Performance Analysis Processes

Last Modified on 20 August 2021

Applies to: Windows, SQL Server, and SSAS targets when the target has Windows (meaning targets on Linux, AWS RDS, and Azure Managed Instance, for example, will not have this tab). It's only available for SQL Server when using the advanced security mode (i.e. **Full Access**).

Overview

Default Processes Collected

The **Processes** tab is available as part of **Performance Analysis** for Windows.

By default, **Performance Analysis** for Windows collects information about well-known or categorized processes. There are several of these predefined, well-known process groups, including groups for:

- SQL Server
- IIS
- SSAS
- SSIS
- SSRS

Collecting Additional Processes

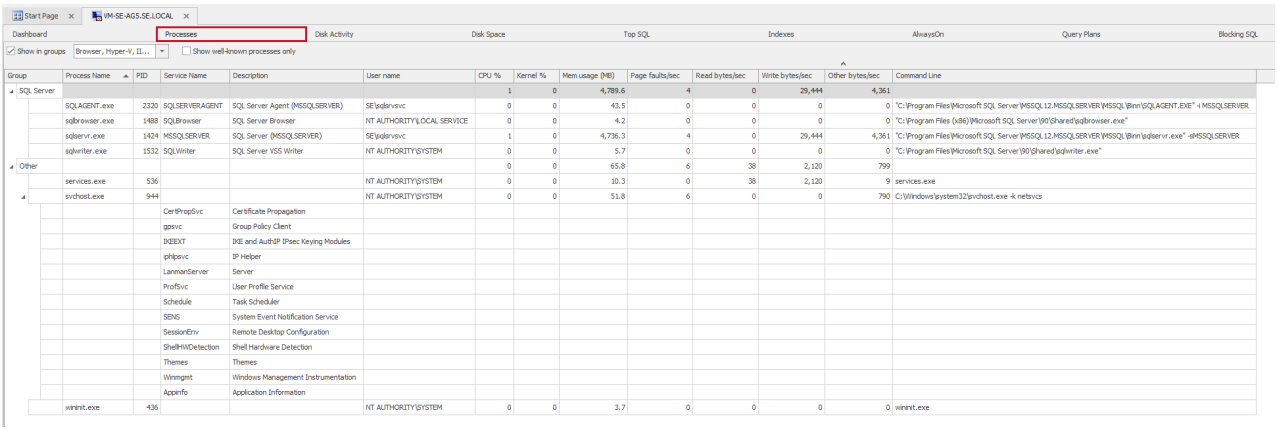
To collect information about processes that aren't in these default groups, you have a couple of different options to do so:

- Add your own well-known groups, and then specify the processes that belong to them.
- Configure an **Uncategorized Process Filter**, that allows for the collection of uncategorized processes.

You may choose to use both of these options. For example, you may wish to define several new well-known process groups that are relevant to the monitored computer's workload, so that information about those processes is always collected. You may also wish to define an **Uncategorized Process Filter**, so that information about processes that are consuming a large amount of resources on the computer are collected.

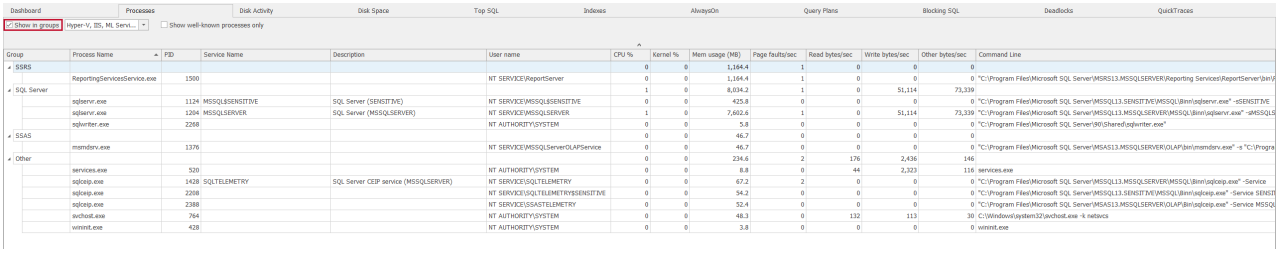
Processes Tab Display

The **Processes** tab contains a grid view of all the processes for which you are collecting information. Processes are intelligently grouped by program and function, including groups for SSRS, SSIS, and SQL Server (well-known process groups). Metrics are displayed for each process, giving you insight into the resource usage for both the individual processes and their associated groups.

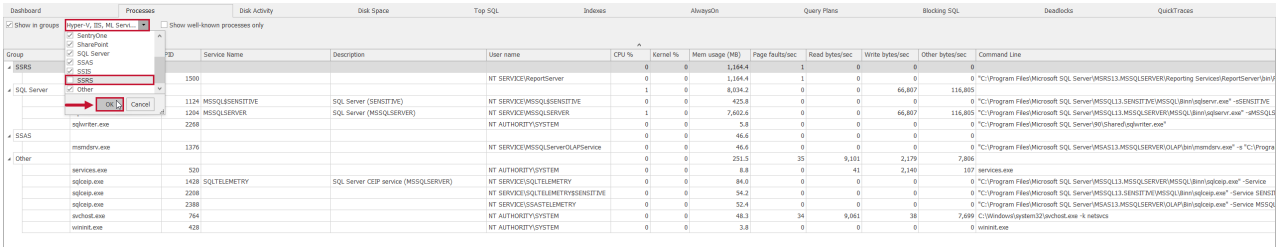


Display Options

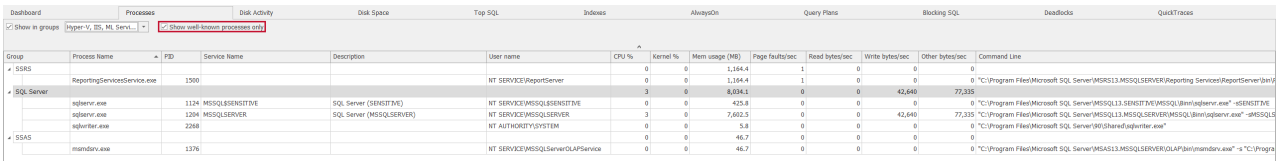
- **Show in groups** — Groups well-known processes by their assigned group or category.



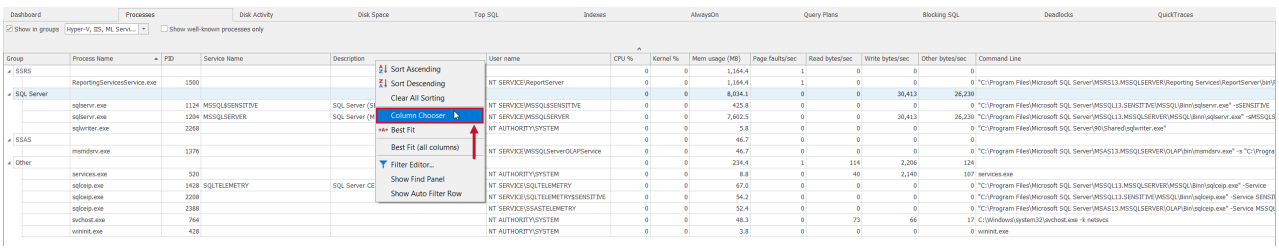
- **Showing/Hiding Groups** — Hides certain groups from view using the group drop-down list box. Uncheck any group that you want to hide.



- **Show well-known processes only** — Hides any processes that aren't part of a well-known group.



- **Additional Columns** — Access **Additional Columns** by right-clicking on any column header and selecting the **Column Chooser** option.



- **Sorting Columns** — Sort the grid view by specific columns by selecting any column header. Additional sorting options are available from the context menu of any column header.

Adding New Groups of Well-Known Processes

By default, there are several well-known processes that are logically grouped by program/function, including predefined groups for SQL Server, IIS, SSAS, SSIS, and SSRS. Add additional groups of your own by using the following SQL Sentry database tables:

- **Performance Analysis Device Process Group**
- **Performance Analysis Device Process Group Mapping**

Performance Analysis Device Process Group

Use this table to define groups and your well-known processes.

Column Name	Description
ID	Identity column
ProcessGroupName	This is the group name that displays in the **Processes** tab.
ProcessGroupNameFull	The full name isn't displayed in the **Processes** tab.
ProcessGroupDescription	The description of the process group.

Performance Analysis Device Process Group Mapping

Use this table to map specific processes to the groups you defined in the **Performance Analysis Device Process Group** table.

Column Name	Description
ID	Identity column
PerformanceAnalysis DeviceProcessGroupID	The ID of the group you would like this process to map to. Should map to the ID column of a group found in the **Performance Analysis Device Process Group** table.
ProcessName	Enter the exact name under which the process is executed, including the executable extension. For example: *services.exe*
ServiceName	Enter the name of the service exactly as it's displayed on the **General properties** tab of the service as shown from the services applet. To find the actual service name, open the services applet and right-click the

Column Name	Description
	service, and then choose **Properties**. Use the exact value found in the **Service name** field.
ShowOnPADashboard	Specifies if the group displays on the **Performance Analysis Dashboard**.
CommandLineMatchRegex	This column is useful when differentiating processes that use the same executable, but perform different functions based upon the passed in parameters. The regex should use a valid .NET compatible regular expression as defined here. **Note:** The match operation is case-insensitive.

Group Mapping Example

As an example, if the target you are monitoring is a Microsoft Exchange server, you may want to create a new group of well-known processes that includes those related services. In this example we create a new group of well-known processes for **Microsoft Exchange,** and then we add the following Microsoft Exchange related services to the group:

- **Microsoft Exchange Information Store**
- **Microsoft Exchange Mailbox Assistants**
- **Microsoft Exchange Transport**
- **Create the Group**

```
INSERT INTO dbo.PerformanceAnalysisDeviceProcessGroup (ProcessGroupName,ProcessGroupNameFull)
VALUES ('Microsoft Exchange','Microsoft Exchange Server 2010')
```

- Add the processes and map them to the group

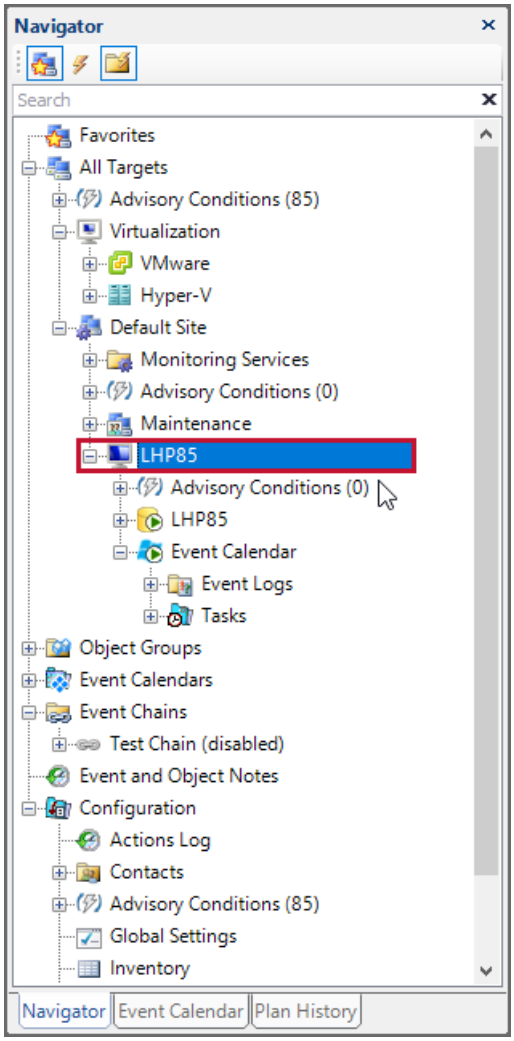
```
INSERT INTO dbo.PerformanceAnalysisDeviceProcessGroupMapping
(PerformanceAnalysisDeviceProcessGroupID,ProcessName,ServiceName,ShowOnPADashboard)
VALUES
(9,'Store.exe','MSExchangeIS',1), (9,'MSExchangeMailboxAssistants.exe','MSExchangeMailboxAssistants',1), (9,'MSExchangeTransport.exe','MSExchangeTransport',1)
```

Important: Be sure that the **PerformanceAnalysisDeviceProcessGroupID** corresponds to the **ID column** of your desired group in the **Performance Analysis Device Process Group** table.

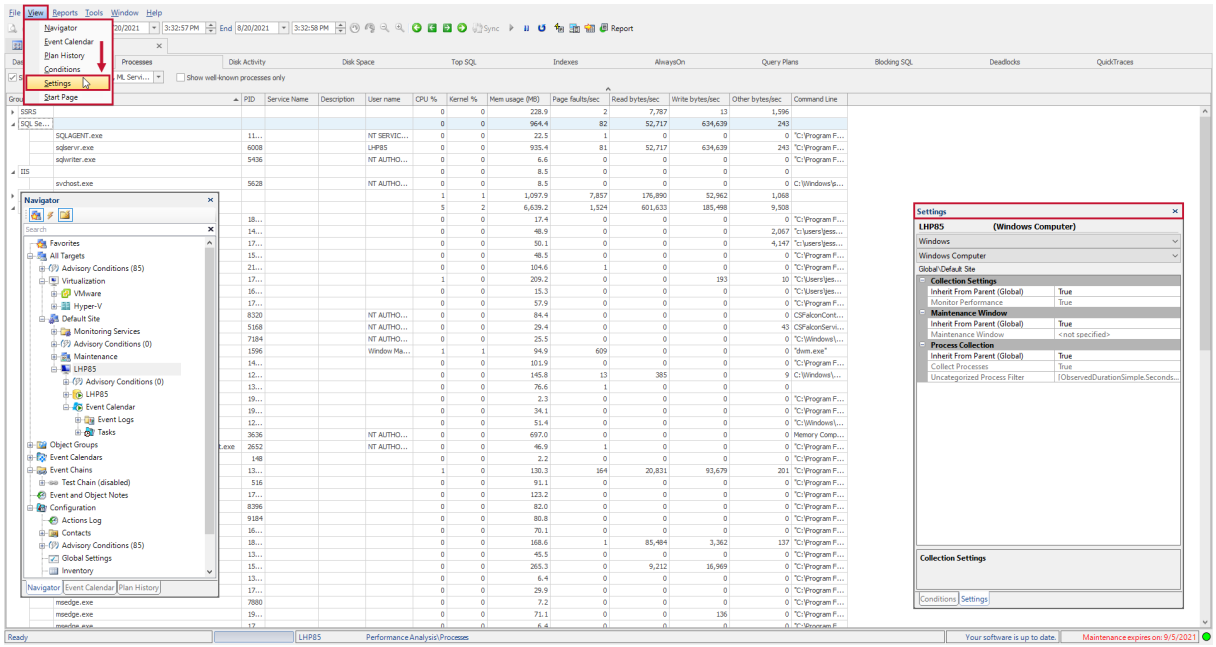
Process Collection Settings

Process collection is controlled with the **Windows Instance Settings** and can be accessed by completing the following steps:

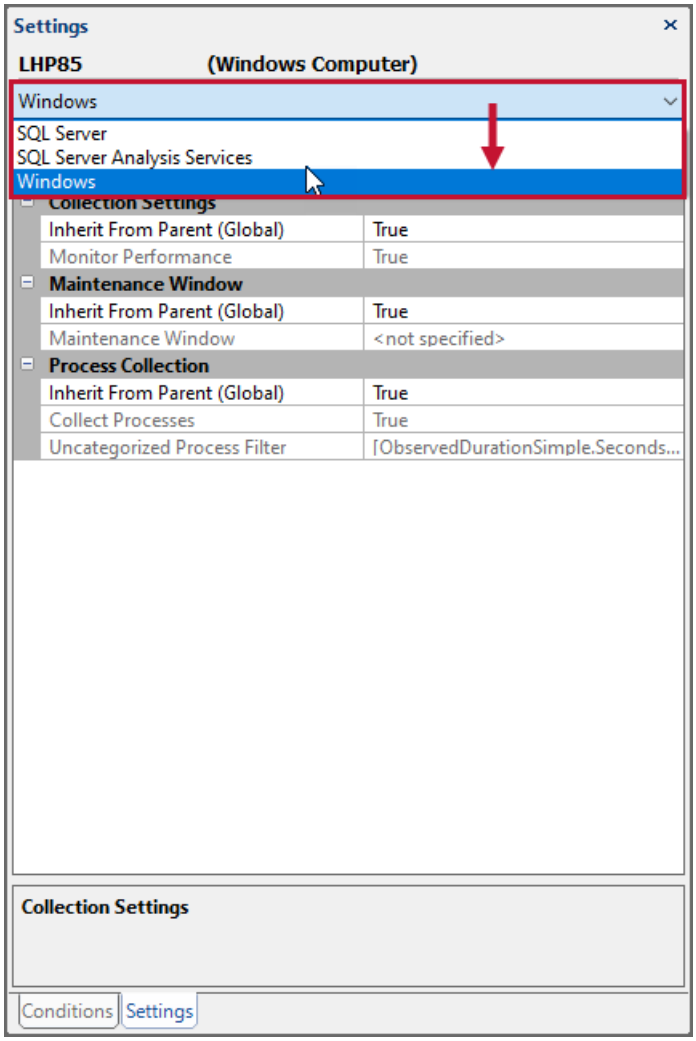
1. Select the desired Windows instance in the **Navigator** pane.



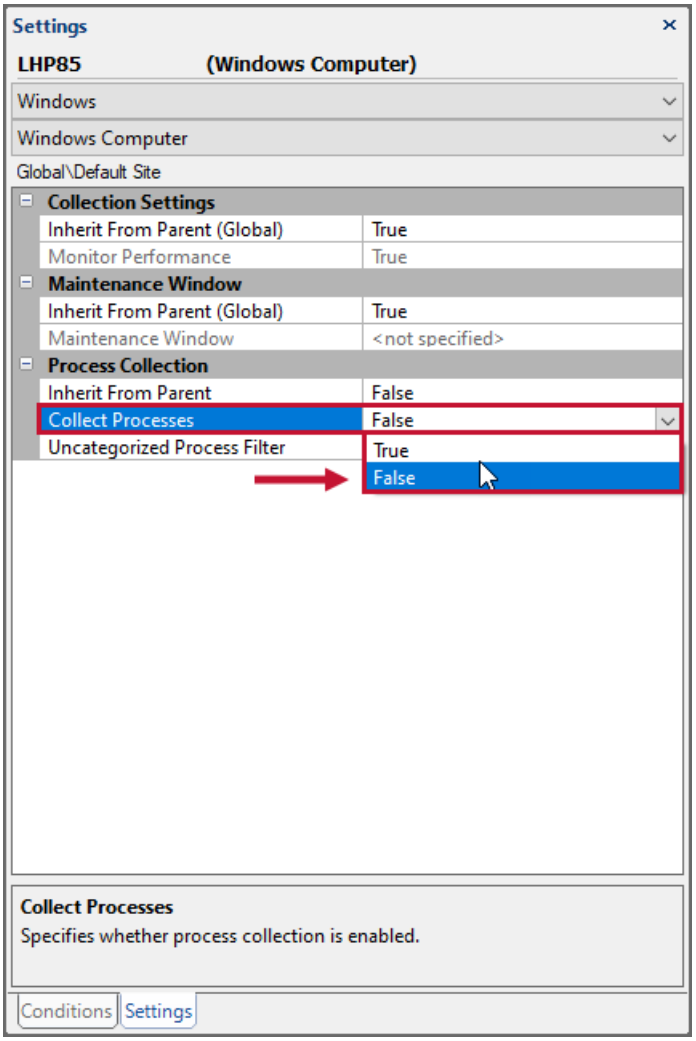
2. Open the **Settings** pane (**View menu** > **Settings**).



3. In the **Settings** pane choose the Windows instance from the drop-down menu.



You should now see the **Process Collection** settings in the **Settings** pane. By default, when a Windows instance is monitored, process information is collected about all well-known processes. Disable the collection of process information by changing the **Collect Processes** setting to **False**.



You may enable the collection of additional processes (non well-known processes) by configuring an **Uncategorized Process Filter**.

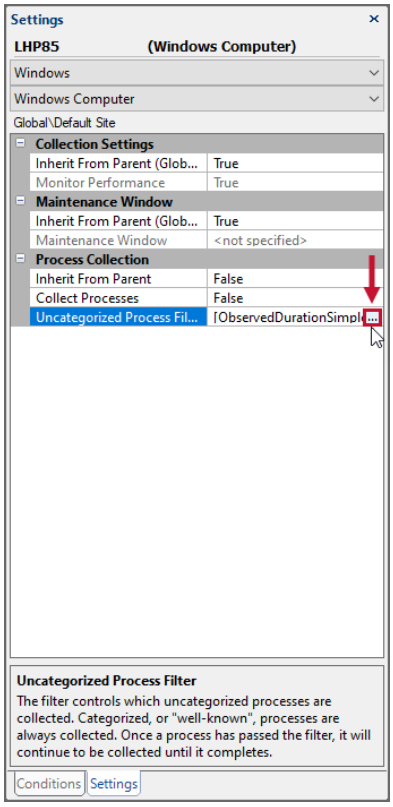
Configuring an Uncategorized Process Filter

By default, Performance Analysis for Windows collects information about well-known or categorized processes. Configure **SQL Sentry** to collect information about Uncategorized processes by specifying an **Uncategorized Process Filter**. This filter may be built around various metrics, including percentage CPU time or Read and Write bytes per second.

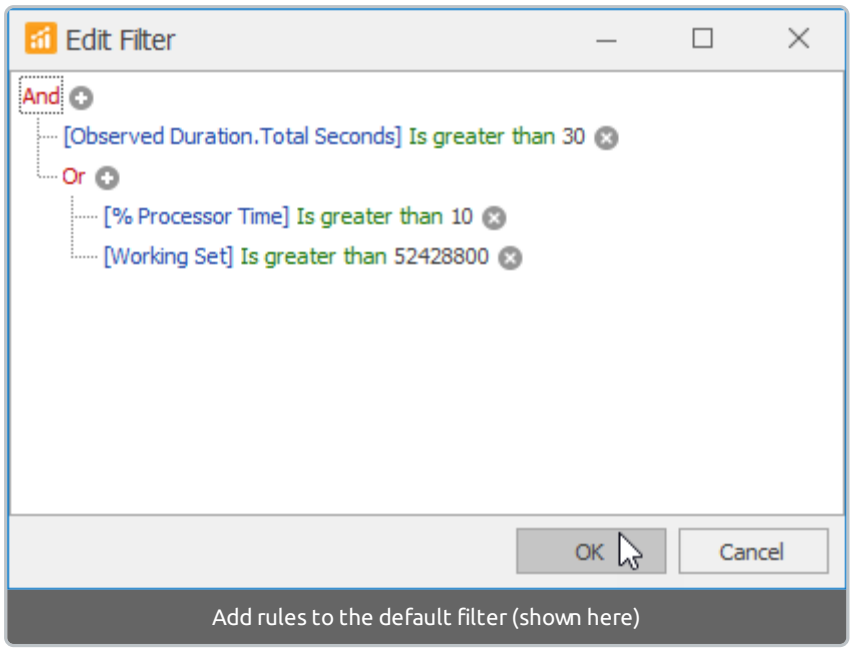
> ⚠ **Important:** This filter exists by default for process collection. Changing the filter to allow for more collection may add significant overhead to your network and the SQL Sentry database. Consider using baselines to measure how your changes impact your environment.

Follow the directions in the previous section to access the **Process Collection** settings for your Windows instance.

1. Select the **Uncategorized Process Filter** row and use the ellipsis **(...)** to open the **Filter Editor**.



2. Add rules to the filter as desired.



Note: The **Filter Editor** was designed with flexibility in mind, and allows you to specify any number of criteria around events. You can define a complex rule with multiple groups and logical operators.

For more information about using the **Filter Editor**, see the Filter Editor topic.

SolarWinds Database Mapper Environment Map

The **SolarWinds Database Mapper Environment Map** shows data collected from the SQL Sentry database to map connections between applications, users, clients and targets (sourced from **Top SQL** and **Windows processes**). This information complements the lineage analysis feature by showing the dynamic usage of targets in the lineage diagram and shedding light on the processes that are using the database.

To learn more about using **Windows processes** data with Database Mapper, see the Environment Map article.