SolarWinds SQL Sentry Portal

Created on: 30 January 2023
Table of Contents

SQL Sentry ........................................................................................................................................... 4

Installation ........................................................................................................................................... 4
  SQL Sentry System Requirements ................................................................................................. 4
  Installing SQL Sentry .................................................................................................................... 10

SentryOne Install Options ..................................................................................................................... 12
  SQL Sentry Installation ................................................................................................................... 12

SQL Sentry Portal ................................................................................................................................. 25
  SQL Sentry Portal Configuration ..................................................................................................... 25
  SQL Sentry Portal Distributed Databases ....................................................................................... 38
  Getting Started with SQL Sentry Portal ........................................................................................ 43
  Using SQL Sentry Portal ................................................................................................................ 53
  SQL Sentry Portal Custom Charts .................................................................................................. 58
  SQL Sentry Portal Health ................................................................................................................ 69
  SQL Sentry Portal Performance Analysis ....................................................................................... 76
  SQL Sentry Portal Storage ............................................................................................................... 90
  SQL Sentry Portal Top SQL ............................................................................................................ 99
  SQL Sentry Portal Alerts Log .......................................................................................................... 111
  SQL Sentry Portal Blocking ........................................................................................................... 116
  SQL Sentry Portal Deadlocks ......................................................................................................... 123
  SQL Sentry Portal TempDB ............................................................................................................ 128
  SQL Sentry Portal VMware Performance ..................................................................................... 137
  Portal Security ............................................................................................................................... 139
SQL Sentry System Requirements

SQL Sentry Components

For information about the SQL Sentry components, see the SQL Sentry Components and Architecture article. The Installation Recommendations article provides details on where to install each component, as well as detailed implementation examples for installations of various sizes.

⚠️ Important: For performance reasons, it’s not recommended that the SQL Sentry client, monitoring service, or SQL Server (including the instance housing the SQL Sentry database) run simultaneously on the same computer.

SQL Sentry Client Machine

- Windows version from supported list below
- Microsoft .NET 4.7.1 (included in the setup package)
  - Microsoft .NET 4.7.2 is supported and recommended
- Minimum single 1.6 GHz CPU, 1 GB RAM

SQL Sentry Monitoring Service Machine

- Windows version from supported list below
- Microsoft .NET 4.7.1 (included in the setup package)
  - Microsoft .NET 4.7.2 is supported and recommended
- Minimum 4 cores 2.0+ GHz and 12 GB RAM
  - ⚠️ Important: These are the minimum requirements for a standard 5 target installation. Review the Installation Recommendations article for detailed guidance on requirements.

⚠️ Note: The Microsoft .NET 4.7.1 (included in the setup package) involves a system restart.

SQL Sentry Database Machine

⚠️ Important: SQL Sentry version 2021.12 or newer does not support hosting the SQL Sentry Database on SQL Server 2012 (Standard, BI, and Enterprise).

- SQL Server 2014 (Standard, BI, and Enterprise)
- SQL Server 2016 (Standard, BI, and Enterprise)
• SQL Server 2017 (Standard, Linux OS, and Enterprise)
• SQL Server 2019 (Standard, Linux OS, and Enterprise)
• Azure SQL Database Managed Instance
• Supported versions of SQL Server running on AWS RDS for SQL Server
• Minimum 4 cores 1.6 GHz, 8 GB RAM, and 10 GB storage
  ▶️ **Important:** These are the minimum requirements for a standard 5 target installation. Review the [Installation Recommendations](#) article for detailed guidance on requirements.

**Note:** SQL Sentry no longer supports hosting the SQL Sentry database on SQL Server 2008 and SQL Server 2008 R2.

**Note:** Azure SQL Database isn't currently supported as a host for the SQL Sentry database.

**Note:** SQL Server Express Edition may be used to host the SQL Sentry database. The scale limits placed on this edition make it suitable for smaller environments where only a few targets are being monitored. Having an unusually high number of databases or disks to monitor on your targets may mean reaching these limitations with only a few targets.

**SQL Sentry Portal Service Machine**

Installing the SQL Sentry Portal is optional, but we highly recommend that you install the SQL Sentry Portal and see everything it has to offer. Review the [SQL Sentry Portal Configuration](#) article for additional requirements, including security.

• 64-bit Microsoft Windows Server 2012 or greater
• Microsoft .NET Core 3.1 Runtime
  ▶️ Install the Runtime and Hosting Bundle if you are hosting in IIS
• Microsoft .NET Framework 4.8 Runtime

**Note:** Both Microsoft .NET Runtime prerequisites need to be installed on the Windows Server that will host SQL Sentry Portal.

**Supported Operating Systems (SQL Sentry Components)**
Supported Operating Systems (x64)

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022
- Windows 8.1
- Windows 10 (Anniversary update, Creators update, or Fall creators update)

**Note:** For more information, see [NET Framework system requirements](#) for additional details on 4.7.1 requirements. Windows server 2008 is no longer supported.

Watched Targets and Instances

**Important:** SQL Sentry version 2021.12 or newer does not support Windows Server 2008, Windows Server 2008R2, Windows XP, Windows Vista, Windows 7, or Windows 8 targets or instances.

Watched (monitored) Windows Targets

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016
- Windows Server 2019*
- Windows Server 2022
- Windows 10

**Important:** *Windows Server 2019 may have some issues honoring firewall rules. You may need to explicitly enable a Remote Event Log Management rule in the firewall to watch a Windows Server 2019 target.

Watched Data Warehouse Targets

- Microsoft analytics platform system (AU3)
- Azure SQL Data Warehouse

Watched (monitored) SQL Server Instances

- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017
- SQL Server 2019

Additional Information: See the SQL Sentry for SQL Server on Linux page for details on monitoring SQL Server on Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and Ubuntu.

Watched (monitored) SQL Server Analysis Services, Tabular and Multidimensional Modes

Important: SQL Sentry version 2021.12 or newer does not support SQL Server 2005, SQL Server 2008, or SQL Server 2008R2 targets.

- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017
- SQL Server 2019

Watched (monitored) Azure SQL Server Instances

- Azure SQL Database V12 or higher
- All service tiers
- Azure SQL Database Managed Instance

Note: When monitoring a Windows cluster with Performance Analysis for Windows, it’s recommended to monitor cluster nodes individually.

Watched (monitored) Amazon Web Services (AWS) Instances

- Amazon Relational Database Service (RDS) for SQL Server Instance
- Amazon Elastic Compute Cloud (Amazon EC2) Instance
Features with Additional Requirements

- SQL Sentry Portal has additional requirements as outlined in the SQL Sentry Portal Configuration article.
- Storage Forecasting requires a SQL Sentry database on SQL Server 2016 or later with R/ML Services in-Database enabled for Advanced Disk Forecasting. There are no additional requirements for Standard Disk Forecasting.
  
  **Important:** For Advanced Disk Forecasting, you must turn on the external scripts enabled option in SQL Server for the instance hosting the SQL Sentry database. See the External Scripts Enabled server configuration option article in Microsoft Docs for instructions on setting that option. The SQL Server Launchpad service must also be running before using Advanced Disk Forecasting. See the SQL Server Launchpad service configuration article from Microsoft Docs for more information. If these were not in place, you should restart the SQL Sentry client after setting them and then select the Advanced Disk Forecasting mode.

- Deadlock tab and associated data in Performance Analysis for SQL Server is available on SQL 2005 and higher.
- Execution plan collection requires SQL 2005 SP2 or higher.
- Monitoring analysis services with Performance Analysis requires SQL Server 2005 or higher.
- Indexes tab and Fragmentation manager require SQL Server 2005 or higher.
- Monitoring the Windows event log with Performance Analysis for Windows is only supported for Windows vista or higher.
- Hyper-V virtualization is supported for Hyper-V v2 (included with Windows Server 2012).
- Monitoring VMware hosts requires vCenter 5.5, vCenter 6, and vCenter 6.5 to get vCenter-related metrics.
  
  *vCenter 6.5 is only supported by SQL Sentry in v11.2 or higher.*

- Monitoring Azure SQL database targets requires access to the SQL endpoint (over outbound port 1433) with the appropriate Azure SQL firewall rules created.

**Note:** Windows Vista introduced Task scheduler 2.0. Task scheduler 2.0 is backwards compatible with Task scheduler 1.0; however, Task Scheduler 1.0 isn’t forwards compatible with Task Scheduler 2.0. To watch or synchronize Task scheduler 2.0 instances, you must have a SQL Sentry monitoring service and SQL Sentry client running Windows Vista or higher. Windows 8 and Windows 2012 also introduced changes to Task scheduler. In order to watch or synchronize Windows 8 and Windows server 2012 targets, you must have a SQL Sentry monitoring service and SQL Sentry client running Windows 8 or Windows 2012.

**Warning:** We do not support Synchronous-commit mode for the Availability Group hosting the SQL Sentry database in large or high latency network environments. See the Recommendations article for more information.
Installing SQL Sentry

Download

- Purchase or download a trial of SQL Sentry.

Prerequisites

1. Review the Installation Recommendations article to determine where you should install the product components and how many monitoring services you will need.
2. Ensure that your system meets all System Requirements for installation and using additional features.
   - Have questions about security requirements and options? Check out the Security Overview article.

Installation

Feature Availability: The SolarWinds Platform Installer for SQL Sentry is available in version 2022.4.

Each article includes instructions for the full product installation, SQL Sentry Portal, additional monitoring services, or additional clients (i.e. how to install only the client on workstation computers/laptops).

1. Choose an installation method:
   - **Option 1:** Original SQL Sentry Installation for smaller environments (i.e. estates with a single monitoring service or client) and those who do not need a command line experience for streamlining the installation or upgrade processes.
   - **Option 2:** SQL Sentry Enhanced Platform Installer increases the speed and simplicity of installing and upgrading SQL Sentry implementations through a command line interface and is recommended for larger estates that will have more than one monitoring service or client.
   - **Option 3:** SolarWinds Platform Installer for SQL Sentry allows you to deploy, manage, and upgrade SQL Sentry components through a command line interface from a single location.

2. Once you’ve installed SQL Sentry, follow the Onboarding instructions for a step-by-step guide to completing the wizard and configuring your SQL Sentry environment, including adding your first target.
   - You’ll be asked to apply your license during onboarding. See the License Management article for help with applying and managing licenses.

>Note: If you need additional help with getting your SQL Sentry installation up-and-running, please reach out to our support team through: support.sentryone.com. Our friendly and knowledgeable team is here to guide you!
What's Next?

1. After onboarding is complete, you may want to monitor additional targets, or jump into using Performance Analysis, setting up alerts, discovering top SQL queries, or using the calendar view.

2. Interested in setting up SQL Sentry Portal on-premises for browser-based access to your SQL Sentry data? See the Portal Configuration article to enable access.
Overview and Notes

Note: The product has been rebranded since the video below was created, however, the overall process is still the same.

Note: The instructions in this article are recommended for smaller environments (i.e. estates with a single monitoring service or client) and those who do not need a command line experience for streamlining the installation or upgrade processes.

SQL Sentry offers an Enhanced Platform Installer which increases the speed and simplicity of installing and upgrading SQL Sentry implementations through a command line interface and is recommended for larger estates that will have more than one monitoring service or client. To use the EPI installation method, see the EPI article.

Note: If you are upgrading SQL Sentry from a previous version, it's strongly recommended that you back up your SQL Sentry database prior to beginning the process. For more information about upgrading SQL Sentry, see Steps to Upgrade to a New Version of SQL Sentry.

Important: SQL Sentry doesn't currently support Group Management Services (GMS) accounts.
Log into your SolarWinds account, and download the latest build of SQL Sentry. Once you’ve saved the download to your computer, copy the installation file to the server where you want to install the SQL Sentry monitoring service, and then double-click the installation file.

Choosing the Locations of the SQL Sentry Components

Determining the Monitoring Service Computer

>Note: For initial installation, it’s recommended that you first determine where you will install the monitoring service, and then install both the SQL Sentry monitoring service and client together on that computer. This is analogous to installing the native client tool and SQL Server on the same computer.

Even if you don’t plan on using the SQL Sentry client regularly from the machine, the SQL Sentry client is used to enter your license key and enables you to complete the licensing process during the initial installation. You are prompted to launch the SQL Sentry client and enter your license key at the end of the installation.

>Note: For more information about where components are typically installed, see the Installation Recommendations topic.

SQL Sentry Portal

SQL Sentry Portal on-premises is part of the Complete Install. It can be installed on a machine along with the SQL Sentry monitoring service and SQL Sentry client, or it can be installed on a machine by itself without any other SQL Sentry components (Custom Install). Review the SQL Sentry Portal Configuration article before installing.

Once you have gone through the SQL Sentry installation process for the SQL Sentry monitoring service or SQL Sentry client, SQL Sentry Portal cannot be installed separately on the same machine after the installation. It may be installed as part of the initial installation or during a SQL Sentry upgrade.

If you need to install SQL Sentry Portal on a machine that already has the SQL Sentry monitoring service or client of the same version, you must first uninstall the SQL Sentry client and SQL Sentry monitoring service from the machine and reinstall them as part of the SQL Sentry Portal installation. This does not require removing/recreating an existing SQL Sentry database. You will need to choose to connect to the existing database during the installation process.

Choosing the Install Location

Setup shows the default install location. Select Next to use the default location. To install to a different location, select Browse, then select the appropriate location.

Executing the SQL Sentry Application File
1. Execute the **SQLSentrySetup-x64.exe** file. You’ll need to select **Accept and Install**. The **Microsoft .NET Framework** and **Report Viewer** will be included with the SQL Sentry installation (if necessary).

![Image of Microsoft .NET Framework and Report Viewer dialog]

2. A welcome dialog displays when the **SolarWinds SQL Sentry Setup** program starts. Select **Next** to continue or **Cancel** to exit.

![Image of SolarWinds SQL Sentry setup wizard]

3. Then, the **End-user license Agreement** dialog appears. Select the checkbox to accept the terms, then choose **Next** to continue.
**Note:** For future reference, a copy of the license file is located in the Client folder of the installation.

**Note:** If the setup program detects that SQL Sentry is already installed, it prompts for removal. The installation process enables you to easily upgrade from previous versions and maintain all of your existing configuration settings, including any users and groups, notification settings, etc. Any time SQL Sentry is upgraded or another component is installed, the existing software is first uninstalled. This ensures that all components are of the latest version, and compatible. This only applies to the client and server files; the SQL Sentry database, where all of your settings and history are kept, is not removed.

**Important:** The .NET Framework 4.7.1 is required for all installations. A reboot may be required if the .NET Framework files are in use. Temporarily stopping any applications that make use of the .NET Framework can help to avoid a reboot.

4. On the **What would you like to install?** screen, choose an installation option. By default, the **Complete Install** option is selected (and used in this example). It installs the SQL Sentry client, SQL Sentry monitoring service, SQL Sentry Portal, and a PDF copy of the user documentation.
5. On the SolarWinds SQL Sentry Database screen, choose Create a New SolarWinds SQL Sentry Database (default) or Connect to an Existing SolarWinds SQL Sentry Database. (most commonly used for an upgrade or installing an additional SQL Sentry components on other machines). This example uses Create a New SolarWinds SQL Sentry Database.

6. On the Create New SolarWinds SQL Sentry Database screen, choose a location and name for the SQL Sentry database. Enter the server instance where you’d like to install the SQL Sentry database in SQL Server Name. Enter a name for the SQL Sentry database in Database Name. The default Database Name is SQLSentry. The SQL Sentry database is created as part of the installation process. If you’re upgrading SQL Sentry, specify the name of the existing SQL Sentry database (past default SQL Sentry database names have included SentryOne). All the necessary schema changes are applied to the existing database during an upgrade.
The Authentication Mode options available for connecting to the SQL Server that hosts the SQL Sentry database are Windows Authentication and SQL Server Authentication. If the Windows user account you’re using for the installation doesn’t have the required privileges on the selected SQL Server, select SQL Server Authentication and enter a SQL Server User Name and Password for an account with required SQL Server privileges. See the Monitoring Service Security article for the account’s full security requirements.

Select Test to validate the database account information. After a successful test, select Next to continue the setup.

Example uses Windows Authentication & includes SQL Sentry Portal

After selecting Test, a popup will confirm your database selection:

Note: If an existing database has been chosen, selecting Test asks you to confirm that you want to upgrade the database.

7. If you’ve selected Windows Authentication for the Monitoring Service SQL Credentials, enter the Windows account User Name and Password on the Monitoring Service Account Information screen, under which the SQL Sentry monitoring service will run. Select Test to validate the chosen credentials. After a successful test, select Next to continue the setup.

Note: See the Monitoring Service Security article for the account's full security requirements.
After selecting **Test**, a popup will confirm your service account validation:

**Note:** Change the service account any time after the initial installation by running the Service Configuration Utility found in the SQL Sentry program group. See the Monitoring Service Logon Account article for instructions.

8. On the **Portal Configuration Information** screen, choose an **Authentication Mode** for the SQL Sentry Portal service to connect to the SQL Sentry database and enter those credentials. **Note:** It is necessary to enter the credentials on this screen for either authentication option. The **Test** will fail if the credentials are blank.

9. On the **Ready to Install** screen, select **Install** to begin installation.
10. Wait for the installation process to complete, then select **Finish** to complete setup. When SQL Sentry Portal is installed, you'll see the URL at which it has been configured to run. You can launch the Portal Configuration Utility to update any settings, such as the server, database, credentials, encryption, port, or apply a certificate and use SSL.

**Note:**

- If you did a custom install that didn't install SQL Sentry Portal, you won't see the **Launch Portal Configuration Utility** option. You may see an option to launch the **Software Client** instead if it was part of the installation.
- If you didn't install the SQL Sentry client, install it on another machine and run it to enter your license key and complete the licensing process.
Success: You've completed the SQL Sentry installation process! What's next?

- Continue to the Onboarding article to get into SQL Sentry and start monitoring your environment.
  - This is not required for an upgrade. All previous settings have been retained.
- Install additional monitoring services as described below (see the Installation Recommendations article to learn more about how many monitoring services you may need).
- Install the SQL Sentry client on additional machines as described below.

Installing only the Monitoring Service

1. On the What would you like to install? screen, choose Custom Install.

2. To install the SQL Sentry monitoring service on a machine, select only the Monitoring Service option on the Custom Setup screen of the installation process.
Installing Additional Monitoring Services

The number of monitoring services required for your SQL Sentry enterprise depends on the number of targets you are monitoring and other factors related to the overall workload. After completing the installation process, you may want to install additional monitoring services. Please see the Monitoring Services and Targets per Site section of the Installation Recommendations article for guidance on selecting the correct number of monitoring services for optimal performance.

Installing SQL Sentry Portal

The SQL Sentry Portal may be installed on a machine by itself. There’s no need to have a SQL Sentry monitoring service or SQL Sentry client on the machine where it is installed. When installing SQL Sentry Portal by itself, it must be done after a SQL Sentry installation has been completed to create the database and at least one installed monitoring service exists in the overall setup. Follow the instructions in this section to install SQL Sentry Portal by itself and connect to an existing SQL Sentry database.

Additional Information: Review the SQL Sentry Portal Configuration article for additional security requirements and information, how to use the Portal Configuration Utility to update settings after installation, and details on how to access the SQL Sentry Portal in your browser.

To install SQL Sentry Portal complete the following:

1. On the What would you like to install? screen, choose Custom Install.
2. On the **Custom Install** screen, select the drop-down arrow next to the **Portal (Web Client)** component. Select the **Entire feature will be installed on local hard-drive** option.

3. Select **Entire feature will be unavailable** next to the other components if you are installing only the **Portal (Web Client)** service.

4. Follow the **instructions above** for filling out SQL Sentry Portal-related screens.

Installing the SQL Sentry Client on Additional Machines

**Workstation Computers**
There's no need to install a monitoring service on any workstation machines. To install just the SQL Sentry Software Client do the following:

1. On the **What would you like to install?** screen, choose the **Software Client Install** option. Select **Next** to continue.

2. On the **Connect to Existing SQL Sentry Database** screen enter the information needed to connect to your existing SQL Sentry database installation.

3. Select **Test** to validate your connection details, then **Next** to continue installing the **Software Client**.
SQL Sentry Portal Configuration

Note: SQL Sentry Portal requires SQL Sentry software version 20.0 or above. The installation and configuration options are a standard part of the setup and EPI commands in versions 2020.8.7 or later. We strongly recommend upgrading to the latest release to install SQL Sentry Portal and get the complete set of features. See the Release Notes for more information.

What is SQL Sentry Portal?

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

Important: The default URL to access the SQL Sentry Portal is http://localhost:9991.

Additional Information: See the SQL Sentry Portal article.

Prerequisites

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

Security

See the SQL Sentry Portal Security article.
Additional Requirements

- Chrome and Microsoft Edge (the Edge version based on Chromium) are the recommended browsers for using SQL Sentry Portal.
  - Only Windows devices are officially supported.
- SQL Sentry database that's accessible by the web server hosting SQL Sentry Portal.
- The preferred IP address and port that SQL Sentry Portal should use to listen for HTTP traffic.

Notes:

- 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. SQL Sentry Portal listens to all IP addresses on the machine that are not listening to the selected port.
  - If you are running SQL Sentry Portal on a virtual machine, it's recommended to keep the default IP address of 0.0.0.0. Setting it to 127.0.0.1 may make it so that it can be accessed from the local host, but not other locations in the domain.
- 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 SQL Sentry Portal

SQL Sentry Portal may be installed via the classic SQL Sentry Setup Wizard, through the EPI commands, or through the SolarWinds Platform Installer for SQL Sentry as long as the method you choose matches your existing SQL Sentry installation.

Where can SQL Sentry Portal be Installed?

SQL Sentry Portal can be installed on-premises with a self-hosted configuration as a service. It can be installed on a machine along with the SQL Sentry monitoring service and SQL Sentry client, or it can be installed on a machine by itself without any other SQL Sentry components.

Notes:

- When using the EPI version, the SQL Sentry controller must exist on the machine where you install SQL Sentry Portal.

If you have more than one SQL Sentry database, you can view them with a single SQL Sentry Portal service. See the distributed databases article for more information.
as a reverse proxy to the SQL Sentry Portal service for HTTPS and request filtering. See the IIS Reverse Proxy Configuration section below for details.

Installation Example

Example of SQL Sentry (SentryOne) components installed across multiple machines (with EPI components when applicable)

Install SQL Sentry Portal using Setup Wizard

Follow the instructions in the SQL Sentry Installation article.

Install SQL Sentry Portal using EPI

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

Install SQL Sentry Portal Using SolarWinds Platform Installer for SQL Sentry

Follow the installation instructions in the SolarWinds Platform Installer for SQL Sentry article.

SQL Sentry Portal Configuration Utility

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

SQL Sentry

Locate the PCU through the file directory or use the **Windows Start** menu as **Portal (Web Client) Configuration**. Use the Run as administrator option to open it.

**File path:**

1. Navigate to the **MonitorPortal** directory in your SQL Sentry installation. The default path is `C:\Program Files\SolarWinds SQL Sentry\MonitorPortal\PCU`. In this example, it is `C:\Program Files\SentryOne\2020.0\MonitorPortal`.

SQL Sentry EPI Version

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 SQL Sentry Portal is installed.

2. You must use the EPI commands **so stopmp** and **so startmp** after making changes to the configuration. The PCU does not restart the portal service in an EPI environment.
Using the PCU

The PCU allows you to change database, network, security, and web server binding-related properties for SQL Sentry Portal. Select the **Verify Connection** button to verify your connection settings and then select **Save** to apply any changes.

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

**Note:** SQL Sentry Portal supports distributed SQL Sentry databases. The drop-down menu at the top allows you to switch between the settings for each SQL Sentry database.

**Additional Information:** If you have multiple SQL Sentry databases and would like to view all of them in the same SQL Sentry Portal, see the distributed databases article for setup and security details.

**Important:** The default URL to access the SQL Sentry Portal is `http://localhost:9991`.

**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.
- **Encrypt Network Traffic:** See the Enable Encrypted Connections to the Database Engine topic from Microsoft Docs.
- **Trust Server Certificate:** See the Setting the Connection Properties topic from Microsoft Docs for more information on the `trustServerCertificate` property.
- **Multi Subnet:** See the Setting the Connection Properties topic from Microsoft Docs for more...
Use TLS

To use TLS for SQL Sentry Portal:

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

**Success:** You have enabled TLS for SQL Sentry 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. SQL Sentry 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 SQL Sentry 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**.

### Adding New SQL Sentry Portal Connections

1. Select **New** to open the Add New Connection window.
2. Enter a name for the new portal connection, and then select **Confirm**.
3. Enter the Server and Database Name for your connection.
4. Select your authentication method and enter your connection credentials. Select **Verify Connection** to test your connection.
5. Configure any applicable Advanced Properties, Bindings, and the User Identity Provider.
6. Select **Save** to save your Portal Connection.

### Deleting SQL Sentry Portal Connections

1. Select the Portal Connection you want to remove from the SQL Sentry Database Connections drop-down list.
2. Select **Delete** to open the Remove Connection window.
3. Select **Yes** to remove the connection.

### Using Azure Active Directory as the Identity Provider

**Note:** Azure Active Directory (AD) is available as an Identity Provider (IDP) for SQL Sentry Portal Version 2023.1 and later. Azure AD authentication is not fully supported with multiple database installations in Version 2023.1.

**Additional Information:** For information about registering an app with Azure AD, see the Tutorial: Register an app with Azure Active Directory MSDN article.

**Warning:** For users that installed new installations of SQL Sentry Portal with Version 2023.1:

You must execute the following script against the SQL Sentry database to use Azure AD as an IDP for SQL Sentry Portal.

Substitute the email address used in the Onboarding wizard in the '{email address}' field and execute the
--NOTE: A record in [dbo].[Contact] must exist with the supplied email address
DECLARE @ContactID UNIQUEIDENTIFIER;
SELECT @ContactID = [ObjectID] FROM [dbo].[Contact] WHERE [EmailAddress] = '{email address}'
DECLARE @RoleID UNIQUEIDENTIFIER = '7E54B2ED-0BEC-4E83-A279-44E6F9BEF1C1'

INSERT INTO [Security].[FeatureRoleAssignment] (ObjectID, RoleID, PrincipalID)
SELECT s.[ObjectID], @RoleID, @ContactID
FROM [dbo].[Site] as s
WHERE ParentSiteObjectID IS NULL

⚠️ Important: You must configure the TLS certificate to use the Azure Active Directory (AD) Identity Provider. See Use TLS above for information about configuring the TLS certificate in the PCU.

Create a New or configure an existing SQL Sentry Portal Connection to use your Azure AD IDP credentials:

1. Open the SQL Sentry Portal Configuration Utility.
2. Select New to create a new repository connection or select the desired connection from the drop-down menu.
3. Select Azure AD from the Provider drop-down menu.
4. Enter the Tenant id, Client id, Redirect URL, and Client Secret associated with your Azure AD application.
5. Select Save to save your credentials.
IIS Reverse Proxy Configuration (Optional)

Unsupported: The following steps cover the process required to set up IIS as a reverse proxy to the SQL Sentry Portal service for HTTPS 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 TLS option in the Portal Configuration Utility section for the preferred method of enabling HTTPS/TLS in SQL Sentry 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 SQL Sentry 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 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 SQL Sentry 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 SQL Sentry Portal service.

DEPRECATED: Installing SQL Sentry Portal On Premises Manually

Important:

- SQL Sentry 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 SQL Sentry 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.
  - You need the original *SentryOne.Monitor.WebClient.Web* artifact or version of the Enterprise Platform Installer that was used for your SQL Sentry installation to follow these deprecated instructions.

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

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 SQL Sentry 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 SQL Sentry Portal service will be turned off during the script's execution.
Distributed SQL Sentry Databases

SQL Sentry Portal allows you to connect to distributed SQL Sentry databases to create a unified view across multiple SQL Sentry environments using a single SQL Sentry Portal service.

SQL Sentry database servers show on the sidebar menu with a (server icon) at the top of the drop-down hierarchy. In the example below there are two SQL Sentry database servers (SquadB-Dev and SQUADB-STABLE2.SQL Sentry), each representing a different monitoring environment.

Implementation Example
Using Distributed Databases

How are security settings used for distributed databases?

Access is at the SQL Sentry database level. A user must meet the security requirements as described in the SQL Sentry Portal User Access Requirements section of the Portal Configuration article for each SQL Sentry database they need to access. SQL Sentry Portal users will only see the SQL Sentry environments listed for the SQL Sentry database servers where they meet those security requirements.

Do all SQL Sentry databases need to be the same version?

Unsupported: SQL Sentry Portal does not officially support adding SQL Sentry databases with different versions.

SQL Sentry Portal does not prevent you from adding SQL Sentry databases that are on different versions, but you may encounter errors or unexpected issues with this setup. It is strongly recommended that all SQL Sentry databases are on the same version for the best experience.
Are dashboards applied across databases?

Dashboards are per SQL Sentry database. Any custom dashboards that you have added are at the SQL Sentry database level and will not appear for other SQL Sentry databases when using distributed databases.

How is the EHO score displayed?

The Environment Health Overview score displays for a single SQL Sentry database at a time. Select the SQL Sentry database at the top level ( ) to view an overall EHO score for that environment.

How are the alerts displayed?

The alerts are displayed for the last SQL Sentry database environment that you selected.

Configuration

Installation

Install SQL Sentry Portal on a machine that can reach all SQL Sentry databases that you want to add to this view. Only one SQL Sentry Portal service is needed per SQL Sentry Portal URL for multiple SQL Sentry databases.

Adding Distributed Databases

Distributed databases are configured in the Portal Configuration Utility (PCU). All SQL Sentry databases added to this SQL Sentry Portal service will be visible at the same SQL Sentry Portal URL.

1. Open the PCU and select the New button.
2. Enter a friendly name for the **New Repository Name**, then select **Confirm**.

3. Enter the connection information for the SQL Sentry database in this additional environment, then select **Verify Connection**. The connection information in the **PCU** is per database.
Deleting Distributed Databases

1. Open the PCU.
2. Select the distributed database environment to delete from this SQL Sentry Portal view, then select Delete.
3. Select Yes on the Remove Connection window to confirm that you are sure you want to remove this connection.
Introduction

What is SQL Sentry Portal?

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

**Additional Information:** See the Using SQL Sentry Portal article to familiarize yourself with the overall layout.

**Note:** Chrome and Edge are the recommended browsers for using SQL Sentry Portal.

Are all target types supported with SQL Sentry Portal?

SQL Sentry Portal currently supports the following target types:

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

How do I install SQL Sentry Portal?
See the SQL Sentry Portal Configuration article. SQL Sentry Portal may be installed through the classic SQL Sentry Setup Wizard or through EPI commands, as long as the method you choose matches your existing SQL Sentry installation.

### How do I open SQL Sentry Portal?

⚠️ **Important:** Feature Based Security is enabled by default for SQL Sentry Portal Versions 2022.2 and greater. The Windows login(s) used to access your portal environment must be associated with a SQL Sentry User or Group to avoid connectivity issues.

Additional Information: During a new installation, no roles are assigned to default users/groups. You need to assign roles to your users/groups on the SQL Sentry Portal Permissions Page using the Feature Access Grid. For more information about Security in SQL Sentry Portal, see the Portal Security article.

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

1. Enter your domain for the SQL Sentry Portal in your preferred browser.

   ⚠️ **Note:** In this example, the SQL Sentry Portal is bound to localhost, `http://localhost/`. The URL to open SQL Sentry Portal includes the port as well: `. You may have a different port to use with your host name. This information can be viewed or changed with the SQL Sentry Portal Configuration Utility. See the SQL Sentry Portal Configuration article for details.

2. Enter your authentication credentials to access the SQL Sentry Portal. Use your `DOMAIN\Username` if needed.

   ⚠️ **Note:** Accounts logging into the SQL Sentry Portal must have access to the Windows Server hosting SQL Sentry Portal.
3. The browser opens the SQL Sentry Portal. See the Using SQL Sentry Portal article for help navigating SQL Sentry Portal’s features.

4. Select the permissions button to open the Users and Groups page.

5. Select the applicable permissions that you want to assign to your user. Repeat this step for all of your SQL Sentry Portal users.

*Success:* The Role assigned successfully prompt displays when you’ve made changes to the selected user role.
Which features are available in Portal?

The SQL Sentry Portal does not have feature parity with the SQL Sentry 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 SQL Sentry features through the SQL Sentry client.

Health Views

The Health view is the default view when opening the SQL Sentry Portal. This screen displays widgets for Alerts by Severity and Alerts by Tag on the default Overview for all servers. You can drill into targets for an individual view.

Additional Information: For more information about Health Views in the SQL Sentry Portal, see the Health Views article.

Performance
The Performance tab displays performance charts for the monitored target type. For example, the Performance tab for a SQL Server target displays charts such as Network, CPU, System Memory, SQL Server Activity, SQL Server Waits, SQL Server Memory, Disk, and Database I/O.

**Additional Information:** The charts that are available in the Performance tab vary based on the target type selected. See the SQL Sentry Portal Performance article for more information about the Performance tab.

**Storage**

The Storage view provides details about the disk space and disk activity in your monitored environment. The Storage is only available on SQL Sentry Portal and acts as a combined view of the Disk Activity and Disk Space tabs in the SQL Sentry Client.
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.

![Top SQL view](image)

Additional Information: For more information about Top SQL in the SQL Sentry Portal, see the Top SQL article.

Alerts

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

**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:** For more information about Alerts in the SQL Sentry Portal, see the Alerts Log article.

**Additional Information:** For more information about Blocking in the SQL Sentry Portal, see the Blocking 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 SQL Sentry Portal and is not available in the SQL Sentry client.

**Dashboards**

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

SQL Sentry Portal offers the ability to create dashboards with configurable custom charts. With custom charts, you can include the performance metric widgets that you want, and you can have custom charts for different targets on the same dashboard. This feature is specific to SQL Sentry Portal and is not available in the SQL Sentry client.

Additional Information: For more information about Custom Charts in the SQL Sentry Portal, see the Custom Charts article.
Custom Charts article.

Additional Information: The SQL Sentry Portal feature page contains a feature overview.
Overview & Navigation

- SQL Sentry Portal is a self-hosted feature for SQL Sentry which has a similar UI experience. See the Getting Started with SQL Sentry Portal article for information on accessing it.

Default Home View

When you first log into SQL Sentry Portal, 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 expanding the sidebar, selecting your alert timeframe, providing feedback, accessing documentation, and accessing user profile information. Full details are provided under the Navigation Bar section below.</td>
</tr>
<tr>
<td>2. Sidebar</td>
<td>The sidebar has areas for overview options (Health Overview &amp; Alerts) and the monitored Targets. Full details are provided under the Sidebar section below.</td>
</tr>
<tr>
<td>3. Health Overview</td>
<td>Health Overview is the default view when opening SQL Sentry Portal.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Health Score</strong> shows a summary of overall health scores for the monitored environment.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Alerts by Severity</strong> 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>- <strong>Alerts by Tag</strong> 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>
</tbody>
</table>
### 4. Alerts

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

### 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 overview options and targets information)</td>
</tr>
<tr>
<td>2. SolarWinds SQL Sentry logo</td>
<td>Selecting the SQL Sentry logo returns you to the home (Health Overview) view.</td>
</tr>
<tr>
<td>3. Time Slicer</td>
<td>The Time Slicer is available on the Health Overview at the Global (All Targets) and individual Target levels. The Time Slicer allows you to organize the alerting events displayed on the Health Overview by the time frame in which they occurred (within the last 24 hour interval). Display events that occurred in the last 24 hours (the default setting) or narrow the displayed events to the last few hours, or minutes to identify more recent issues. The following time frames can be selected:</td>
</tr>
<tr>
<td>4. Documentation</td>
<td>Select the button to open expanded menu options:</td>
</tr>
<tr>
<td></td>
<td>- Select Documentation to open the SQL Sentry Portal documentation.</td>
</tr>
<tr>
<td></td>
<td>- Select SolarWinds support to go to SolarWinds.support.com</td>
</tr>
<tr>
<td></td>
<td>- Select Submit a feature request to submit ideas about any issues, features you’d like to see, or suggested changes.</td>
</tr>
<tr>
<td>5. Permissions</td>
<td>Select the Permissions button to manage permissions in SQL Sentry Portal.</td>
</tr>
</tbody>
</table>

### Sidebar

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

**Unwatched Servers**

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

In SQL Sentry Portal, the bottom of the sidebar contains a group for **Unwatched Servers** (similar to Inventory view). Unwatched servers are not currently being monitored by SQL Sentry. Use the SQL Sentry client to watch a target (you may need to purchase additional licenses).
**Feature Menu**

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

The following options are available for individual targets:

1. Health
2. Performance
   - **Performance Analysis** is available in SQL Sentry Portal for VMware targets with Version 2022.3.
3. Storage
4. Top SQL
5. Blocking (SQL Sentry Portal only)
6. Deadlocks (SQL Sentry Portal only)
7. TempDB (SQL Sentry Portal only)

**Distributed Databases**

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

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

Custom charts in SQL Sentry Portal allow you to create personalized dashboards with the performance counters, labels, colors, and chart types you select.

Additional Information: See the Your Performance Data, Your Way with Custom Charts in Portal blog post for more examples.

Viewing Dashboards

Select Dashboards from the sidebar. If dashboards with custom charts already exist, they will be listed here under the Dashboard Digest. Select a Dashboard to view its contents.

Filter Dashboards

If you have a large number of dashboards, you may want to filter what is displayed in the Dashboard Digest. Start typing in the Filter Dashboards box to filter the dashboard options displayed.

Creating Dashboards with Custom Charts
To create a Custom Dashboard in SQL Sentry Portal, complete the following steps:

1. Select **Dashboards** from the sidebar to open the **Dashboard Digest**.
2. Select the **Create Dashboard** button.
3. Enter a **Dashboard Name** on the **Create a New Dashboard** screen. This is how the dashboard will appear on the **Dashboard Digest**.
4. Select a **Dashboard Template** from the drop-down menu. Use **Blank Dashboard** to start fresh or choose an existing dashboard name as a starting point for this dashboard.
5. Select the 📋 Edit button to launch the **Add Charts** options bar.
   - **Note:** Use **Custom Chart** to start fresh with the specific metrics you are looking for, or select an existing group from one of the options (e.g. **Host** ➔ CPU or **SQL Server** ➔ **Database I/O**). If you select an existing chart option, you can still customize it.
6. Add a chart and use the **Customize Chart** screen to make changes with the custom chart options (described below) to create a chart.
7. Select **Save** to save your chart and return to the edit dashboard screen.
8. Add additional charts or select **Save & Exit** to complete your dashboard.
Custom Chart Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preview</td>
<td>The preview area displays your chart with the selected options.</td>
</tr>
<tr>
<td>Chart Name</td>
<td>A descriptive name for the chart. This is how it will be identified on the</td>
</tr>
</tbody>
</table>

*Dashboards view and used for filtering.*
The name of the monitored target to select metrics from for the custom chart. When building your dashboard, you can include charts for different targets and target types on the same dashboard.

**Note:** Targets types are identified with icons in the **Target** drop-down list. When you select the **Target** from the list you will only have access to applicable metrics in the **Performance Metrics** section below. For example, a Windows target has access to the applicable Windows or host metrics and SQL Server or Azure SQL Database targets have access to the applicable SQL Server or Azure metrics. You must be collecting tempdb metrics to have tempdb metrics listed.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target</strong></td>
<td>The name of the monitored target to select metrics from for the custom chart. When building your dashboard, you can include charts for different targets and target types on the same dashboard. <strong>Note:</strong> Targets types are identified with icons in the <strong>Target</strong> drop-down list. When you select the <strong>Target</strong> from the list you will only have access to applicable metrics in the <strong>Performance Metrics</strong> section below. For example, a Windows target has access to the applicable Windows or host metrics and SQL Server or Azure SQL Database targets have access to the applicable SQL Server or Azure metrics. You must be collecting tempdb metrics to have tempdb metrics listed.</td>
</tr>
<tr>
<td><strong>Sample Period</strong></td>
<td>Option to include historical metrics for 1 hour, 1 day, or 1 week.</td>
</tr>
<tr>
<td><strong>Add Metric</strong></td>
<td>Select the <strong>Add Metric</strong> button to add another metric to the custom chart. You may add up to 10 metrics per chart.</td>
</tr>
</tbody>
</table>
### Metric

A drop-down menu of performance counter metrics (\texttt{host.disk.reads}, \texttt{host.network.in.total}, \texttt{sqlserver.batches}, etc.) applicable to the selected \textbf{Target} type.

\textbf{Note:} Start typing to filter the available metrics. For example, type \texttt{cpu} to see all CPU-related metrics or \texttt{tempdb} for tempdb-related metrics.

#### CPU filtered metrics

- os.cpu.maxFrequency.pct
- os.cpu.privilegedTime.pct
- os.cpu.processorTime.pct
- os.cpu.processorTime.sqlnstances.pct
- sqlserver.topsql.applications.cpu
- sqlserver.topsql.databaseapplications.cpu
- sqlserver.topsql.hosts.cpu
- sqlserver.topsql.logins.cpu
- sqlserver.topsql.queries.cpu
- sqlserver.topsql.resources.cpu

#### Tempdb filtered metrics

- sqlserver.tempdb.nonsnapshotversiontransactions
- sqlserver.tempdb.snapshottransactions
- sqlserver.tempdb.tempTables
- sqlserver.tempdbusage.free.size
- sqlserver.tempdbusage.internalobjects.size
- sqlserver.tempdbusage.mixedextentsize
- sqlserver.tempdbusage.userobjects.size
- sqlserver.tempdbusage.versionstore.size

### Instance

The instance of the metric (if applicable) such as \textit{Average} or a specific \textit{Node/Core} for \texttt{host.cpu.cores} or a specific disk for \texttt{host.wmi.disk.size}.

### Legend Label

The text that appears on the chart legend to associate the metric and color.

\textbf{Note:} You can use emoji in the legend label. So if certain metrics make you happy, sad, or angry, you can add a little flair to your legend.
### Color

Select the dot to open the color palette window. To choose a color, use the color slider and select a shade from the rectangle, enter a HEX value, or enter RGB values for your custom color.

With **Version 2022.3** or later, you can use the transparency slider (under the color slider) to adjust the transparency of your selected color.

Select **Set** when you have chosen the appropriate color for the associated metric.

### Chart Type

The following chart type options are available:
- **Line** (default)
- **Area**
- **Column**

**Note:** There’s much to debate when it comes to choosing the right type of chart to visualize your data. Keep in mind that:

- A **line** chart displays data points that are connected with a single line. Its simplicity makes it a versatile and common chart type.
- An **area** chart displays the data points with a connected line and fills in the area below. It may be difficult to see all metrics in an area chart (depending on the data being displayed).
- A **Column** chart displays data in vertical bars, grouped by metric. It can be particularly useful for comparing series of data.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| Line   | Check the **Line** box to display the metric as a line on the chart.  

![Example of a Column chart with Writes and Network In metrics added as a line.](chart.png)  

**Note:** This option only appears on **Area** and **Column** charts. It allows you to add a metric as a line to those chart types. |
| Use the ✖️ button to remove the metric from the chart. |
### Custom Time Range

A **Custom Time Range** allows you to configure a chart to display metrics a defined number of minutes, hours, or days.

![Example chart using a custom time range of 3 hours](image)

#### Note:
You can mix charts with custom time ranges and default time ranges on the same dashboard. Charts that use a custom time range will not be affected when a new time period is selected from the date selector in the top navigation bar.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Amount</strong></td>
<td>The numerical value associated with the <strong>Units</strong>. For example, if you want the past 12 hours displayed, enter <strong>12</strong> in <strong>Amount</strong> and select <strong>Hours</strong> from <strong>Units</strong>.</td>
</tr>
</tbody>
</table>
| **Units** | The following units of time are available:  
  - *Minutes*  
  - *Hours*  
  - *Days* |
| **Save** | Select **Save** to save the changes to your custom chart, and add it to your Dashboard. |
| **Cancel** | Select **Cancel** to dismiss your custom chart and not save any changes. |

### Editing Dashboards

1. Select **Dashboards** from the sidebar.
2. On the **Dashboard Digest** screen you can either:
   1. Select the **options** button on the dashboard you want to edit.
   2. Select the dashboard you want to edit, then select the **edit** button from the **Dashboards** screen.

**Edit** mode allows you to do any of the following:
1. Delete the dashboard.
2. Rename the dashboard.
3. Add additional charts to the dashboard.
4. Configure, delete, or copy existing charts on the dashboard.

Remember to select **Save & Exit** after making your changes, or **Cancel** to dismiss them.

**Organizing Charts**

When you are in **Edit** mode, you can drag and drop the charts to change the order or position on the dashboard. If you want more than one chart per row, drag it into a row that contains another chart to create another chart column.

**Note:** To remove a chart from a multi-column row and place it in a new row, drag and drop it on the bar with the add symbol (plus sign in a circle).
Editing Custom Charts

Adding charts

Selecting ✏ edit on the dashboard allows you to add new charts using the Add Charts option bar. See the Custom Chart Options above to complete adding a chart.

Editing charts

1. Select the ⚙ configure button on the chart you want to edit.
2. Edit the options as needed. See the Custom Chart Options above.
3. Select Save.

Removing charts

1. Select the ✕ edit button on the dashboard that contains the chart you want to remove.
2. Select the ✗ delete button on the chart you want to remove.
3. Select Save & Exit to complete the chart deletion.
   * 🔄 Note: Select Cancel if you do not want to delete the chart.

Copying charts

1. Select the ✏ edit button on the dashboard that contains the chart you want to copy.
2. Select the copy button on the chart you want to copy.
3. Edit the options as needed. See the Custom Chart Options above.
4. Select Save to save your changes.
   * 🔄 Note: Select Cancel if you do not want to copy the chart.
5. Select Save & Exit to save the changes to your dashboard.

Deleting Dashboards

An entire dashboard may be deleted from the Dashboard Digest or Dashboard screen.

From the Dashboard Digest screen:
1. Select the options button on the dashboard you want to remove.
2. Select Delete.
3. Select Delete Dashboard on the Delete Dashboard Confirmation screen to complete the dashboard deletion.
   ◦ Note: Select Cancel if you do not want to delete the dashboard.

From the Dashboard screen:

1. Select the Edit button on the dashboard you want to remove.
2. Select the Delete button on the dashboard you want to remove.
3. Select Delete Dashboard on the Delete Dashboard Confirmation screen to complete the dashboard deletion.
   ◦ Note: Select Cancel if you do not want to delete the dashboard.
Health Overview

Overview

The Overview tab is displayed by default. As indicated in the name, this view is showing the overall health, and the 10 most recent alerts that fall within the selected time frame (within 24 hours) for all monitored targets in your environment.

![Overview Tab](image)

Health Score Calculation

The Health Score (shown above) is calculated by incorporating open alerts and their associated severity. It uses a 24-hour window in the calculation, and can be adjusted with the Time Slicer to display more recent alerts in varying hour and minute increments. A high health score (with a maximum of 100 possible) is an indicator of good health.

The points per severity level are:

- Low - 1 point
- Medium - 3 points
- High - 6 points
- Critical - 9 points

**Note:** The weight of an open alert diminishes with its age. The score being presented is 100 minus the sum of the weighted value of open alerts. Closed alerts are not factored into the score. Alerts can be closed in the Events Log in the SQL Sentry configuration client.
Select a time increment from the Time Slicer to adjust the Health Score, widgets, and associated alerts based on your selection:

![Image of SQL Sentry Portal EHO displaying the last 3 hours data](image)

Alert Details

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

![Details Table](image)

**EHQ for Sites and Groups**

The Environment Health Overview (EHO) is available for sites and groups in SQL Sentry Portal. Use the Show Hierarchy switch to toggle the list of targets between a flat list and a list grouped by sites.
Expand a site to view the groups within it and select a group to view a group EHO.

Example of the EHO at the site level (using Default Site)

Example of the EHO at the group level (using Azure SQL DB)

Note: The Site and Group icons and labels note which type of overview you are viewing.

All SQL Servers View

Select All SQL Servers from the sidebar to display the overall health view for all SQL Servers in your monitored environment.
Note:  Uptime is a measure of the percentage of time that SQL Sentry Portal is able to connect to the server to collect data.

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

All Windows Servers View

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

Target Health View

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

### Alerts by Severity

In the example below, the **Alerts by Severity** widget is displaying **5 Critical** and **9 Medium** severity alerts. In the **Alerts** table, you can see 2 rows of alerts with a severity of **critical** (4, and 1) totaling 5, and 8 rows of **medium** alerts (with one row displaying 2 alerts) totaling 9. This is how the number in the color-coded alert is calculated.

### 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.

---

**Note:** Selecting the blue hyperlinked target name on the left provides an option to switch to the **Health**, **Performance**, **Storage**, **Top SQL**, **Blocking**, **Deadlocks**, or **TempDB** view for that target.
Wait Time / Session

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

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

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

As seen in the image above, there is a list of **Related Targets** (1 SQL Server and 1 Windows Server in this example). When there are multiple instances on a SQL Server, they will be listed as a related target, as will the Windows Server where the SQL Server is installed.
**SQL Sentry Portal Performance Analysis**

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

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

**Performance Navigation**

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

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

1. A drop-down menu to switch between monitored targets.
2. A pause button (which flips to a play button) to toggle the view of live data.
3. A date selector.

**Performance tab Options**

**Date Selector**

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

Hover over a point in a chart to open a tooltip that displays additional details.

Filter Dots

Use the filter dots (Disk, Other, Memory, CPU, and Network in the image below) to customize what appears on the chart from the available options:
The same chart with **Disk** filtered out:

**Zoom**

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

**Jump To**

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

**Performance Analysis Charts**

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

**Windows**

**Network**

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

**CPU Usage**

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

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

SQL Server

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

File cache

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

Other

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

Disk I/O

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

ms/Read

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

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

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

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

ms/Write

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

SQL Server

SQL Server Activity

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

Batches

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

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

**Compiles**

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

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

**Recompiles**

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

**Transactions**

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

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

**Key Lookups**

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

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

**Forwarded Records**

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

**Backup/Restore MB**

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

**SQL Server Waits**

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

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

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

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

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

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

**SQL Server Memory**

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

**Buffer**

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

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

**Plan (Objects)**

The current size of the cache used for object plans (in MB). This includes stored procedures, functions, and triggers. A high value in proportion to the buffer cache may indicate query plans aren't being effectively 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 IO** chart displays information about the read and write latency for the databases.

**ms/Read**

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

**ms/Write**

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

**Log Flushes**

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

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

**Checkpoint Pages**

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

**Lazy Writes**

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

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

**Azure SQL Database**

**Resource Usage**

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

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

**Total DTU %**

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

**Data I/O**

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

**Log I/O**

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

**CPU %**

This metric is the average CPU percentage based on the limit of the service tier. This is one of the metrics that
Memory Usage

Allocated Memory Usage

Each service tier has a maximum amount of memory allowed for the Azure SQL Database to use. This metric provides the percentage of the allowed memory being used for the database.

It will be very common for this metric to be high. If much of the data your applications need is in memory it means better performance because the database doesn’t have to read from the physical disk to return the data.

Database Size

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

Used Space

The space used by the database in MB.

Free Space

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

SQL Database Waits

The SQL Database 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 Database Waits is an important chart on the dashboard, because it provides an instant profile of the Azure SQL DB and where it’s spending the most time waiting for physical resources. If SQL DB waits 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 is effectively handling the load.

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.

Database I/O

The Database IO chart displays information about the read and write latency for the databases.
VMware Host Metrics

Now available: Starting with Version 2022.3, you can view the Performance tab for VMware targets.

Network
The Network chart displays information about the percentage of network bandwidth across the VMHost that is used for downloads (IN) and uploads (OUT).

CPU
The CPU chart displays CPU across your VMHost:

Total CPU
The total amount of CPU used across the VM as a percent.

Co-stop %
Co-Stop time measures when a virtual machine is ready to run, but unable to run due to co-scheduling constraints.

Ready Time %
Percentage of time when a vCPU is ready to do work, but must wait for the hypervisor to schedule that work on one or more physical CPUs. CPU ready time is a summation, and is dependent on the number of virtual machines on the host and their CPU loads.

System Memory
The System Memory chart displays information about the amount of memory being used by different processes on the VMHost:

System Memory Active
Amount of memory that is actively used, as estimated by VMkernel based on recently touched memory pages.

System Memory Consumed
Amount of machine memory used on the host. Consumed memory includes memory used by the service console, the VMkernel, vSphere services, and the total consumed metrics for all running virtual machines.

System Memory Overhead
Total of all overhead metrics (including VM overhead and VMKernel overhead) for powered-on virtual machines, and the overhead of running vSphere services on the host.

System Memory Granted
Amount of machine memory or physical memory that’s mapped for the host.
System Memory Available

Amount of machine memory or physical memory that's available for the host.

System Memory Swap In

The percentage of memory read in by the VMkernel swap file from memory to disk across all the virtual machines on the VMHost.

System Memory Swap Out

The percentage of memory written by the VMKernel swap file from memory across all the virtual machines on the VMHost.

Disk

The **Disk** chart displays the read and write latency for each disk on the VMHost.

Disk ms/Read

The amount of time it takes in milliseconds for the VMHost to read data from each Logical Unit Number (LUN) on the datastore.

Disk ms/Write

The amount of time it takes in milliseconds for the VMHost to write data from each Logical Unit Number (LUN) on the datastore.

Disk Read

The amount of data in megabytes read by the VMHost from each Logical Unit Number (LUN) on the datastore per second.

Disk Write

The amount of data in megabytes written to the VMHost from each Logical Unit Number (LUN) on the datastore per second.

VMware Guest Metrics

Network

The **Network** chart displays information about the percentage of network bandwidth across the VMGuest that is used for downloads (IN) and uploads (OUT).

CPU

The **CPU** chart displays CPU utilization per VMGuest.

System Memory

The **System Memory** chart displays memory utilization per VMGuest.
Disk

The **Disk** chart displays the read and write latency for each disk on the VMGuest.

**Disk ms/Read**

The average amount of time it takes in milliseconds for the VM to read data from the virtual disk.

**Disk ms/Write**

The average amount of time it takes in milliseconds for the VM to write data to the virtual disk.

**Disk/MB sec Read**

The average amount of data used for reads by each VM in megabytes per second.

**Disk/MB sec Write**

The average amount of data used for writes by each VM in megabytes per second.
Storage

The Storage view provides details about the disk space and disk activity in your monitored environment. Use the Storage tab to view details about your monitored environment storage by disk, including capacity, free space, databases, disk activity, etc.

View data about your monitored environment's disk activity and disk space. Selecting a link from the table on the left populates details for your selection on the right in the Details and Trends tabs.

Overview Chart

The Overview chart displays storage capacity information for your monitored environment.
### Note
Select any blue link to display more information about your selection in the **Details** tab.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of your monitored storage unit.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of storage unit (Disk, Volume, Transaction Log, Data, etc).</td>
</tr>
<tr>
<td>% Used</td>
<td>The percentage of used space for the storage unit.</td>
</tr>
<tr>
<td>Size</td>
<td>The overall size of the Storage unit in MB or GB.</td>
</tr>
</tbody>
</table>
Details

The Details tab displays recorded storage metrics and any storage activity for your monitored environment. The details tab is divided into Capacity and Activity Metrics sections that display real-time data or a selected range of data.

### Capacity Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>The overall size of the selected Storage unit in MB or GB.</td>
</tr>
<tr>
<td>% Used</td>
<td>The percentage of used space for the selected storage unit.</td>
</tr>
<tr>
<td>File Group</td>
<td>The file group that the storage unit is associated with.</td>
</tr>
</tbody>
</table>

**Details**

The Details tab displays recorded storage metrics and any storage activity for your monitored environment. The details tab is divided into Capacity and Activity Metrics sections that display real-time data or a selected range of data.

### Capacity Metrics

- **Size**: 4.4 MB
- **% Used**: 88.73%
- **File Group**: PRIMARY

**Database**

**master**

**File Path**

C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL...
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>The database for the selected storage unit.</td>
</tr>
<tr>
<td>File Path</td>
<td>The file path for the selected storage unit.</td>
</tr>
<tr>
<td>Active VLFs</td>
<td>The number of active virtual log files for the selected storage unit.</td>
</tr>
<tr>
<td>Autogrowth</td>
<td>The percentage of autogrowth and amount of autogrowth memory in (KB), or (MB).</td>
</tr>
<tr>
<td>Autogrowth VLF Count</td>
<td>The number of virtual log files that have used autogrowth for the selected storage unit.</td>
</tr>
<tr>
<td>Autogrowth VLF Size</td>
<td>The size in (KB) or (MB) of the autogrowth VLF(s).</td>
</tr>
<tr>
<td>Last Backup Time</td>
<td>The time in UTC that the selected storage unit was last backed up.</td>
</tr>
<tr>
<td>Last Backup Type</td>
<td>The type of backup performed during the last backup period.</td>
</tr>
<tr>
<td>Log Reuse Wait</td>
<td>The status of the transaction log for the monitored connection.</td>
</tr>
</tbody>
</table>

**Additional Information:** For more information about the Transaction Log in SQL Server and possible statuses, see the [Transaction Log (SQL Server) MSDB article](#).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max File Size</td>
<td>The maximum file size for the selected storage unit.</td>
</tr>
<tr>
<td>Max VLF Size</td>
<td>The maximum virtual log file size for the selected storage unit.</td>
</tr>
<tr>
<td>Min VLF Size</td>
<td>The minimum virtual log file size for the selected storage unit.</td>
</tr>
<tr>
<td>Total VLFs</td>
<td>The total number of virtual log files for the selected storage unit.</td>
</tr>
<tr>
<td>Used Space</td>
<td>The total amount of consumed space for the selected storage unit.</td>
</tr>
</tbody>
</table>

**Activity Metrics**

**Note:** Activity Metrics update automatically in five minute increments by default.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads</td>
<td>The number of reads in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Reads/sec</td>
<td>The number of reads per second in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Read Bytes</td>
<td>The total number of read bytes in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Read Bytes/sec</td>
<td>The total number of read bytes per second in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Read Bytes %</td>
<td>The total percentage of read bytes in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Bytes/Read</td>
<td>The number of bytes per read in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Writes</td>
<td>The number of writes in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Writes/sec</td>
<td>The number of writes per second in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Write Bytes</td>
<td>The total number of write bytes in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Write Bytes/sec</td>
<td>The total number of write bytes per second in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Write Bytes %</td>
<td>The total percentage of write bytes in your selected storage unit for the specified time period.</td>
</tr>
<tr>
<td>Bytes/Write</td>
<td>The number of bytes per write in your selected storage unit for the specified time period.</td>
</tr>
</tbody>
</table>

**Note:** Activity Metrics update automatically in five minute increments by default.

**Trends**

The **Trends** tab displays Latency, IOPS, and MB/sec graphs for your selected storage unit.
Select a spike on a trends graph to display detailed read and write tooltips for your selection.

View storage trends historically for your selection. Select the calendar, and then select the desired Range, Days, Weeks, or Months that you want to display. Select Apply to update the trends graphs based on your selection.
**Note:** Select the button to switch to live data.

Select a graph metric to remove that data from the graph.

Select a Disk Volume to display a disk space forecast for the selection.
Warnings

SQL Sentry Portal tracks multiple warnings types across the storage entities in your monitored environment. Warnings display as red icons and are visible on the Overview chart. Select a storage entity with a warning icon to display more information about the warning in the Details tab.

⚠️ Important: Warnings in your monitored environment display hierarchically. For example, if you select your Controller, the Details tab will display all of the warnings in your environment. If you select an individual disk, the Details tab will display the warnings associated to the selected disk.

Select the ⬆️ on the column bar to sort by warnings.

Select a Storage Warning link to be redirected to the selection.
SQL Sentry Portal will notify you if your environment encounters any of the following warnings:

<table>
<thead>
<tr>
<th>Warning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TotalVlfsover100 (vlfs = virtual log files)</td>
<td>Total VLFs over 100. Too many VLFs can cause long startup and backup times.</td>
</tr>
<tr>
<td>TotalVlfsover300 (vlfs = virtual log files)</td>
<td>Total VLFs over 300. Too many VLFs can cause long startup and backup times.</td>
</tr>
<tr>
<td>AutogrowthUnderTenMegabytes</td>
<td>Autogrowth under ten megabytes.</td>
</tr>
<tr>
<td>AutogrowthOverOneGigabyte</td>
<td>Autogrowth over one gigabyte.</td>
</tr>
<tr>
<td>AutogrowthOverRemainingDiskSpace</td>
<td>Autogrowth over remaining disk space.</td>
</tr>
<tr>
<td>NonSimpleRecoveryModelAndNotBackedUp</td>
<td>This database is in a non-simple recovery model and is not backed up.</td>
</tr>
<tr>
<td>ForecastExhaustionWithinThirtyDays</td>
<td>The forecasted exhaustion of this volume is within 30 days.</td>
</tr>
<tr>
<td>LowFreeSpacePercentage (Free space &lt; 10%)</td>
<td>This volume has low free space.</td>
</tr>
</tbody>
</table>
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 SQL Sentry Portal feature for SQL Sentry, this default value may be changed and is controlled by the Monitoring Service Settings.

Available Charts

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

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

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

Show/hide axes has the following options:

- Avg Duration (ms)
- CPU Time (ms)
- Exec Count
- Reads Logical
- Writes Logical
- Reads Physical

On the options with the (chevron-right), additional choices similar to the above are available by selecting the (symbol. For example, By App - Duration (ms) and Queries - Reads (P) are available chart options.

Waits

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

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

Resources

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

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

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

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

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

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

By App

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

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

The information displayed here is from trace data.

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

By DB

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

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

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

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

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

The information displayed here is from trace data.

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

By Login

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

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

The information displayed here is from trace data.

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

Totals

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

**Note:** By default, the **Totals** grid displays the top 10 queries by logical reads (descending 🔄). For all grids, the arrow with the circle around it highlights by which column the data is sorted; ascending 🔄 or descending 🔄.

Use the search bar to further filter your Top SQL totals. Type your search parameters into the search bar and
select enter to filter the Totals grid.

Trace Events

Select the (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 (details), which flips all totals and statements tables to the detailed Trace Events and Trace Events Statements view on the page.

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

Trace Events Statements

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

Plan Diagram

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

Note:

- Sometimes the plan XML may be populated, but there’s not a statement that can be matched for the plan diagram. This can be caused by things like nested procedures or individual statements falling outside of the collection thresholds on their own. In this case, the plan diagram will display a message such as "The selected statement was not found in the plan XML. Download the full plan to view in SentryOne Plan Explorer" and provide a Download Plan XML button.
- If the Query History chart shows a disabled point (i.e. a gray triangle), then the plan diagram will display a message to indicate that there are no plans available (e.g. "There is no data to display").

Text Data

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

Parameters

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

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

### Query History

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

### Query Event

Each triangle represents a Query Event.

---

**Note:** The Copy button is only available when you are using HTTPS (requires an SSL certificate for your SQL Sentry Portal installation).
**Note:** The triangle colors represent execution plans. Triangles of the same color are using the same plan. If there is excessive plan drift (beyond 25 plans), then the 25 colors will start to be reused in the same order. Select a specific event to the Plan # represented.

**Query Stats Sample**

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

**Proc Stats Sample**

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

The SQL Sentry Portal 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. The Alerts Log displays a list of all of the conditions in your SQL Sentry environment that have evaluated to True.

From the Alerts Log, you can sort and filter current and active alerts across your environment monitored by SQL Sentry. View the most recent 15 alerts by default or dig deeper into previously recorded alerts. With Versions 2022.4 and later, you can assign a user to an active alert directly from the Alerts Log, add/edit notes on selected alerts, and close out alerts.

Alerts Log

The default view on the Alerts Log page displays the 15 most recent alerts logged across your environment (Start Time). You can sort the table by any of the available columns.

The following Alerting information is displayed on the Alerts Log grid:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 5 Servers</td>
<td>The top 5 servers by alert count in real-time, or across the selected timeframe.</td>
</tr>
<tr>
<td>Top 5 Conditions</td>
<td>The top 5 conditions by alert count in real-time, or across the selected timeframe.</td>
</tr>
<tr>
<td>Top 5 Tags</td>
<td>The top 5 tags by alert count in real-time, or across the selected timeframe.</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Close</td>
<td>Select the checkbox to close out the selected alert. Select the ✅ button to filter by the alert resolution status.</td>
</tr>
<tr>
<td>Target</td>
<td>The covered area of the watched target that triggered the alert.</td>
</tr>
<tr>
<td></td>
<td>‣ Note: If the scope of the target is at the server or instance level (i.e. Q-REGRESSION and not Q-REGRESSION: SQL Server Agent Jobs), then it will be a hyperlink to display Health, Performance, Top SQL, Blocking, Deadlocks, or TempDB for that target.</td>
</tr>
<tr>
<td>Type</td>
<td>The alert type, such as SQL Server, a deadlock (Deadlocks: Deadlock), etc.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the alert.</td>
</tr>
<tr>
<td>Condition Type</td>
<td>Conditions can fall into the General, Audit, Failsafe, and Advisory categories. For more information about the types of conditions categorized by SQL Sentry, see the General Conditions, Audit Conditions, Failsafe Conditions, and Advisory Conditions articles. Select the ✅ button to filter by specific condition types.</td>
</tr>
<tr>
<td>Action Type</td>
<td>The action that is set to occur once the monitored condition has been met. Actions are configured in the SQL Sentry client in the Conditions pane. For more information about configuring actions, and the available actions in SQL Sentry, see the Actions article. Select the ✅ button to filter by specific action types.</td>
</tr>
<tr>
<td>Assigned User</td>
<td>The user or group assigned to the alert. If a user or group has not been assigned, select the drop-down menu and select the user or group you want to assign. You can also search for a user or group with the drop-down search bar.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity may be high, medium, or low. For more information about alert severity within SQL Sentry, see the</td>
</tr>
<tr>
<td></td>
<td>Select the ✅ button to filter by specific severity levels.</td>
</tr>
<tr>
<td>Start Time</td>
<td>The time the alert started evaluating to true.</td>
</tr>
<tr>
<td>End Time</td>
<td>The time the alert stopped evaluating to true.</td>
</tr>
<tr>
<td>Duration</td>
<td>The amount of time that the alert was true.</td>
</tr>
<tr>
<td></td>
<td>‣ Note: The smallest value displayed is in seconds. If an alert was active for 500ms, it would display a duration of &lt; 1s.</td>
</tr>
<tr>
<td>Notes</td>
<td>Add, edit, or delete notes for a selected alert. See the Notes section below for more information.</td>
</tr>
</tbody>
</table>
Alerts Details

The details logged for any selected alert are displayed on Alert details screen below the Alerts Log grid.

The alert window displays a **Severity** at the top. Only alerts with a **Category** of Advisory Conditions have an associated severity level. The severity levels may be *High, Medium, or Low*.

The available details vary depending on the condition, and may contain information such as the step of a failed SQL Server Agent job and the error behind the failure.

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

**Filtering Alerts**

You can filter the alerts log by the Close, Condition Type, Action Type, or Severity categories; or you can filter by a combination of those categories. Filter the Alerts Log by completing the following steps:

1. Select the ✰ button for the desired category to open the filter drop-down menu.
2. Select the options you would like to filter by, and then select the X icon to close the list of options.
3. Select **Apply** to apply your filter.
4. Repeat the steps above for any additional categories.

**Note:** Select the Clear button at the top of the Alerts Log to clear any filter.

### Filtering alerts with the Top 5 charts

Select an option from any of the Top 5 charts to filter the Alerts Log by your selection. For example, in the image below we selected Q-Regression from the Top 5 Servers chart to filter by the Q-Regression server.

### Filtering with the Search bar

Use the search bar to further filter your Alerts Log. Type your search parameters into the search bar and select enter to filter the Alerts Log.

### Adding, Editing, and Deleting Notes

Starting with Versions 2022.4 and later, you can add, edit, and delete notes for any given alert using the Alerts Log grid.

Select **Add/View Notes** to open the Note Text column in the Alerts Log grid. The following options are
available:

**Buttons**

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Select Add to open the Add Note window. Enter the text for your note in the textbox and then select Create to save your note. Note: Select Include in Notifications to include your note in the alert notification.</td>
<td></td>
</tr>
<tr>
<td>Include in Notifications</td>
<td>Indicates whether this note is included in alert notifications. A checkmark means the note is included.</td>
<td></td>
</tr>
<tr>
<td>Edit</td>
<td>Select ☰ to open the Edit Note window. Enter the text for your note in the textbox and then select Save to save your note.</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>Select ☰ to open the Delete Note window. Select Delete to permanently delete the note.</td>
<td></td>
</tr>
</tbody>
</table>

⚠️ Note: A prompt displays when you have successfully added, edited, or deleted a note.

**Grid Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note Text</td>
<td>A sample of the note text.</td>
</tr>
<tr>
<td>Username</td>
<td>The username that logged the note.</td>
</tr>
<tr>
<td>Log Time</td>
<td>The date and time the note was added or edited.</td>
</tr>
</tbody>
</table>
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.

SQL Sentry Portal 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:

![Blocking by Application Chart]

- Depending on the screen width and number of items in the chart legend, you may need to scroll to see all items:
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.

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: [Transaction Locking and Row Versioning Guide](https://docs.microsoft.com) on Microsoft Docs.
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>SPID</td>
<td>The session process ID of the associated blocked/blocking process.</td>
</tr>
<tr>
<td>Start Time</td>
<td>Start time of the request.</td>
</tr>
<tr>
<td>Duration</td>
<td>The length of time that the block exists.</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><strong>Additional Information:</strong></td>
<td>See the <a href="https://docs.microsoft.com/en-us/sql/relational-databases/system-functions/sys-dm-os-wait-stats-transact-sql">sys.dm_os_wait_stats (Transact-SQL) MSDN article</a></td>
</tr>
<tr>
<td>Wait Resource</td>
<td>Name of the resource on which the request is currently waiting.</td>
</tr>
<tr>
<td>Host</td>
<td>Name of the server hosting the associated database.</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>
</tbody>
</table>
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.

Text Data

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

```
DELETE PerformanceAnalysisQuickTraceRow
FROM PerformanceAnalysisQuickTraceRow
INNER JOIN deleted ON PerformanceAnalysisQuickTraceRow.PerformanceAnalysisQuickTraceID = deleted.ID
```

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 or Hosts.
Blocking Filters

- **Applications**
  - (All)
  - .Net.SqlClient Data Provider
  - Core Microsoft.SqlClient Data Provider
  - Microsoft SQL Server
  - Microsoft SQL Server Management Studio - Query
  - SentryOne 18.5-CustomCondition
  - SQL Server Agent: Job 'Block A'

- **Databases**
  - (All)
  - AdventureWorks
  - BGinfoDatabase
  - CorporateDW
  - master
  - SentryOne
  - SQLSentry
  - sqlSentry20

- **Hosts**
  - (All)
  - LHP26-0419-B46E
  - LHP26-0419-D119
  - LHP26-0519-D15A
  - LMS32-0718-9950
  - Q-REGRESSION
  - SQUADZ-DEV
  - SSSWSS7
  - VM-SSSRV1
  - VM-SSSRV3

**Reset**  **Apply**

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 (✦) 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.
Applies to the following products and features: The on-premises SQL Sentry Portal feature for SQL Sentry. The Deadlocks tab is available for SQL Server, Azure SQL DB, and Amazon RDS targets. See the Getting Started with SQL Sentry Portal article for more details.

Deadlocks

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

Select a SQL Server, Azure SQL DB, or Amazon RDS target and then select Deadlocks to open the Deadlocks view.

**Note:** Deadlocks for Amazon RDS are not supported in SentryOne Monitor or the on-premises SQL Sentry Portal.

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 56 [0]* 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 *Key Lock*, this could also be at a different level such as an *Object Lock* or a *Page Lock*), and a **process** node (shown as *Process 246 [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 Deadlock Priority. Zero (0) or Normal is the default priority. In cases where each session has the same Deadlock Priority, SQL Server chooses the victim based on the least expensive session to roll back.</td>
</tr>
<tr>
<td>Wait Time</td>
<td>Time in (ms) milliseconds spent waiting on the resource.</td>
</tr>
</tbody>
</table>

**Additional Information:** For general information about the DEADLOCK_PRIORITY option, see the Set Deadlock_Priority MSDN article.
### 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/watcher.</td>
</tr>
<tr>
<td>Lock Mode</td>
<td>The requested lock mode (e.g. Shared (S), Update (U), Exclusive (X), etc.).</td>
</tr>
<tr>
<td>Host</td>
<td>The server or workstation name.</td>
</tr>
<tr>
<td>Application</td>
<td>The associated application name.</td>
</tr>
<tr>
<td>Login</td>
<td>The user login associated with the session.</td>
</tr>
<tr>
<td>Text Data</td>
<td>The associated text data (e.g. T-SQL statement).</td>
</tr>
</tbody>
</table>

#### 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>Text Data</td>
<td>The associated text data (e.g. T-SQL statement).</td>
</tr>
</tbody>
</table>

### Locks

- **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.

- **Trans Name**: Name of the associated transaction.
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>States whether the lock is held by the Owner or is a Waiter for the lock.</td>
<td>See the Transaction Locking and Row Versioning Guide and Lock Modes articles on Microsoft Docs.</td>
</tr>
<tr>
<td>Lock Mode</td>
<td>The requested lock mode (e.g. Shared (S), Update (U), Exclusive (X), etc.).</td>
<td></td>
</tr>
<tr>
<td>Lock Type</td>
<td>Points to the lock type, such as a page or object.</td>
<td>See the Lock Granularity and Hierarchies section of the Transaction Locking and Row Versioning Guide on Microsoft Docs.</td>
</tr>
<tr>
<td>Object</td>
<td>The object involved in the deadlock, such as a table, index, or view name.</td>
<td>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>
</tr>
<tr>
<td>Index</td>
<td>Index associated with the lock (if applicable).</td>
<td></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>
<td></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.
Note: The Copy button is only available when you are using HTTPS (requires an SSL certificate for your SQL Sentry Portal installation).
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 SQL Sentry 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 SQL Sentry, 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.
- For more information about this TempDB feature, see the SentryOne Clears the Fog Around TempDB blog post.

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.

**Note:** Long running open transactions can cause the version store to grow because newer versions can't be cleaned up until the oldest version is no longer needed.

### 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
**Note:** TempDB Object Stats must be enabled through Settings.

- Go to Settings Pane > SQL Server > SQL Server > Collect TempDB Object Stats. Set this option to True. This may be done at the global, site, or target level.

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 > SQL Server > Collect Session TempDB Usage**. Set this option to **True**. This may be done at the global, site, or target level.
    - On older versions, it was located under **Settings Pane > SQL Server > Miscellaneous > Collect Session TempDB Usage**.
  - This feature requires SQL Server 2012 or later on the watched target.

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 Summary** or **TempDB Activity** charts. Highlight a duration on either the charts and select **Filter** from the context menu to filter into a specific period of time.
Note that selecting TempDB from this context menu will zoom the TempDB Summary and TempDB Activity charts into the highlighted time period.

## 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>Max Granted Mem</td>
<td>The maximum amount of memory granted (in MB) for the associated session(s) during the defined range.</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>
</tbody>
</table>

**Note:** 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.

**Note:** 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.
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 SQL Sentry follows the standard Performance Analysis Dashboard Retention & Resolution data retention policies.

**TempDB Collection Settings**

- **TempDB Session Usage** collection must be enabled through the Collect TempDB Session Usage option in the Settings pane.
  - The TempDB Session Usage charts are not populated when this is turned off.
    - It is turned off by default.
- **TempDB Object Statistics** collection must be enabled through the Collect TempDB Object Stats option in the Settings pane.
  - The TempDB Objects charts are not populated when this is turned off.
    - It is turned off by default.
  - In some environments, you may experience issues with a query that starts with this text data: `WITH tempdbObjects AS...` being logged in SQL Sentry when collecting these statistics. See the
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.

![Example of TempDB metrics in Top SQL](image)

**Note:** You can highlight an area of activity and use the context menu to jump to Top SQL within SQL Sentry 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 Performance Metrics article.

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 SQL Sentry Reporting.

Report example:
Now available: The VMware performance dashboard is available with SQL Sentry Portal Version 2022.3

View the performance of your monitored VMware Host and VMware Guests on the performance tab. The Performance tab displays live or historical Network, CPU, System Memory, and Disk data for monitored VMware Host and Guests.

The left side of the VMware Performance tab displays your monitored VMware Host metric charts.

The right side of the VMware Performance tab displays your monitored VMware Guest metric charts.

Note: The context menu displays a selectable option for all monitored VMware Guests that exist on the portal installation.

Click and drag over a VMware Guest metric chart to highlight a spike or point of interest and display the Jump-To context menu. Select the VMware Guest that you want to view to open the Performance tab for your selection.
The VMware Host charts only display the Health and Performance Jump-To context menu options.

⚠️ Important: A VMware Guest that was previously monitored will display as a selectable option in the Jump-To context menu. Selecting an unwatched VMware Guest displays the performance tab for that selection with a This target is unwatched notification. You can still access historical data by selecting the calendar and selecting a date the VMware Guest target was monitored.
Portal Security
Last Modified on 21 November 2022

Authentication Methods

SQL Sentry Portal Service

The following authentication methods are available for connecting the SQL Sentry Portal service to your SQL Sentry database:

- **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 SQL Sentry database.
- **Azure Active Directory** is available in Versions 2023.1 and later.

SQL Sentry Portal User Access Requirements

⚠️ **Note:** The access requirements in this section are for users logging into SQL Sentry Portal. **SQL Server Authentication** is not supported for users logging in to use the portal through a browser.

⚠️ **Important:** **Feature Based Security** is enabled by default for **SQL Sentry Portal Versions 2022.2 and greater**. The Windows login(s) used to access your portal environment must be associated with a SQL Sentry User or Group to avoid connectivity issues.

**Additional Information:** For more information about Users and Groups in SQL Sentry, see the Add Users and Groups article.

Users logging into SQL Sentry Portal through a browser must have access to the Windows Server hosting SQL Sentry Portal. For a user to access SQL Sentry Portal, the following sets of requirements must be met:

- The **Windows user identity** is associated with a **SQL Sentry contact** (or is in an **Active Directory group** that is associated with a SQL Sentry contact) or it is associated with a **SQL Sentry contact group** (or is in an Active Directory group that is associated with a SQL Sentry contact group).

Rights Based Security

Starting with version 2021.1, SQL Sentry Portal supports **Rights Based Security**. See the Rights Based
Feature Based Security

⚠️ **Important:** Feature Based Security is enabled by default for SQL Sentry Portal Versions 2022.2 and greater. The Windows login(s) used to access your portal environment must be associated with a SQL Sentry User or Group to avoid connectivity issues.

Additional Information: For more information about Users and Groups in SQL Sentry, see the Add Users and Groups article.

Starting with version 2021.8, SQL Sentry Portal offers Feature Based Security which layers on top of Rights Based Security.

Feature Based Security is applied to contacts or contact groups in the SQL Sentry client, which are linked to an AD user or AD user group. Once the users are added through the client, they are available on the Users and Groups page in the SQL Sentry Portal. Feature Based Security uses roles to control access and is Default Deny. This means new users will only be able to access the Health view until they are assigned to roles that add more permissions.

🔗 Note: New Installations vs. Upgrades

- During a new installation, no roles are assigned to default users/groups.
- When upgrading SQL Sentry from a version that predates Feature Based Security (earlier than 2021.8) to one that supports Feature Based Security (versions 2021.8 or later), existing users and groups receive Alerts, Performance, and Query Tuner roles to maintain their existing access.

Permissions Page

Select the Permissions button from the top right in the navigation menu to open the Permissions page.

The Permissions page has four main sections worth noting:

1. SQL Sentry database environment drop-down
   - 🔄 Note: If you are using distributed SQL Sentry databases, roles must be configured per SQL
Sentry database (if you have multiple SQL Sentry databases such as SQLSentryUS and SQLSentryUK).

2. Users or Groups
   - With Versions 2022.4 and later, create new users and groups in SQL Sentry portal from the Users and Groups tabs.

3. User or Group Details
4. Feature Access grid

Users

The Users section lists all users that exist as contacts in SQL Sentry.

If a user belongs to one or more groups, they will be listed under groups in the User Details section.

User Details showing user Test in the Default Alerts group

**Note:** Inherited permissions from groups are not currently displayed in the Feature Access grid for the user in the roles.
The following information is displayed in User Details:

<table>
<thead>
<tr>
<th>Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>The user's first name.</td>
</tr>
<tr>
<td>Last Name</td>
<td>The user's last name.</td>
</tr>
<tr>
<td>Email</td>
<td>The user's registered email address.</td>
</tr>
<tr>
<td>Login</td>
<td>The user's login information for SQL Sentry Portal.</td>
</tr>
<tr>
<td>Groups</td>
<td>Any group(s) where the user is included.</td>
</tr>
<tr>
<td>Edit</td>
<td>Select <strong>Edit</strong> to open the Edit user window.</td>
</tr>
<tr>
<td>Delete</td>
<td>Select <strong>Delete</strong> to open the Delete user window.</td>
</tr>
</tbody>
</table>

Creating New Users

Starting with Version 2023.1, you can create new SQL Sentry Users on the Permissions page in SQL Sentry Portal. Create a new user by completing the following steps:

1. Select the **Users** tab and then select **Create New User** to open the New User window.
2. Enter a **First Name**, **Last Name**, and **Email** for the new user.
3. Select any groups you want the user to join from the drop-down menu.
4. Select **Create** to finalize your changes and create the new user.

![Create User Interface](image)
Editing Users

Starting with Version 2023.1, you edit existing SQL Sentry Users on the Permissions page in SQL Sentry Portal.

Edit a user by completing the following steps:

1. Select the **Users** tab and then select the user you want to edit.
2. Select the **Edit** button to open the Edit User window.
3. Make any desired changes to the user profile and then select **Save** to apply your changes.

Deleting Users

Starting with Version 2023.1, you can delete SQL Sentry Users on the Permissions page in SQL Sentry Portal.

Delete a user by completing the following steps:

1. Select the **Users** tab and then select the user you want to delete.
2. Select the **Delete** button to open the Delete User window.
3. Select **Delete** to delete the user.

Groups

The **Groups** section lists all groups that exist in SQL Sentry. Roles applied at the group level are inherited by the users in that group. The groups display the number of members next to the group name. Select a group to
Creating New Groups

Starting with Version 2023.1, you can create new SQL Sentry Groups on the Permissions page in SQL Sentry Portal. Create a new group by completing the following steps:

1. Select the Groups tab and then select Create New Group to open the Create Group window.
2. Enter a Group Name for the new Group.
3. Select any user(s) you want to add to the group from the drop-down menu.
4. Select Create to finalize your changes and create the new Group.

Editing Groups

Starting with Version 2023.1, you edit existing SQL Sentry Groups on the Permissions page in SQL Sentry Portal. Edit a Group by completing the following steps:

1. Select the Groups tab and then select the Group you want to edit.
2. Select the Edit button to open the Edit Group window.
3. Make any desired changes to the Group and then select **Save** to apply your changes.

Deleting Groups

Starting with Version 2023.1, you can delete SQL Sentry Groups on the Permissions page in SQL Sentry Portal. Delete a Group by completing the following steps:

1. Select the **Groups** tab and then select the Group you want to delete.
2. Select the **Delete** button to open the Delete Group window.
3. Select **Delete** to delete the Group.

Feature Access Grid

The **Feature Access** grid is where you assign roles to users and groups.

>Note: There are no edit or save buttons on the **Permissions** page. Any changes that are made in the **Feature Access** grid are automatically saved and are immediately in effect.

Roles set at the site level will flow through to the targets in that site. In this example, all sites and targets belong to the **Default Site**.
If a role is applied to a site within a site, then anything within that sub-site will inherit the role. In the following example, the Performance role is applied to site SquadD, (which is within the Default Site). The user (or group) will gain Performance role access to all targets within SquadD (SQUADD-PE.ENG.LOCAL, SQUADD-STABLE.ENG.LOCAL, and SQUADD-DEV.ENG.LOCAL) without gaining that additional access to other targets.

**Note:** Use the Show Hierarchy toggle to show which targets are impacted by a setting at a higher level in the hierarchy.
Default Roles

The following default roles are available:

Administrator

The Administrator role provides access to the Permissions page in SQL Sentry Portal. Administrators can only see targets for which they are an administrator for on the Permissions page. The targets are also filtered by Rights Based Security for the administrator and selected user/group. This means that an administrator cannot assign Feature Based Security permissions to a target that the selected user cannot already access.

⚠️ Note: The Permissions page can be accessed without the Administrator role if SQL Sentry Portal is accessed via the localhost domain. This gives the user admin-type rights at the Default Site for every tenant.

Alerts

The Alerts role provides access to viewing the Alerts view in SQL Sentry Portal.

⚠️ Note: Global alerts are only shown when the user has the Alerts role permission for every target/group.

Performance

The Performance role provides access to all metrics & chart data, including the Performance Analysis Dashboard, Custom Charts, Custom Dashboards, and charts on other views such as Top SQL, Storage, and TempDB. Users with the Performance role may create, edit, and delete the custom charts and dashboards that they can access.

⚠️ Note: If a dashboard contains a custom chart for a target that is restricted from the user, the chart will display a "This data is restricted" message instead of the chart.
Query Tuner

The **Query Tuner** role provides access to the query and session information on the **Top SQL, Blocking, Deadlocks, and TempDB** views in SQL Sentry Portal.

**Note:** The Top SQL and TempDB performance charts are hidden on these views unless the user has both **Performance** and **Query Tuner** roles applied.

Custom Roles

**Unsupported:** Custom roles may be created directly through the SQL Sentry database. There is no UI or support available at this time.

Once a custom role is inserted into the SQL Sentry database, the role appears in the **Feature Access** grid and may be assigned to users and groups the same way as default roles.

**Step 1. Create a custom role**

Run the following to create a role.

```sql
DECLARE @newRoleID UNIQUEIDENTIFIER;
SET @newRoleID = NEWID(); /*a GUID*/

INSERT INTO [Security].[FeatureRole] ([ID], [Name])
VALUES (@newRoleID, 'CustomRoleName');
```

**Step 2. Apply permissions to your custom role**

Use the following permissions and GUIDs to build your insert statement:

<table>
<thead>
<tr>
<th>Permission</th>
<th>GUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>418A7923-244B-4D44-A5A5-393EE3C2261C1</td>
</tr>
<tr>
<td>Alerts</td>
<td>A99A187E-925F-49C7-9037-F4EE9443874E</td>
</tr>
<tr>
<td>Blocking</td>
<td>86988B31-AE09-4145-989D-964C7196DF42</td>
</tr>
<tr>
<td>Permission</td>
<td>GUID</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>1CF9D61E-45EA-4851-B39C-0916FF1A5A24</td>
</tr>
<tr>
<td>Performance</td>
<td>01E6DA0E-EDF9-4A96-8336-63E225AF4CDA</td>
</tr>
<tr>
<td>TempDB</td>
<td>A8367D7C-FAB2-4979-A3BA-3887FDB6A52D</td>
</tr>
<tr>
<td>Top SQL</td>
<td>C5D894C2-B721-4ECB-A596-7C9002F31783</td>
</tr>
<tr>
<td>Storage</td>
<td>7AFF7534-912E-48C9-93CB-EB201648861D</td>
</tr>
</tbody>
</table>

Run the following insert statement to apply permissions to the role. This example adds **Blocking** permissions to the `CustomRoleName`.

```sql
INSERT INTO [Security].[FeatureRolePermission] ([RoleID], [FeatureID])
VALUES (@newRoleID, '1CF9D61E-45EA-4851-B39C-0916FF1A5A24');
```

On the Permissions page, a fifth role named `CustomRoleName` now exists and provides **Deadlocks** permissions to any user/group assigned this role.