

# Building SQL Sentry Advisory Conditions

Last Modified on 29 March 2021

## Introduction


Advisory conditions in SQL Sentry enable you to check for many types of scenarios and apply an [action to respond or alert you](#) to the existence of the condition. The SQL Sentry installation includes dozens of default conditions in the [advisory conditions pack](#) that you can immediately apply actions to.

On GitHub, you'll find [even more advisory conditions](#) that have been created by members of the SQL Sentry community (e.g. employees, partners, and customers). The conditions from GitHub must be downloaded and [imported into your SQL Sentry environment](#) and may require some additional configuration for your exact needs.

After exploring all the existing advisory conditions available, you may find that you want to create your own condition (or modify an existing one). This article walks through some examples to help you get started with building advisory conditions. See the [conditions section of the SQL Sentry documentation](#) for a complete guide to all options available.

## Tutorial: Creating a Condition

This tutorial walks through creating a condition by showing the steps to create an existing condition ( [High CPU](#)). This condition uses a Windows performance counter to check for CPU that is greater than 90%.

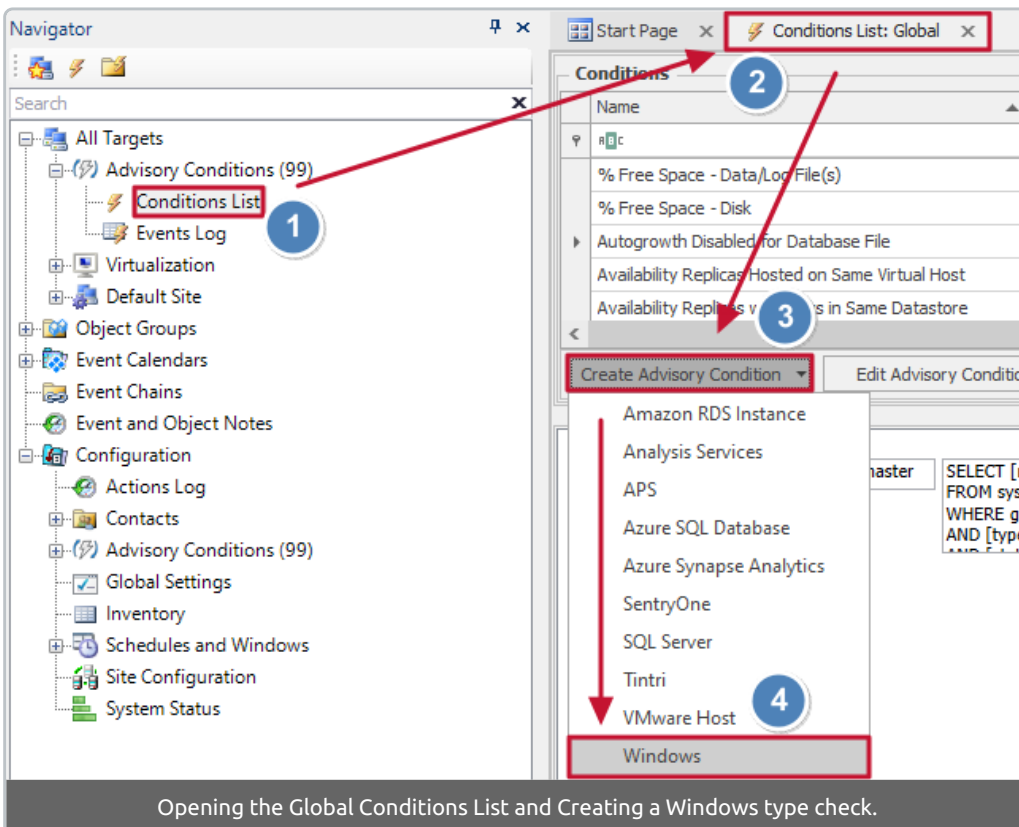
 **Additional Information:** See the [comprehensive Advisory Conditions](#) article for details about the field definitions, features, and options not used or explained in this tutorial.

### Step 1: Select the Condition Type

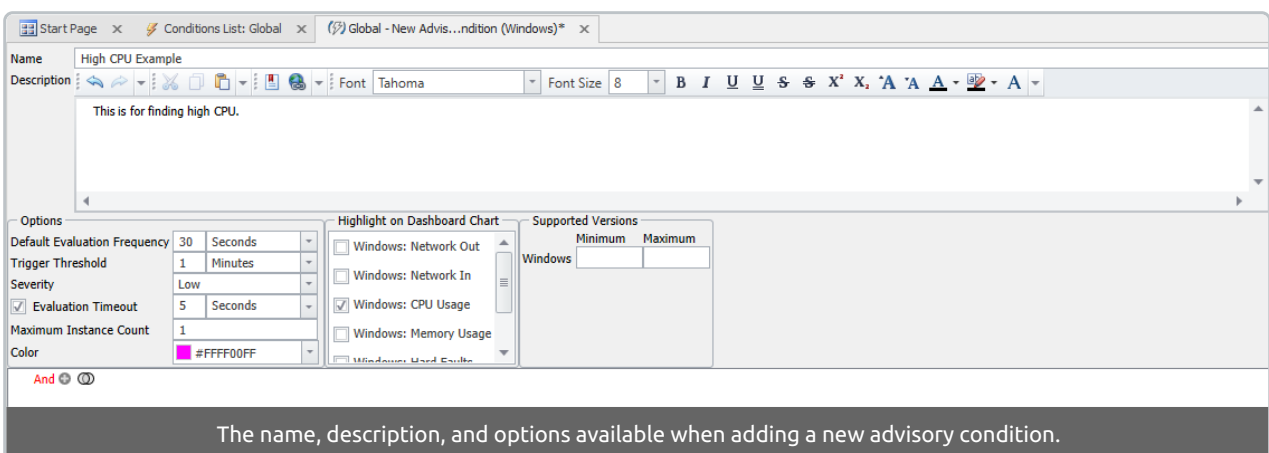
There are several types of advisory conditions that you can create (e.g. Amazon RDS Instance, Analysis Services, APS, Azure SQL Database, Azure Synapse Analytics, SentryOne, SQL Server, Tintri, VMware Host, and

Windows).

1. Select the **Conditions List** from the **Navigator** pane. **Note:** The conditions list is available at different levels in the hierarchy of targets. See the [hierarchy](#) section in the [Introducing Conditions, Actions, and Settings](#) article.
2. Open the **Conditions List** (this example used the global one).
3. Select the **Create Advisory Condition** button to get a list of condition type options.
4. Select the condition type from the drop-down (this example uses **Windows**).



## Step 2: Set the Condition Options



1. Enter a **Name** for your condition. ( *This appears in emails, logs, calendar views, etc. when the condition evaluates to true.* )

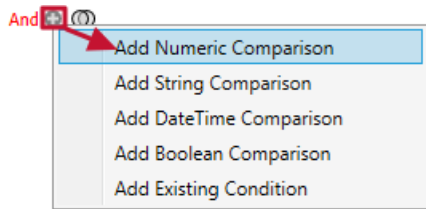
- This example uses "High CPU Example".
2. Enter a meaningful **Description** for your condition. (*Rich text word processing options are available on the toolbar above.*)
    - This example uses "This is for finding high CPU".
    - **Note:** You may want to be more descriptive, use exact values, and provide helpful links in a condition that you create for how someone should respond to the condition.
  3. Set the **Default Evaluation Frequency**. (*How often this condition check executes. Consider the performance implications of the condition check and how soon someone would need to be notified should this situation occur. It's not uncommon to have conditions that run only hourly, daily, or weekly.*)
    - This example uses the default value of 30 Seconds.
  4. Set a **Trigger Threshold**. (*How long does this condition need to take place to evaluate to true?*)
    - This example uses 1 Minute.
    - See this [blog post on the trigger threshold](#) to learn more.
  5. Select the **Severity** associated with this condition. (*How concerning is this condition if it occurs?*)
    - This example uses Low.
  6. Set the **Evaluation Timeout**. (*How long can this condition run this check before failing due to timeout?*)
    - This example uses the default value of 5 Seconds.
  7. Set the **Maximum Instance Count** if it needs to be more than the default value of one. (*If this condition can return multiple results, how many should be returned?*)
    - This example uses the default value of 1.
    - See this [blog post on maximum instance count](#) to learn more as it requires a specific combination of settings to use.
  8. Select a **Color** to highlight this condition. (*Which color should be used to highlight this condition in the application?*)
    - This example uses #FFFF00FF.
  9. Choose where to **Highlight on Dashboard Chart**. (*Where should this condition be highlighted on the Performance Analysis Dashboard, using the selected Color?*)
    - This example uses Windows: CPU Usage.
    - See the [Send to Alerting Channels](#) section of the [Actions](#) article for an example of how this appears when a condition evaluates to true.
  10. Constrain the condition to **Supported Versions** if needed. (*Control which environments this condition checks against. Is there a minimum or maximum version of Windows where this counter is available, or is this situation only a problem in certain versions? This uses the numerical release number of the product, i.e. Windows Server 2019 is NT 10.0, so 10.0 would be used in these fields. Leave one or both of the fields blank if this isn't applicable.*)

## Step 3: Define the Condition

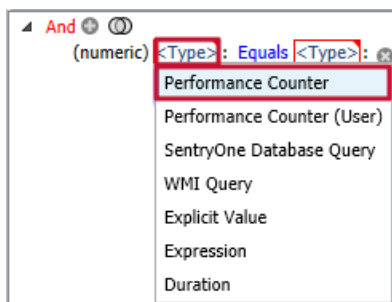
Below the area where the options are set, there's the word **And** followed by the **Add New Condition** icon (plus sign in a circle) and the **Add New Condition Group** icon (a Venn diagram symbol). This is where you

define the scenario for your condition. The *High CPU* condition is a one-step condition, so we'll ignore the **And** and **Add New Condition Group** features for now (see more about those in the **Conditions with multiple steps** section below).

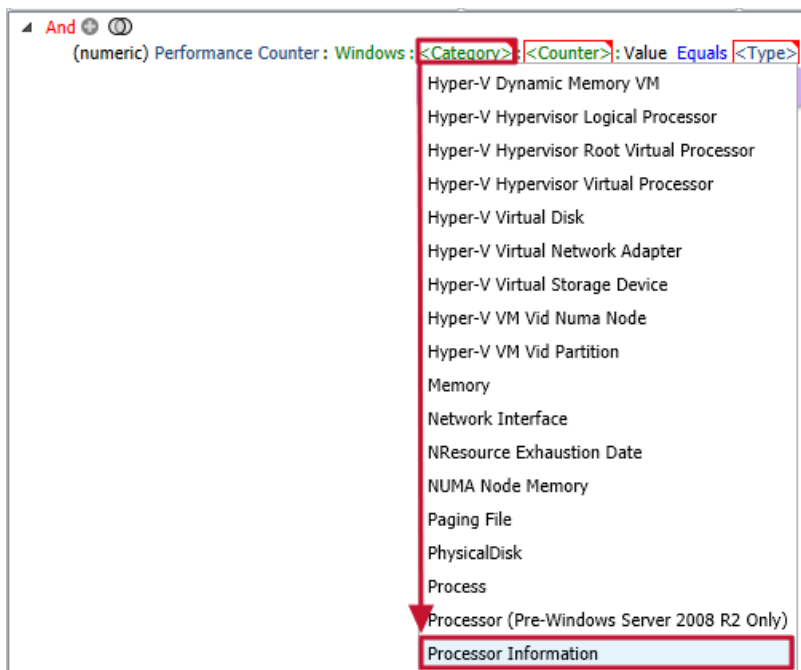
1. Select the **Add New Condition** icon.
2. Select **Add Numeric Comparison** from the **Add New Condition** dropdown menu.



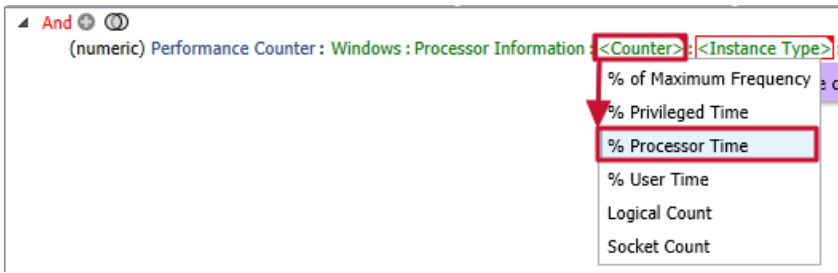
3. Select the **Type** box, then choose **Performance Counter** from the dropdown menu.



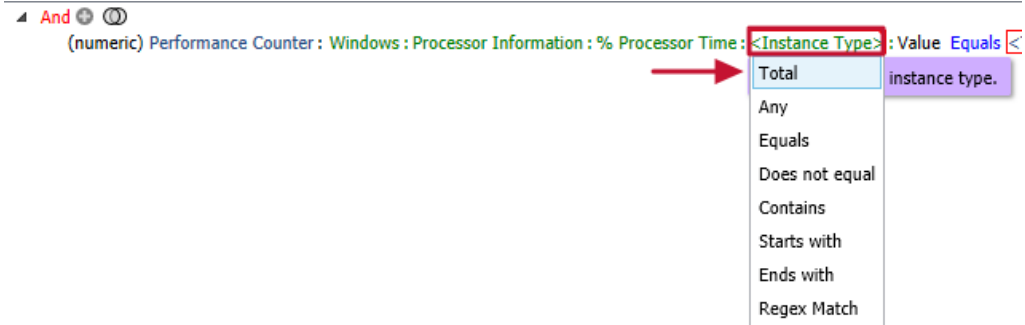
4. Select the **Category** box, then choose **Processor Information** from the dropdown menu. **Note:** The **Performance Counter** is set to **Windows** performance counters because the advisory condition type was set to **Windows** in the first step.



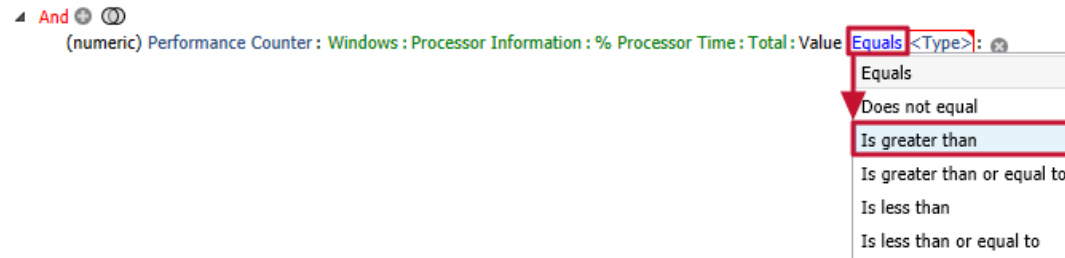
5. Select the **Counter** box, then choose **% Processor Time** from the dropdown list.



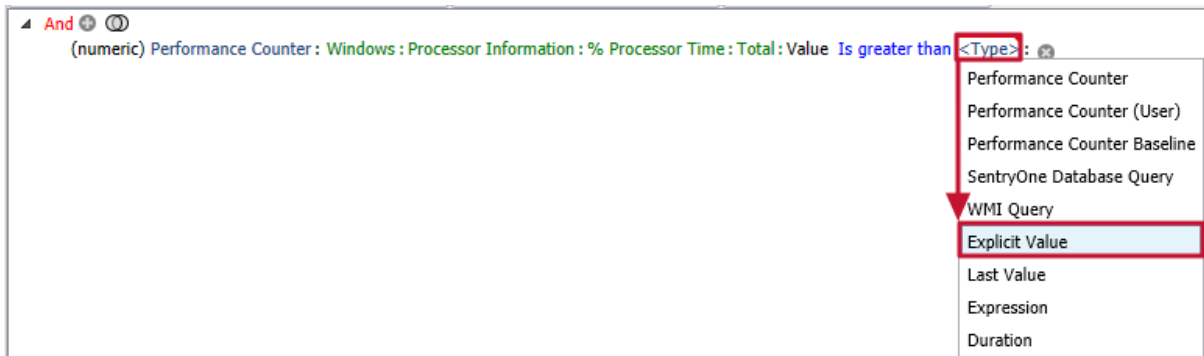
6. Select the **Instance Type** box, then choose **Total** from the dropdown list.



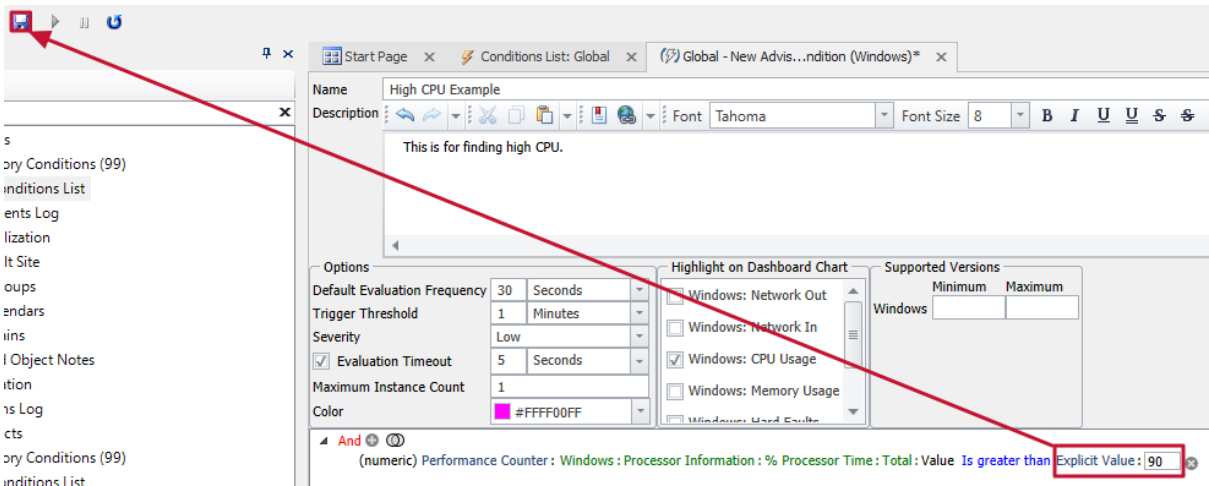
7. Select the word **Equals**, then choose **Is greater than** from the dropdown list.



8. Select the **Type** box, then choose **Explicit Value** from the dropdown list.



9. Enter **90** in the **Explicit Value** box, then select the **save** button.



10. Your condition has been saved.

**Success:** You have created a new advisory condition!

**Note:** The **Add Actions** popup appears after saving the condition.

Add Actions

Would you like to assign actions for this advisory condition now?

Yes No

Select **Yes** if you would like to apply an action to this condition. This opens the **Actions Selector** window. See the [Configuring Actions](#) article for instructions on adding and configuring actions.

## Additional Examples and Resources

### Examples

As you can see from the various menus and options available in the tutorial, there are several different types of advisory conditions that you can create, and a staggering number of combinations that can be created. You can compare performance counter values to the result of a SQL Server query, a current value to the last value, match string values using Regex, perform calculations on the values, and so on. Use the existing conditions installed with SQL Sentry and available on GitHub as a source of inspiration for building your own conditions.

### Clone conditions

1. Right click on a condition in the **Conditions List**.
2. Select **Clone** from the context menu. A small window will pop up and ask you to name your copy of the condition.

3. Enter the **Name** of your condition, then select **OK**.

**Success:** You have cloned an existing condition and can now use it as a template for a new condition. Make edits as needed and save it.

## Conditions with multiple steps

The tutorial used a single step condition to check for *% Processor Time > 90*. If you wanted to check for *% Processor Time > 90* and some other condition, such as *Available Memory (MB) < 500*, you can do that in the same condition. See the [Using Step Logic in SQL Sentry Advisory Conditions](#) blog post for a deep dive into those possibilities.

## Nested conditions

You can embed existing conditions into conditions to nest the steps and checks. For example, the **High Ad Hoc Query Plans** condition (shown below) contains multiple checks, including the **Low Page Life Expectancy** condition.

High Ad Hoc Query Plans condition

**Note:** If you take a look at the *High Compiles + High CPU* condition, you'll see three different advisory conditions nested in the check in steps and condition groups. If you have a complex condition in mind, you can build it.

## Blog Posts

- [Creating Advisory Conditions in SQL Sentry Part 1](#)
- [Creating Advisory Conditions in SQL Sentry Part 2](#)
- [Create Advisory Conditions for SQL Server on Amazon RDS](#)
- [Setting Up a MaxDOP Advisory Condition](#)

- [Using SentryOne to Alert on SQL Server Database Object Changes](#)
- [Monitoring Replication with SQL Sentry](#)