

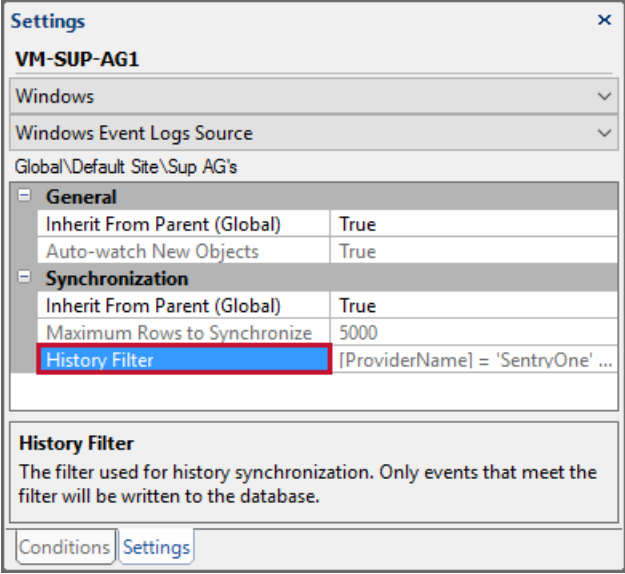
SQL Sentry Filtering Events

Last Modified on 18 May 2021

History Filter

SQL Sentry provides advanced source history filtering for your monitored targets and instances.

Source History Filter settings are available in the **Synchronization** group for select **Event Sources** ([View > Settings](#)).



The screenshot shows the 'Settings' window for 'VM-SUP-AG1'. The 'Synchronization' section is expanded, and the 'History Filter' row is highlighted with a red border. The filter expression is '[ProviderName] = 'SentryOne' ...'. Below the settings table, there is a description of the History Filter: 'The filter used for history synchronization. Only events that meet the filter will be written to the database.' At the bottom, there are tabs for 'Conditions' and 'Settings'.

General	
Inherit From Parent (Global)	True
Auto-watch New Objects	True

Synchronization	
Inherit From Parent (Global)	True
Maximum Rows to Synchronize	5000
History Filter	[ProviderName] = 'SentryOne' ...

History Filter
The filter used for history synchronization. Only events that meet the filter will be written to the database.

With the **History Filter**, you define rules that control the types of events the monitoring service writes into **Event History** concerning your monitored servers. If an event doesn't meet the criteria you've defined in the **History Filter**, information about that event isn't written to your SQL Sentry database.

Using the **History Filter**, you may choose to ignore certain types of events, as you can filter multiple columns including **Duration**, **Message Text**, and **Run Status**. The filter was designed with flexibility in mind. For example, you may define a filter as simple as **Event Duration** being greater than five seconds, or you may choose to create complex filter conditions, using conditional groups and multiple logical operators.

Depending on your environment and the type of information you collect, using a **History Filter** with the various Event Sources can also decrease the size of your SQL Sentry database. On busy servers, applying a **History Filter** tailored for those specific environments can also significantly reduce the amount of noise, leaving you with a smaller set of actionable information. One powerful application of the source **History Filter** is that it can be used with the Windows **Event Logs** source. The [Windows Event Logs source](#) is available when you are monitoring a Windows target.

Note: If you're looking for more control over alerts, [Condition settings](#) give you granular control over what you're being alerted about within your environment.

Configuring History Filters

The **Source History Filter** settings are available in the **Synchronization** group for select **Event Sources**

(**View > Settings**). For more information about using the **Filter Editor**, including context menu information, see the [Filter Editor](#) topic.

Synchronization of Performance Analysis Sources

It's important to understand that the **History Filter** synchronization settings for Deadlocks affect only what's displayed on the **Event Calendar** and doesn't impact what's captured or displayed within **SQL Sentry**.

The synchronization of the **Deadlocks** source uses a different process than those sources that are strictly **Event Calendar** only. This data is first written to various **Performance Analysis** specific tables within the [SQL Sentry database](#), and is then used to populate the **Performance Analysis Deadlocks** tab.

This data is then correlated and written into **Event Calendar** tables according to your defined synchronization settings, including any filters you have specified with the **History Filter**. This **Event History** data is used to populate the **Event Calendar**, showing you **Deadlocks** alongside your regularly scheduled agent jobs and various other event objects.

If you're looking for more control over alerts related to **Performance Analysis** sources, [Condition settings](#) provide granular control over what you're being alerted about within your environment. **Condition** settings allow you to build complex alert criteria for [conditions](#), including those related to **Top SQL**, **Blocking SQL**, and **Deadlocks**.

Custom Event View Filtering

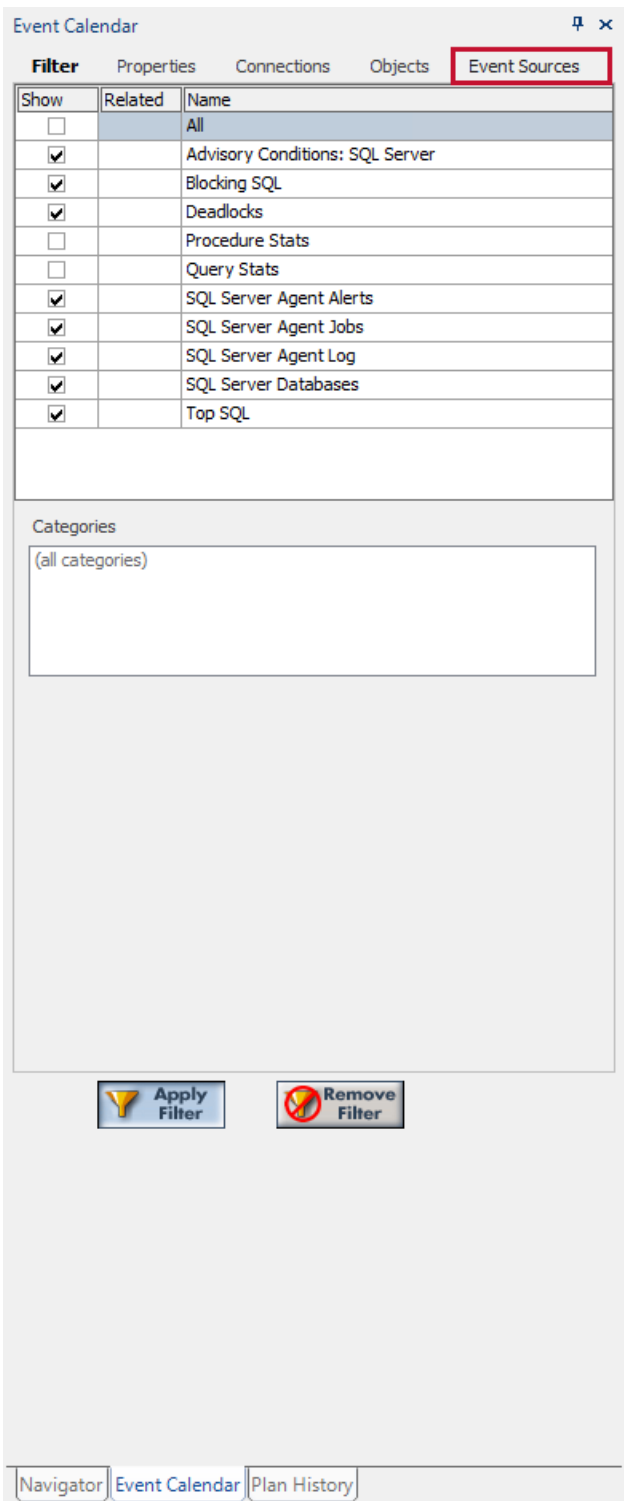
When setting up a **Custom Event View**, it's important to understand the two levels of object filtering that take place. The first level of filtering, source level filtering, determines which objects are included in the view. Any performance counters that are applied to this view are applied to every object that's included after this level of filtering. The second level of filtering, general filtering, only determines under which circumstances the above objects are visible on the calendar. They still have performance counters applied to them whether they are visible or not.

Source Level Filtering

Any filter text specified from the [Add Instances](#) window when setting up or making changes to a view is at the source level. This restricts the number of objects associated with the view, and therefore the number of objects that are monitorable.

After the view is created, additional source level filtering can be applied through the **Event Sources** tab such as **Jobs**, **Alerts**, **Tasks**, etc. By checking or deselecting **Show > Source >** checkboxes, you determine whether objects of this type will be included in the view.

On the **Event Sources** tab are **Event Objects** and **Categories** boxes that allow you to further filter objects from the view. These boxes are typically populated when working with a view with only one [instance](#).



General Filtering

Any filters applied from the **Filter** tab of the **Event Views** pane determine whether certain event objects are visible on the **Calendar**. Even if they're made invisible by this filter, they still have any performance counters applied to them that are applied to the view. This is an important distinction for performance purposes. If you don't want a performance counter activated for an object in a view, you must ensure that it's filtered at the source level, and not simply made invisible by the **Filter** tab.

Event Calendar 🔍 ✕

Filter | Properties | Connections | **Objects** | Event Sources

Show: History Future Running

Show: Conflicts Only Watched Only

Run Status

And (all) Failed Succeeded

Review Status

And (all) Open Closed

And

Text Filters

Event Object Name

Step Text

Owner/Creator

And

Duration Filters

Average/Actual Duration

Greater Than 0 Seconds

Action Show Instances

Event View Filter Process Flowchart

