter.</td>
</tr>
<tr>
<td>Host</td>
<td>The server or workstation name.</td>
</tr>
<tr>
<td>Application</td>
<td>The associated application (e.g. a SQLAgent Job, .Net SqlClient Data Provider, name of a specific application running SQL statements against the associated database, etc.).</td>
</tr>
<tr>
<td>Database</td>
<td>The associated database.</td>
</tr>
<tr>
<td>Login</td>
<td>The user login associated with the session.</td>
</tr>
<tr>
<td>Log Used</td>
<td>The amount of log space used by the process.</td>
</tr>
<tr>
<td>Deadlock Priority</td>
<td>Specifies the <strong>Deadlock Priority</strong>. Zero (0) or <strong>Normal</strong> is the default priority. In cases where each session has the same <strong>Deadlock Priority</strong>, SQL Server chooses the victim based on the least expensive session to roll back.</td>
</tr>
</tbody>
</table>

> **Additional Information:** For general information about the DEDEADLOCK_PRIORITY option, see the Set Deadlock_Priority MSDN article.
### Column Description

**Wait Time**
Time in (ms) milliseconds spent waiting on the resource.

**Lock Mode**
The requested lock mode (e.g. Shared (S), Update (U), Exclusive (X), etc.).

- Additional Information: See the Transaction Locking and Row Versioning Guide and Lock Modes articles on Microsoft Docs.

**Isolation Level**
The current transaction isolation level.

- Additional Information: For general information on isolation levels see the Isolation Levels in the Database Engine MSDN article.

**Last Transaction Start Time**
Time that the last transaction began.

**Trans Name**
Name of the associated transaction.

### Node Details

### Processes

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>States whether the process is the Owner or Waiter for the lock.</td>
</tr>
<tr>
<td>SPID [ecid]</td>
<td>The session process ID of the associated owner/waiter.</td>
</tr>
<tr>
<td>Lock Mode</td>
<td>The requested lock mode (e.g. Shared (S), Update (U), Exclusive (X), etc.).</td>
</tr>
</tbody>
</table>

- Additional Information: See the Transaction Locking and Row Versioning Guide and Lock Modes articles on Microsoft Docs.

| Host            | The server or workstation name.                                            |
| Application     | The associated application name.                                           |
| Login           | The user login associated with the session.                                |
| Text Data       | The associated text data (e.g. T-SQL statement).                            |

### Call Stack

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>The associated object name.</td>
</tr>
<tr>
<td>Line Number</td>
<td>The line number which was being executed when the lock occurred.</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Context</td>
<td>States whether the lock is held by the <em>Owner</em> or is a <em>Waiter</em> for the lock.</td>
</tr>
<tr>
<td>Lock Mode</td>
<td>The requested lock mode (e.g. <em>Shared (S)</em>, <em>Update (U)</em>, <em>Exclusive (X)</em>, etc.).</td>
</tr>
<tr>
<td><strong>Additional Information:</strong> See the Transaction Locking and Row Versioning Guide and Lock Modes articles on Microsoft Docs.</td>
<td></td>
</tr>
<tr>
<td>Lock Type</td>
<td>Points to the lock type, such as a page or object.</td>
</tr>
<tr>
<td><strong>Additional Information:</strong> See the Lock Granularity and Hierarchies section of the Transaction Locking and Row Versioning Guide on Microsoft Docs.</td>
<td></td>
</tr>
<tr>
<td>Object</td>
<td>The object involved in the deadlock, such as a table, index, or view name.</td>
</tr>
<tr>
<td><strong>Additional Information:</strong> See the sys.all_objects (Transact-SQL) article for more information about objects and the Lock Granularity and Hierarchies section of the Transaction Locking and Row Versioning Guide on Microsoft Docs.</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Index associated with the lock (if applicable).</td>
</tr>
<tr>
<td>Wait Resource</td>
<td>The resource associated with the deadlock. This could be the exact SQL Server page of data, for example.</td>
</tr>
</tbody>
</table>

**Deadlock XML**

Select **DEADLOCK XML** to view the raw XML file for the selected deadlock. Use the **Copy** button to copy all the XML text to your clipboard.
DEADLOCK XML

<transaction>
  <transaction-id>1028</transaction-id>
  <transaction-status>COMMITTED</transaction-status>
  <transaction-isolation>SERIALIZABLE</transaction-isolation>
  <transaction-locks>
    <lock-type>SHARE</lock-type>
    <lock-object>1555555555</lock-object>
    <lock-state>CLOSED</lock-state>
    <lock-owner>user1</lock-owner>
  </transaction-locks>
</transaction>

BEGIN TRANSACTION

UPDATE T SET S = 1 WHERE T = 1

UPDATE DELAY '00:00:00';

UPDATE T SET S = 99 WHERE T = 99

COMMIT

-- Press enter to continue
Monitor TempDB

Last Modified on 23 July 2020

 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.

Overview

The performance of the TempDB system database is critical to your overall SQL Server performance. Things like temporary tables, work tables, the version store, sort operations, index rebuilds (when using SORT_IN_TEMPDB) all rely on the TempDB database.

The TempDB view in SentryOne provides you with insight into what is using your TempDB database and how effectively it is being used. In conjunction with other TempDB monitoring in SentryOne, you can be sure that you have optimized both the TempDB configuration and your applications that are using it.

 Additional Information: See the SentryOne blog post Be Mindful of SQL Server TempDB Use (aka TempDB Parasites!) for examples of things that use TempDB resources.

TempDB Summary

The TempDB Summary is the default chart when opening the TempDB view. It provides an overview of file space usage across your TempDB data files. This chart helps you see how space is being used between uniform and mixed extents. The values shown are in MB.

TempDB Summary Metrics

Version Store

The total space reserved for the version store. When using snapshot isolation, the old versions of the rows are stored here until they can be cleaned up by SQL Server. Trigger activity also uses the version store.
**Internal Objects**

The total space allocated for **internal objects** (e.g. work tables, work files, etc.) across all TempDB data files. This space is within uniform extents and includes all allocated space, even if it is unused.

**User Objects**

The total space for **user objects** (e.g. table variables, temporary tables, etc.) in the TempDB database from uniform extents. This includes space that has been allocated, but is unused.

**Mixed Extents**

The total space that has been allocated to mixed extents across all TempDB data files. The pages in a mixed extent can be shared and owned by different objects in SQL Server, but they do not get allocated for the version store.

**Free Space**

The total amount of free space across all TempDB data files. This doesn’t include unused space that is allocated in an extent.

See the following articles to learn more about the metrics on the **TempDB Summary** chart and the differences between uniform (dedicated) and mixed extents:

- *sys.dm_db_file_space_usage* article on Microsoft Docs
- Pages and Extents Architecture Guide on Microsoft Docs
- Inside the Storage Engine: Anatomy of an extent on SQLskills

**TempDB Objects**

The **TempDB Objects** charts provide a breakdown of the various TempDB objects by type (User Tables, User Temp Tables, Global Temp Tables, Internal Tables, System Tables, User Tables, and Query Objects).
TempDB Objects Chart Options

Note: Use the Objects menu to view the TempDB Objects chart by Reserved Space, User Space, Row Count, or Object Count.

Additional Information: See the Tables article on Microsoft Docs to learn more about these table objects.

TempDB Activity

This chart provides a high-level view of the type and level of activity occurring in tempdb.

TempDB Activity Metrics

Active Temp Tables

The number of active temp tables (system or user-generated) that exist.

Non-snapshot Version Transactions

The number of active transactions that are using the TempDB version store, but are not part of Read
Committed Snapshot Isolation (**RCSI**). The **Non-snapshot Version Transactions** metric paints an overall profile of the server workload.

**Note:** If you’re not running **RCSI** or doing any snapshot level isolation on a system, this value is usually caused by triggers. Triggers use snapshot isolation, so this can be a top source of TempDB consumption if you have a lot of triggers.

**Additional Information:** See the How Does SQL Server 2019 Accelerated Database Recovery Affect TempDB and TempDB Parasites blog posts for examples of triggers using TempDB.

### Snapshot Transactions

The number of active transactions that are using the TempDB version store as part of **RCSI**. This is an explicit representation of **RCSI** or snapshot isolation overhead.

**Additional Information:** See the SQL Server, Transactions Object article on Microsoft Docs to learn more about the **TempDB Activity** metrics.

### TempDB Session Usage

**Note:** **TempDB Session Usage** must be enabled through Settings.

Go to Settings Pane ➔ SQL Server ➔ Miscellaneous ➔ Collect Session TempDB Usage. Set this option to **True**. This may be done at the global, site, or target level.

This chart shows details related to the number of sessions that were using space in TempDB, as well as how much of TempDB was in use, grouped by the **Host**, **Application**, and **Login**. Use this information to understand if there were a small number of sessions with each consuming a large amount of space, or many sessions, each consuming a small amount of space that caused a large amount of consumption when aggregated.
Note: The values in the TempDB Session Usage table represent the aggregated consumption over the time period in the top navigation bar, or highlighted time period from the TempDB Activity chart. Highlight a duration on the chart and select Filter from the context menu to zoom into a specific period of time.

### TempDB Session Usage Metrics

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>The server or workstation name associated with the session(s).</td>
</tr>
<tr>
<td>Application</td>
<td>The name of the application associated with the session(s).</td>
</tr>
<tr>
<td>Login</td>
<td>The login name associated with the session(s).</td>
</tr>
<tr>
<td>Total TempDB</td>
<td>Aggregate of all TempDB allocations (in MB) that occurred during the defined range.</td>
</tr>
<tr>
<td>Active TempDB</td>
<td>Aggregate of allocations (in MB) that were active during the defined range. This removes deallocated usage from the total to provide the active consumption for the associated session(s) by showing only the allocations since the last collection of metrics.</td>
</tr>
<tr>
<td></td>
<td>The maximum amount of memory granted (in MB) for the associated session(s) during the defined range.</td>
</tr>
</tbody>
</table>
### TempDB Session Usage Metrics

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Granted Mem</td>
<td>Insufficient memory grants are a common cause of spills to TempDB. The insufficient memory grants are caused by poor estimates, which may be caused by inaccurate statistics, missing indexes, and similar scenarios. Consider using Plan Explorer to learn more about your queries, including estimated vs. actual plans and indexes and statistics.</td>
</tr>
<tr>
<td>Total Time</td>
<td>Aggregate of all time spent on the associated session(s) during the defined range.</td>
</tr>
<tr>
<td>Total CPU</td>
<td>Aggregate of all scheduled CPU time (in milliseconds) for the associated session(s) during the defined range.</td>
</tr>
<tr>
<td>Total Reads (L)</td>
<td>Aggregate of logical reads completed for the associated session(s) during the defined range.</td>
</tr>
<tr>
<td>Total Writes (P)</td>
<td>Aggregate of physical writes completed for the associated session(s) during the defined range.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The writes may be coming from operations or snapshot isolation which explicitly use TempDB. You can correlate this value against other metrics (e.g. Snapshot Transactions, Non-snapshot Version Transactions, Version Store, and Query Objects) to get a better picture of your Total Writes (P). If the writes are coming from queries that are unintentionally spilling to TempDB, these are often query tuning opportunities.</td>
</tr>
<tr>
<td>Total TempDB User</td>
<td>Aggregate of all space (in MB) reserved or allocated for user objects (e.g. table variables, temporary tables, etc.) by the associated session(s) and task(s) during the defined range.</td>
</tr>
<tr>
<td>Total TempDB Internal</td>
<td>Aggregate of all space (in MB) reserved or allocated for internal objects (e.g. work tables, work files, etc.) by the associated session(s) and task(s) during the defined range.</td>
</tr>
<tr>
<td>Active TempDB User</td>
<td>Aggregate of all space (in MB) reserved or allocated for currently active user objects (e.g. table variables, temporary tables, etc.) by the associated session(s) and task(s) during the defined range.</td>
</tr>
<tr>
<td>Active TempDB Internal</td>
<td>Aggregate of all space (in MB) reserved or allocated for currently active internal objects (e.g. work tables, work files, etc.) by the associated session(s) and task(s) during the defined range.</td>
</tr>
<tr>
<td>Session Count</td>
<td>The number of sessions associated with this aggregation during the defined range.</td>
</tr>
</tbody>
</table>

See the following articles on Microsoft Docs for additional information about the TempDB Session Usage metrics:

- `sys.dm_exec_sessions`
- `sys.dm_db_session_space_usage`
- `sys.dm_db_task_space_usage`
**TempDB Data Retention**

The counter and session data collected for TempDB in SentryOne follows the standard Performance Analysis Dashboard Retention & Resolution data retention policies.

**Additional TempDB Monitoring**

You have access to additional TempDB monitoring through the following features in the SQL Sentry client.

**Top SQL**

Top SQL in the SQL Sentry client allows you to view **Tempdb Internal (KB)**, **Tempdb Internal (KB) Dealloc**, **Tempdb User (KB)**, and **Tempdb User (KB) Dealloc** metrics within the grid.

*Note:* You can highlight an area of activity and use the context menu to jump to Top SQL within SentryOne Portal for that time to get more information about queries and wait stats.

**Disk Space Analysis**

The **Disk Space** tab shows a high-level status of TempDB consumption, including the number of TempDB files.

**Disk Activity Analysis**

The **Disk Activity** tab shows a high-level status of TempDB activity.

**SQL Server Metrics**

Many SQL Server metrics on the dashboard can be correlated to TempDB as explained in the SQL Sentry...
Advisory Conditions

The default advisory conditions pack includes the following TempDB conditions to allow you to create alerts related to the size and number of TempDB files:

- Tempdb Data Files
- Tempdb Large Version Store
- Tempdb Low Unallocated Page Count
- Tempdb Unequal File Size
- Tempdb/CPU Configuration Warning
- Tempdb/CPU Configuration Warning > 8 CPUs

Reporting

The TempDB metrics collected are available via the Performance Counter History report in SentryOne Reporting.

Report example:
What is SentryOne Portal?

SentryOne Portal is a browser-based option for accessing your SentryOne environment data that uses your existing SentryOne database. It replaces the previous mobile applications and Cloud Sync options.

☑ Additional Information: See the SentryOne Portal article.

Prerequisites

Before installing SentryOne Portal on-premises, ensure your credentials and machine(s) meet the System Requirements as well as the security and additional requirements listed below:

Security

- Integrated Windows Authentication is available in versions 2020.8.31 or later. It is not available in version 2020.8.
- SQL Server Authentication is available in all versions.
  - The SQL Server account must have read, write, and execute access to the SentryOne database.
- Accounts logging into the SentryOne Portal through a browser must have access to the Windows Server hosting SentryOne Portal.

Additional Requirements

- Chrome and Edge are the recommended browsers for using SentryOne Portal.
- Configured SentryOne database (version 20.0 or greater) that’s accessible by the web server
  - For information about configuring a SentryOne database, see the SentryOne Installation article.
- Latest SentryOne.Monitor.WebClient.Web artifact or the latest version of the Enterprise Platform Installer
- The preferred IP address and port that SentryOne Portal should use to listen for HTTP traffic

Note:

- If you plan to change the binding address or port, ensure that there isn’t already something listening to that address and port on the machine.
- The default IP address is 0.0.0.0. SentryOne Portal listens to all IP addresses on the machine that are not listening to the selected port.
- The default port is 9991.
- It’s recommended to set the IP address to 127.0.0.1 if you’re planning to route requests.
through IIS or other reverse proxy on the same machine as the service. This will prevent external requests from directly reaching the service.

Installing SentryOne Portal

SentryOne Portal may be installed via the classic SentryOne Setup Wizard or through the EPI commands, as long as the method you choose matches your existing SentryOne installation.

Where can SentryOne Portal be Installed?

SentryOne Portal can be installed on-premises with a self-hosted configuration as a service.

Note: SentryOne Portal cannot run as an Internet Information Services (IIS) site. IIS may only be used as a reverse proxy to the SentryOne Portal service for SSL and request filtering. See the IIS Reverse Proxy Configuration section below for details.

Install SentryOne Portal using Setup Wizard

Follow the instructions in the SentryOne Installation article.

Install SentryOne Portal using EPI

Follow the Installation, Upgrade, and Uninstall instructions in the SentryOne Enhanced Platform Installer article.

SentryOne Portal Configuration Utility

Changes to your SentryOne Portal configuration must be made through the Portal Configuration Utility (PCU).

Note: For the EPI version of SentryOne, the Portal Configuration Utility is only available in versions 2020.8.31 or later. Earlier EPI releases must uninstall/reinstall via command line to make changes.

Accessing the PCU

1. Navigate to the MonitorPortal directory in your SentryOne installation. The default path is C:\Program Files\SentryOne\MonitorPortal. In this example, it is C:\Program Files\SentryOne\2020.0\MonitorPortal.
SentryOne EPI

1. Use the `so configmp` command to launch the Portal Configuration Utility from Command Prompt.

   **Note:** You must run this command on the machine where SentryOne Portal is installed.

Using the PCU

The PCU allows you to change database, network, security, and web server binding-related properties for SentryOne Portal. Select the Verify & Save button to apply any changes.

The PCU also provides an option to stop/start the SentryOne Portal service (SentryOneMonitorPortal in Windows Services).

**Additional Information:** For more information about the settings in the Advanced Properties:

- **Port:** See the Setting the Connection Properties topic from Microsoft Docs for more information on the `portNumber` property.
- **Packet Size:** See the Configure the network packet size Server Configuration Option topic from Microsoft Docs for additional information on SQL Server network pack sizes.
Use SSL
To use SSL for SentryOne Portal:

1. Select the box next to **Use SSL**. Once selected, you'll see the **SSL Certificate** section.
2. Enter the name of the certificate in **Subject**.
3. Select **Verify & Save**.
4. The **Messages** section displays the progress. Note that the SentryOne Portal service will be restarted during this process.

**Success:** You have enabled SSL for SentryOne Portal. Use **HTTPS://** at the beginning of the URL to open it in your browser.

**Note:**
- For a signed certificate from a trusted authority, you must register it on the machine so it goes into the **LocalMachine/My** store. **Additional Information:** See the System Store Locations and Local Machine and Current User Certificate Stores articles on Microsoft Docs for details.
- When updating a certificate, you need to add it to the machine. SentryOne Portal will use the...
latest valid certificate (by expiration date) without requiring a restart of the machine or service. Older, invalid, and expired certificates will be ignored.

- If you do not have IIS installed and are not using port 443 on this machine as part of any other web server, you can update the **Port** in the **Binding** section to **443**. When SentryOne Portal uses **port 443**, you do not need to specify the port in the URL. For example, you can use **https://localhost** instead of **https://localhost:443**.

### IIS Reverse Proxy Configuration (Optional)

**Unsupported:** The following steps cover the process required to set up IIS as a reverse proxy to the SentryOne Portal service for SSL and request filtering. For information about IIS administration, see [IIS.net](#).

This information is provided as an example to get you started with IIS Reverse Proxy Configuration. Please refer to the official IIS administration documentation for support with this process and up-to-date documentation.

See the **Use SSL** option in the **Portal Configuration Utility** section for the preferred method of enabling **HTTPS/SSL** in SentryOne Portal.

### IIS Reverse Proxy Prerequisites

The following modules must be installed before configuring your reverse proxy:

- Microsoft Advanced Request Routing 3.0
- Microsoft URL Rewrite 2.0

**Note:** These required modules are not installed by default.

### IIS Reverse Proxy Instructions

Configure a reverse proxy in IIS to host SentryOne Portal by completing the following steps:

1. Create a website with your desired outward bindings. If you want to use HTTPS, this is where you will register your SSL certificate. Point the site to the default IIS directory.

   **Note:** The default IIS directory is often **C:\inetpub\wwwroot**. The Application Pool settings won't have an effect on the behavior of this site because it will not be executing code. You can set the .NET CLR version to **No Managed Code**, but this is not required.
2. Open the Home window for the new site, and select the **URL Rewrite** feature.

3. Select the **Add Rule** action from the right window pane, and then select **Reverse Proxy rule** from the Inbound and Outbound Rules category.

4. Enter the IP address and port of the service in the **Inbound Rules** server name input. Ensure that **Enable SSL Offloading** is selected. Select **OK** to save the rule.

**Note:**
- *Localhost:9991* is the default IP address. When you are setting this up, you may need to use your server’s DNS name (e.g. *ServerDNS:9991*).
- If your server has no IIS conflicts with port 443, you can bind SentryOne Portal to port 443, and use `https://ServerDNS` as the URL (no port required).

**Success:** IIS now routes all requests to the website to the SentryOne Portal service.

### DEPRECATED: Installing SentryOne Portal On Premises Manually

⚠️ **Important:**

- SentryOne Portal must be installed outside of the user’s directory.
- These steps are included only for users who do not have the current release of SentryOne with the installation built in. It was added to the installer in version 2020.8. If you have version 2020.8 or later, you must use the EPI or Setup Wizard instructions to install it.

After you have ensured that your machine meets the prerequisite requirements, you can begin installing SentryOne Portal. Install SentryOne Portal on your machine manually by completing the following steps:

1. Unzip the **SentryOne.Monitor.WebClient.Web** artifact at the desired location on the machine hosting SentryOne Portal.
2. Update the `appSettings.json` file in the root of the project with the correct connection string for the client’s database.

3. Run the service install script `OnPremServiceInstall.ps1` as administrator. Enter the IP address and port that you want to bind the service to in the appropriate prompt.

⚠️ **Note:** To use integrated authentication for the database connection, you need to change the account that the SentryOne portal service is running under in the Windows Services Control panel after installation. The default account is `LocalSystem`.

⚠️ **Note:** You can re-run the service install script `OnPremServiceInstall.ps1` at any time to change the IP address or port and update the service. The SentryOne Portal service will be turned off during the script’s execution.
Introduction

What is SentryOne Portal?

SentryOne Portal is a browser-based option for accessing your SentryOne environment data that uses your existing SentryOne database. It replaces the previous mobile applications and Cloud Sync options.

SentryOne Portal shares the same UI experience as the cloud-based SentryOne Monitor product. See the Using SentryOne Monitor article to familiarize yourself with the overall layout.

Note:
- Chrome and Edge are the recommended browsers for using SentryOne Portal.
- The configuration client referenced in the SentryOne Monitor documentation does not exist for SentryOne Portal users. You will still use the SentryOne client for any of those cases.

Are all target types supported with SentryOne Portal?

SentryOne Portal currently supports the following target types:

- SQL Server (including Amazon RDS)
- Azure SQL Database
- Windows

How do I install SentryOne Portal?

See the SentryOne Portal Configuration article. SentryOne Portal may be installed via the classic SentryOne Setup Wizard or through the EPI commands, as long as the method you choose matches your existing SentryOne installation.

How do I open SentryOne Portal?

Once you’ve installed SentryOne Portal, you can open it by completing the following steps:

1. Enter your domain for the SentryOne Portal in your preferred browser.

Note: In this example, the SentryOne Portal is bound to localhost, http://localhost/
2. Enter your authentication credentials to access the SentryOne Portal. Use your `DOMAIN\Username` if needed.

   ![Windows Security dialog](image)

   **Note:** Accounts logging into the SentryOne Portal must have access to the Windows Server hosting the SentryOne Portal.

3. The browser opens the SentryOne Portal. See the Using Monitor article for help navigating SentryOne Portal's features.

   ![SentryOne Portal](image)

Which features are available in Portal?

In general, the SentryOne Portal offers the same features available in SentryOne Monitor. It does not have feature parity with the SentryOne client for SQL Sentry. By offering a simpler view, it creates an option that can be more attractive for different stakeholders who don’t need the depth of information and functionality.
offered in the client, or as more of an everyday view for DBAs who may only need to open the client for more advanced troubleshooting.

**Note:** Remember, you still have access to all the additional SentryOne features through the SentryOne client.

### Health Views

The **Health** view is the default view when opening SentryOne Monitor. This screen displays wedges for **Alerts by Severity** and **Alerts by Tag** on the default view for **All Servers**.

**Additional Information:** See the Health Views article.

### Top SQL

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.

**Additional Information:** See the Top SQL article.
Alerts

The Alerts view displays a list of all the conditions that have evaluated to True.

Additional Information: See the Alerts article.

Blocking

The Blocking view maps the relationships between all blocking and blocked sessions (SPIDs) in a blocking chain, allowing you to pinpoint the cause and fix the blocking issue.

Additional Information: See the Blocking article.

Deadlocks
The **Deadlocks** view provides details about deadlocks within your monitored environment. Use it to identify and fix deadlock issues on your monitored servers.

- **Additional Information:** See the **Deadlocks** article.

---

**TempDB**

The **TempDB** view displays a wealth of information about what is using your tempdb database and how effectively it is being used. This feature is specific to SentryOne Portal and is not available in the SentryOne client.

- **Additional Information:** See the **TempDB** article.

---

**Dashboards**

The **Dashboard** view displays the performance charts for monitored targets.

- **Additional Information:** See the **Dashboards** article.
Custom Dashboards

SentryOne Portal 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. This feature is specific to SentryOne Portal and is not available in the SentryOne client.

☑️ Additional Information: See the Custom Dashboards article.

☑️ Additional Information: The SentryOne Portal feature page contains a feature overview.