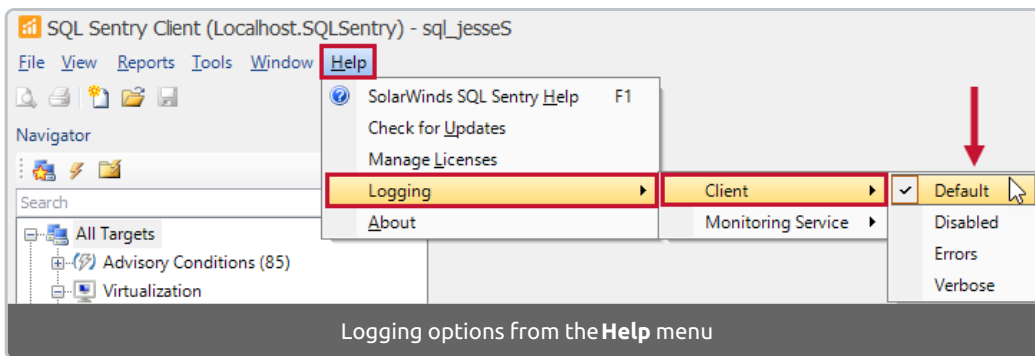


# SQL Sentry Troubleshooting

Last Modified on 05 August 2021

## Log files

By default, a certain level of logging takes place for the SQL Sentry client and monitoring service(s). You may increase or disable the logging level from the **Help** menu. Go to **Help > Logging** then select **> Client** or **Monitoring Service**.



Each component has the following options:

- **Default**
- **Disabled**
- **Errors**
  - Logs errors coming from the client or monitoring service
- **Verbose**
  - Logs most activity from the client or monitoring service

## Client

The location may vary between versions of SQL Sentry. Check the following locations for **console-log-file.txt**:

### SQL Sentry Unified Setup

C:\Users\\AppData\Local\SolarWinds SQL Sentry

or

C:\Users\\AppData\Local\SentryOne

or

C:\Program Files\SentryOne\

### Enhanced Platform Installer

C:\Users\\Documents\sentryone\client\\bin\

**Note:** The EPI version also has a client bootstrapper with its own logging:

```
C:\Users\\AppData\Local\sentryone\clientbootstrapper-log.txt
```

## Monitoring Service

The location may vary between versions of SQL Sentry . You may have multiple log files for the monitoring service. Check the following locations for **rolling-log.txt**:

### SQL Sentry Unified Setup

```
C:\Program Files\SolarWinds SQL Sentry\
```

or

```
C:\Program Files\SentryOne\
```

### Enhanced Platform Installer

```
C:\ProgramData\SentryOne\monitoringservice\bin\
```

**Note:** The EPI version also has a monitoring service bootstrapper with its own logging:

```
C:\ProgramData\SentryOne\servicebootstrapper\logs\
```

## Portal Service

**Note:** Consider using the built in [developer tools console](#) in your browser (open with **F12**) to reveal additional errors and information related to the SQL Sentry Portal.

The location may vary between versions of SQL Sentry . You may have multiple log files for the portal service. Check the following locations for **log.txt**:

### SQL Sentry Unified Setup

```
C:\Program Files\SolarWinds SQL Sentry\\MonitorPortal\logs
```

or

:\ProgramData\SolarWinds SQL Sentry\MonitorPortal\logs

or

:\ProgramData\SentryOne\MonitorPortal\logs

### Enhanced Platform Installer

:\ProgramData\SentryOne\monitor\monitorportal\logs

**Note:** The EPI version also has a portal service bootstrapper with its own logging:

C:\ProgramData\SentryOne\servicebootstrapper\logs\

## Installation and Upgrade Failure

Log files that are generated during the **SQL Sentry Setup** process can assist in troubleshooting a failure. Select **View Log** from the **SQL Sentry Setup** wizard if there's a failure.

You can also look for these files in the following locations:

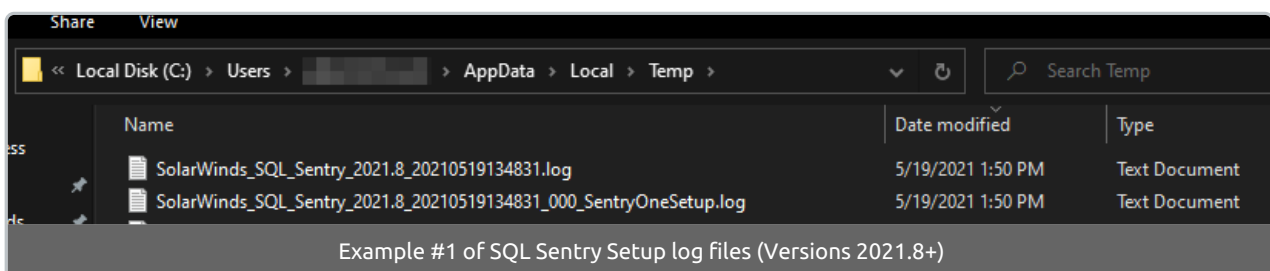
C:\Users\\AppData\Local\temp

and

C:\Users\\AppData\Local\temp\sqlsentry

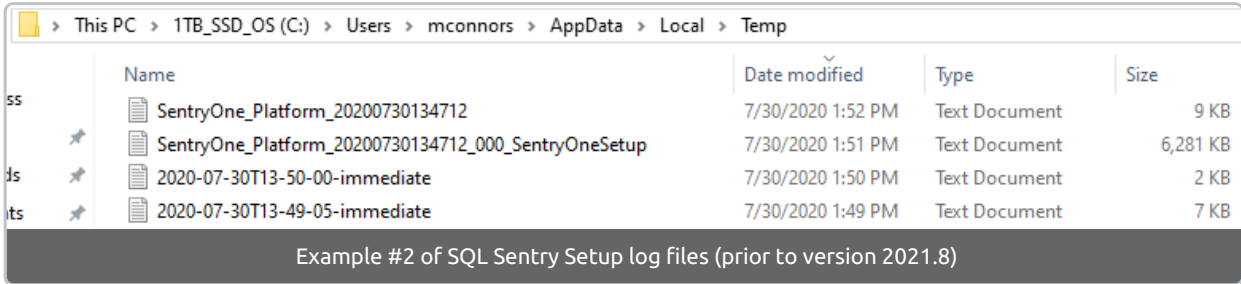
There are **immediate** logs related to the setup process that following a naming convention of *year-month-dayThour-minute-second-immediate* (shown below in Example #2 as **2020-07-30T13-50-00-immediate** and **2020-07-30T13-49-05-immediate**). Search these files for errors and exceptions.

There's also a more verbose log with a naming convention of *SolarWinds\_SQL\_Sentry\_Version\_DateTime\_LogNumber\_SentryOneSetup.log* (shown below as **SolarWinds\_SQL\_Sentry\_2021.8\_20210519134831\_000\_SentryOneSetup.log**)



In older versions of SQL Sentry (prior to version 2021.8), the naming convention was *SentryOne\_Platform\_DateTime\_SentryOneSetup* (shown below as

## SentryOne\_Platform\_20200730134712\_000\_SentryOneSetup).



The screenshot shows a Windows File Explorer window with the following path: This PC > 1TB\_SSD\_OS (C:) > Users > mconnors > AppData > Local > Temp. The window displays a table of files with columns for Name, Date modified, Type, and Size. There are four files listed, all of which are text documents. The second file, 'SentryOne\_Platform\_20200730134712\_000\_SentryOneSetup', is the largest at 6,281 KB. A dark grey bar at the bottom of the window contains the text: 'Example #2 of SQL Sentry Setup log files (prior to version 2021.8)'.

Name	Date modified	Type	Size
SentryOne_Platform_20200730134712	7/30/2020 1:52 PM	Text Document	9 KB
SentryOne_Platform_20200730134712_000_SentryOneSetup	7/30/2020 1:51 PM	Text Document	6,281 KB
2020-07-30T13-50-00-immediate	7/30/2020 1:50 PM	Text Document	2 KB
2020-07-30T13-49-05-immediate	7/30/2020 1:49 PM	Text Document	7 KB

Example #2 of SQL Sentry Setup log files (prior to version 2021.8)

**Note:** Often a search for **1603** in the *SolarWinds\_SQL\_Sentry\_Version\_DateTime\_LogNumber\_SentryOneSetup* (or *SentryOne\_Platform\_DateTime\_SentryOneSetup*) file can point you to the error stack that caused a failure.

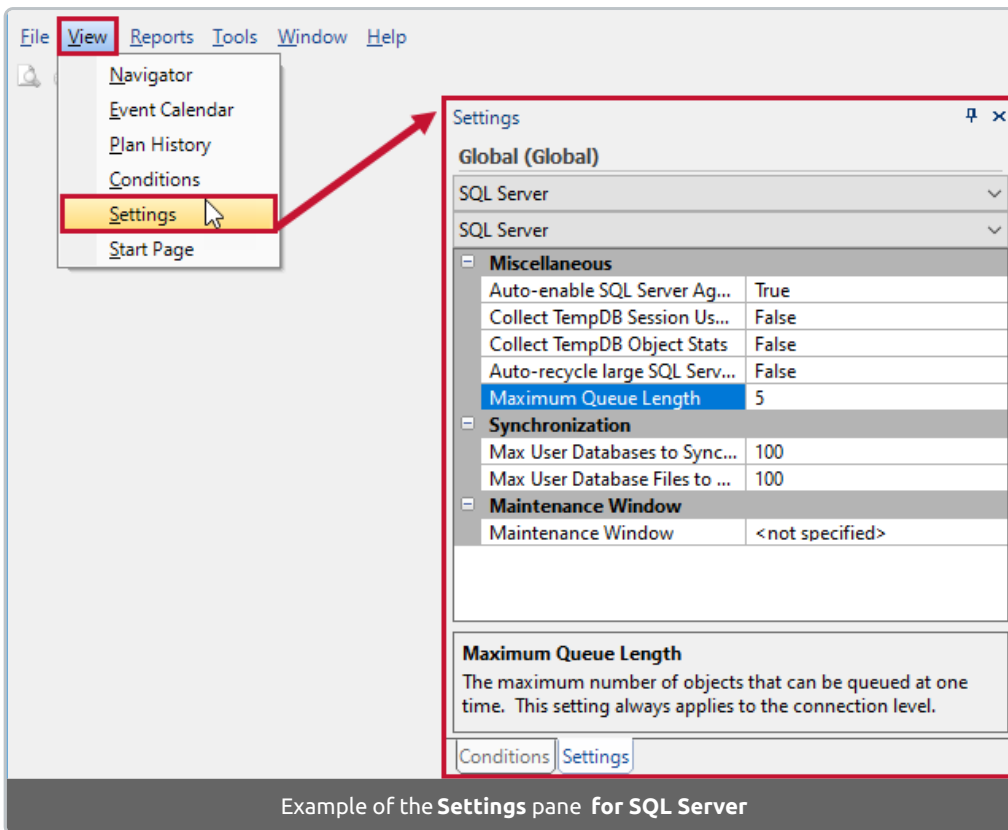
**Important:** If you are using **EPI**, don't forget to check the service bootstrapper logs (listed above under the components) for additional installation or upgrade errors and information.

## Miscellaneous

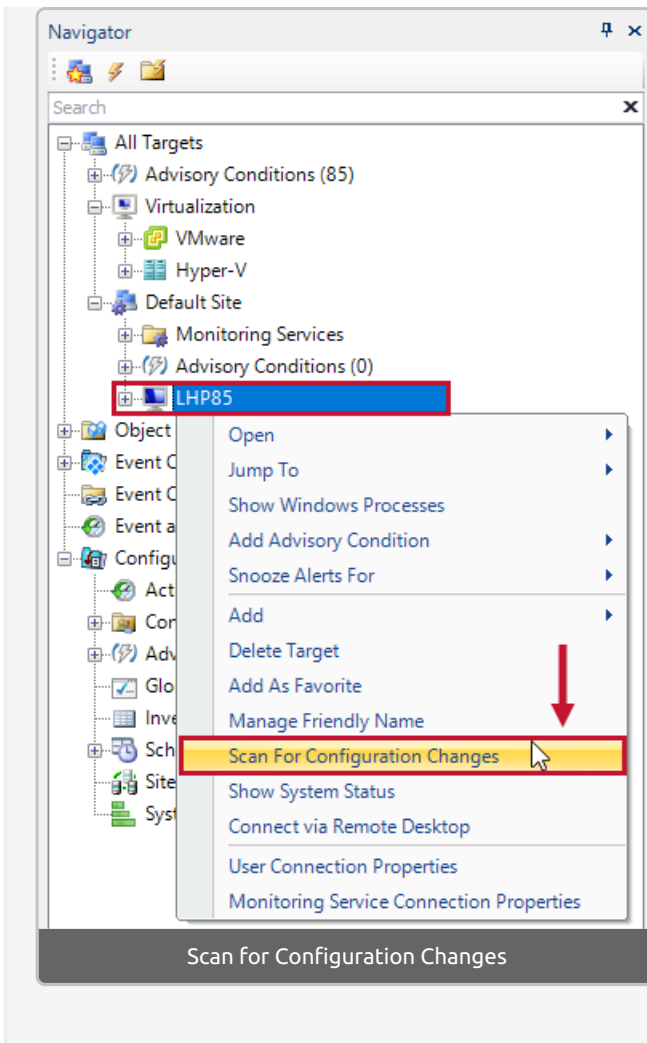
### Not all databases are being monitored

By default, SQL Sentry is set to sync 100 user databases. If you have more than 100, you can change what is synced via the **Settings** pane to include more if needed by adjusting the **Max User Databases to Synchronize** and **Max User Database Files to Synchronize (most active & largest first)** values.

These **Synchronization** settings exist for both **Amazon RDS for SQL Server** and **SQL Server**. Ensure you set them at the correct level by target type. See the [Settings Pane](#) article for more guidance on these options.



**Note:** Once you change the value, you should scan for configuration changes on the monitored target. Right click the target and select **Scan For Configuration Changes** from the context menu.



## AlwaysOn tab is blank

If you have **AlwaysOn** configured, but the **AlwaysOn** tab is blank or displays a message such as *No AlwaysOn resources are configured at this level* or *Charts Not Available*, you may have orphaned registry keys.

1. To diagnose this issue, run the following query against the primary node to confirm that the correct target has been added:

```
SELECT SERVERPROPERTY('servername')
```

2. Once that is confirmed, run the following query to check for orphaned registry keys:

```
USE master
SELECT * FROM sys.dm_hadr_instance_node_map
GO
```

3. If the query returns no results, then the Availability Group has orphaned records. Open **regedit** and navigate to **HKEY\_Local\_Machine > Cluster > HadrAgNameToldMap**.
4. Delete any orphaned keys.
5. Complete this process on all servers that are part of the Availability Group. **Note:** Some servers may not have orphaned keys in this scenario.

**Note:** Clusterless Availability Groups are not supported at this time and will also cause the AlwaysOn tab to be blank.

## tempdbObjects Query

If you are seeing aborted queries that have text data starting with something like the following, you may want to turn off performance data collection for [TempDB Object Stats](#):

```
WITH tempdbObjects AS ( SELECT ObjectType = CASE o.type_desc...
```

**Note:** There is no additional overhead on the monitored target because of this query being logged.

In Version 2021.12.20 or later, this collection can be turned off through [Settings](#). Go to **Settings Pane > SQL Server > SQL Server > Collect TempDB Object Stats**. Set this option to *False*. This may be done at the global, site, or target level.

Prior to Version 2021.12.20, there is no setting for this. To stop collecting data for **TempDB object statistics** on an earlier version, for all targets in your monitored environment, do the following:

**Note:** Use the name of your SQL Sentry database in the scripts. This may be the default or *SQLSentry* (previously *SentryOne*) or a custom database name that you created such as *SQLSentryUS*

1. Run this update against your [SQL Sentry database](#):

```
USE [SQLSentry]
UPDATE PerformanceAnalysisCounterCategory
SET PerformanceAnalysisSampleIntervalID = 0
WHERE ID = 115
```

2. Restart the [monitoring service\(s\)](#)

To turn it back on:

1. Run this update against your [SQL Sentry database](#):

```
USE [SQLSentry]
UPDATE PerformanceAnalysisCounterCategory
SET PerformanceAnalysisSampleIntervalID = 2
WHERE ID = 115
```

2. Restart the [monitoring service\(s\)](#)

## Invalid column name 'open\_transaction\_count'

When using the [TempDB Session Usage](#) feature, the following error message displays:

*SQL Sentry encountered an error with component: 'TempDB Session Usage Data Collector' in category: 'Data*

Collectors'.

The error message is: *TempDB Session Usage Detector: An error has occurred:*

*Message: Invalid column name 'open\_transaction\_count'.*

This occurs when the watched target does not meet the SQL Server version requirement. The **open\_transaction\_count** column was added to the `sys.dm_exec_sessions` DMV in SQL Server version 2012. Set the **Collect Session TempDB Usage** option to *False* on any targets that are running an older version of SQL Server.

[Additional Information](#): See the [Settings Pane](#) article for instructions on updating the settings.

## SQL Sentry client crashes when starting the application

If the SQL Sentry client crashes when you are attempting to open the application, there may be an issue with your preferences file.

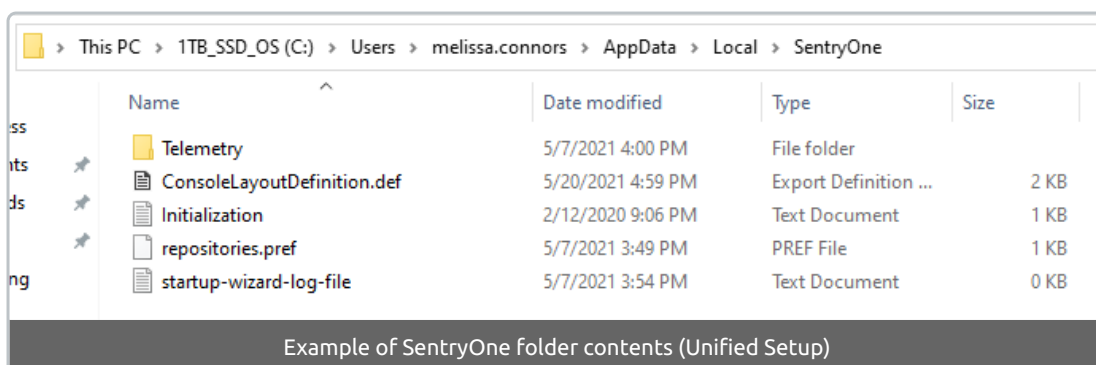
Delete the preferences file.

1. Go to to **C:\Users\\AppData\Local\SentryOne**
2. Delete the **repositories.pref** file
3. Restart the SQL Sentry client

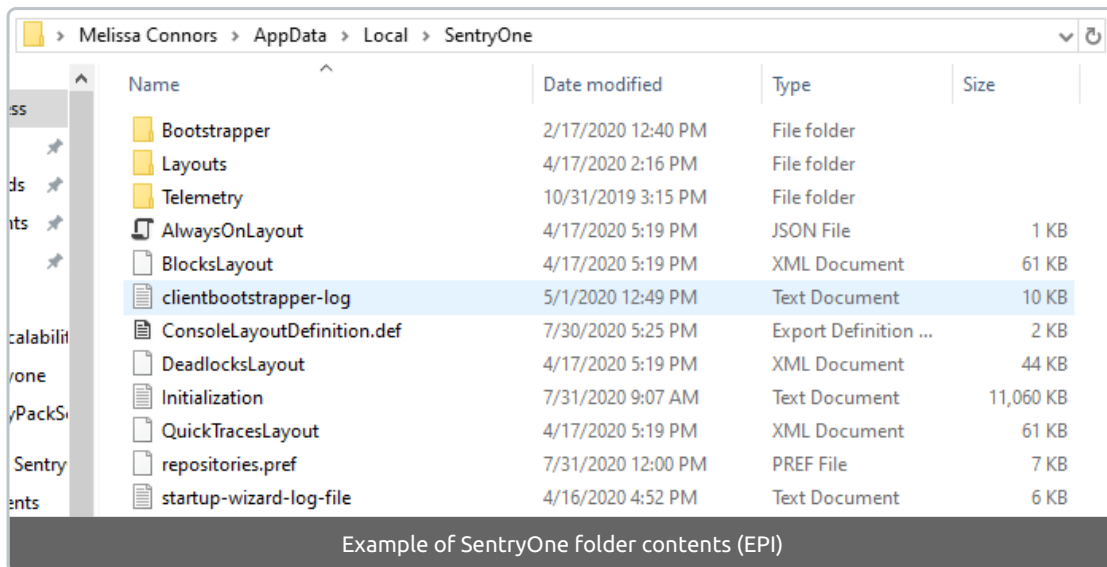
## SQL Sentry client doesn't complete loading

If the SQL Sentry client gets stuck on the splash screen when you are attempting to open the application, there may be an issue with your local user files.

1. Go to **C:\Users\\AppData\Local\SentryOne**.  
**Note:** For the EPI version, you may need to go to **C:\Users\\Documents\sentryone**.
2. Delete the contents of the SentryOne folder.







3. Restart the SQL Sentry client.