Splunk® Enterprise Getting Data In 8.0.0
Monitor Windows performance

Generated: 11/16/2019 2:02 pm
Monitor Windows performance

Splunk Enterprise supports the monitoring of all Windows performance counters in real time and includes support for both local and remote collection of performance data.

The Splunk Enterprise performance monitoring utility gives you the abilities of Performance Monitor in a web interface. Splunk Enterprise uses the Performance Data Helper (PDH) API for performance counter queries on local machines.

The types of performance objects, counters and instances that are available to Splunk Enterprise depend on the performance libraries installed on the system. Both Microsoft and third-party vendors provide libraries that contain performance counters. For information on performance monitoring, see "Performance Counters" on MSDN.

Both full instances of Splunk Enterprise and universal forwarders support local collection of performance metrics. Remote performance monitoring is available through WMI (Windows Management Instrumentation) and requires that Splunk Enterprise runs as a user with appropriate Active Directory credentials. If you have Splunk Cloud and want to monitor Windows performance metrics, you must use the Splunk universal Forwarder to collect the data and forward it to your Splunk Cloud deployment.

The performance monitor input runs as a process called splunk-perfmon.exe. It runs once for every input defined, at the interval specified in the input. You can configure performance monitoring with Splunk Web, or either inputs.conf (for local performance data) or wmi.conf (for performance data from a remote machine).

Why monitor performance metrics?

Performance monitoring is an important part of the Windows administrator's toolkit. Windows generates a lot of data about a system's health. Proper analysis of that data can make the difference between a healthy, well functioning system, and one that suffers downtime.

What do you need to monitor performance counters?

The following table lists the permissions you need to monitor performance counters in Windows. You might need additional permissions based on the performance objects or counters that you want to monitor.
For additional information on performance metrics monitoring requirements, see Security and remote access considerations.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Required permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor local performance metrics</td>
<td>* Splunk Enterprise must run on Windows.</td>
</tr>
<tr>
<td></td>
<td>* Splunk Enterprise must run as the Local System user.</td>
</tr>
<tr>
<td>Monitor remote performance metrics on another</td>
<td>* Splunk Enterprise must run on Windows.</td>
</tr>
<tr>
<td>computer over WMI</td>
<td>* Splunk Enterprise must run as a domain or remote user with at least read access to WMI on the target computer.</td>
</tr>
<tr>
<td></td>
<td>* Splunk Enterprise must run as a domain or remote user with appropriate access to the Performance Data Helper libraries on the target computer.</td>
</tr>
</tbody>
</table>

**Security and remote access considerations**

Splunk Enterprise gets data from remote machines with either a forwarder or WMI. Splunk recommends using a universal forwarder to send performance data from remote machines to an indexer.

If you install forwarders on your remote machines to collect performance data, then you can install the forwarder as the Local System user on those machines. The Local System user has access to all data on the local machine, but not to remote computers.

If you want Splunk Enterprise to use WMI to get performance data from remote machines, then you must configure both Splunk Enterprise and your Windows network. You cannot install Splunk Enterprise as the Local System user, and the user that you choose determines what Performance Monitor objects that Splunk Enterprise can see.

After you install Splunk Enterprise with a valid user, you must add that user to the following groups before you enable local performance monitor inputs:

- Performance Monitor Users (domain group)
- Performance Log Users (domain group)

To learn more about WMI security, see Security and remote access considerations in "Monitor WMI Data". To learn about how to use a universal forwarder, see About the universal forwarder.
Enable local Windows performance monitoring

You can configure local performance monitoring either in Splunk Web or with configuration files.

Splunk Web is the preferred way to add performance monitoring data inputs. You can make typos with configuration files, and it is important to specify performance monitor objects exactly as the Performance Monitor API defines them. See "Important information about specifying performance monitor objects in inputs.conf" later in this topic for a full explanation.

Configure local Windows performance monitoring with Splunk Web

Go to the Add New page

You can get there by two routes:

- Splunk Home
- Splunk Settings

By Splunk Settings:

1. Click Settings in the upper right corner of Splunk Web.
2. Click Data Inputs.
3. Click Local performance monitoring.
4. Click New to add an input.

By Splunk Home:

1. Click the Add Data link in Splunk Home.
2. Click Monitor to monitor performance data from the local Windows machine, or Forward to receive performance data from another machine.
3. If you selected Forward, choose or create the group of forwarders you want this input to apply to.
4. Click Next.

Select the input source

1. In the left pane, locate and select Local Performance Monitoring.
2. In the Collection Name field, enter a unique name for this input that you will remember.
3. Click **Select Object** to get a list of the performance objects available on this Windows machine, then choose the object that you want to monitor from the list. Splunk Enterprise displays the "Select Counters" and "Select Instances" list boxes.

You can only add one performance object per data input. This is due to how Microsoft handles performance monitor objects. Many objects enumerate classes that describe themselves dynamically upon selection. This can lead to confusion as to which performance counters and instances belong to which object, as defined in the input. If you need to monitor multiple objects, create additional data inputs for each object.

4. In the **Select Counters** list box, locate the performance counters you want this input to monitor.
5. Click once on each counter you want to monitor. Splunk Enterprise moves the counter from the "Available counter(s)" window to the "Selected counter(s)" window.
6. To unselect a counter, click on its name in the "Available Items" window. Splunk Enterprise moves the counter from the "Selected counter(s)" window to the "Available counter(s)" window.
7. To select or unselect all of the counters, click on the "add all" or "remove all" links.

Selecting all of the counters can result in the indexing of a lot of data and possibly lead to license violations.

8. In the **Select Instances** list box, select the instances that you want this input to monitor by clicking once on the instance in the "Available instance(s)" window. Splunk Enterprise moves the instance to the "Selected instance(s)" window.

The "_Total" instance is a special instance, and appears for many types of performance counters. This instance is the average of any associated instances under the same counter. Data collected for this instance can be significantly different than for individual instances under the same counter. For example, when you monitor performance data for the "Disk Bytes/Sec" performance counter under the "PhysicalDisk" object on a system with two disks installed, the available instances include one for each physical disk - "0 C:" and "1 D:" - and the "_Total" instance, which is the average of the two physical disk instances.

9. In the **Polling interval** field, enter the time, in seconds, between polling attempts for the input.
10. Click the green **Next** button.

**Specify input settings**

The **Input Settings** page lets you specify application context, default host value, and index. All of these parameters are optional.

Setting the **Host** on this page only sets the **host** field in the resulting events. It does not direct Splunk Enterprise to look on a specific host on your network.

1. Select the appropriate **Application context** for this input.
2. Set the **Host** name value. You have several choices for this setting. Learn more about setting the host value in About hosts.
3. Set the **Index** that Splunk Enterprise should send data to. Leave the value as "default", unless you have defined multiple indexes to handle different types of events. In addition to indexes for user data, Splunk Enterprise has a number of utility indexes, which also appear in this dropdown box.
4. Click **Review**.

**Review your choices**

After you specify input settings, review your selections. Splunk Enterprise lists all options you selected, including the type of monitor, the source, the source type, the application context, and the index.

1. Review the settings.
2. If they do not match what you want, click < to go back to the previous step in the wizard. Otherwise, click **Submit**.

Splunk Enterprise then loads the "Success" page and begins indexing the specified performance metrics. For more information on getting data from files and directories, see Monitor Windows performance in this manual.

**Configure local Windows performance monitoring with configuration files**

 inputs.conf controls performance monitoring configurations. To set up performance monitoring using configuration files, you must create or edit inputs.conf in `%SPLUNK_HOME%\etc\system\local`. If you have not worked with configuration files before, see About configuration files.

The `[perfmon://<name>]` stanza defines performance monitoring inputs in inputs.conf. You specify one stanza per performance object that you wish to
In each stanza, you can specify the following attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>interval</strong></td>
<td>Yes</td>
<td>How often, in seconds, to poll for new data. If this attribute is not present, the input runs every 300 seconds (5 minutes).</td>
</tr>
<tr>
<td><strong>object</strong></td>
<td>Yes</td>
<td>The performance object(s) that you want to capture. Specify either a string which exactly matches (including case) the name of an existing Performance Monitor object or use a regular expression to reference multiple objects. If this attribute is not present and defined, the input will not run, as there is no default.</td>
</tr>
<tr>
<td><strong>counters</strong></td>
<td>Yes</td>
<td>One or more valid performance counters that are associated with the object specified in object. Separate multiple counters with semicolons. You can also use an asterisk (*) to specify all available counters under a given object. If this attribute is not present and defined, the input will not run, there is no default.</td>
</tr>
<tr>
<td><strong>instances</strong></td>
<td>No</td>
<td>One or more valid instances associated with the performance counter specified in counters. Multiple instances are separated by semicolons. Specify all instances by using an asterisk (*), which is the default if you do not define the attribute in the stanza.</td>
</tr>
<tr>
<td><strong>index</strong></td>
<td>No</td>
<td>The index to route performance counter data to. If not present, the default index is used.</td>
</tr>
<tr>
<td><strong>disabled</strong></td>
<td>No</td>
<td>Whether or not to gather the performance data defined in this input. Set to 1 to disable this stanza, and 0 to enable it. If not present, it defaults to 0 (enabled).</td>
</tr>
<tr>
<td><strong>showZeroValue</strong></td>
<td>No</td>
<td>Advanced option. Whether or not Splunk Enterprise should collect events that have values of zero. Set to 1 to collect zero-value events, and 0 to ignore these events. If not present, it defaults to 0 (ignore zero-value events.)</td>
</tr>
<tr>
<td><strong>samplingInterval</strong></td>
<td>No</td>
<td>Advanced option. How often, in milliseconds, that Splunk should collect performance data.</td>
</tr>
</tbody>
</table>
Enables high-frequency performance sampling. When you enable high-frequency performance sampling, Splunk Enterprise collects performance data every interval and reports the average of the data as well as other statistics. It defaults to 100 ms, and must be less than what you specify with the `interval` attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stats</td>
<td>No</td>
<td><strong>Advanced option.</strong> A semicolon-separated list of statistic values that Splunk Enterprise reports for high-frequency performance sampling. Allowed values are: <code>average</code>, <code>min</code>, <code>max</code>, <code>dev</code>, and <code>count</code>. The default is no setting (disabled).</td>
</tr>
<tr>
<td>mode</td>
<td>No</td>
<td><strong>Advanced option.</strong> When you enable high-performance sampling, this attribute controls how Splunk Enterprise outputs events. Allowed values are: <code>single</code>, <code>multikv</code>, <code>multiMS</code>, and <code>multikvMS</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>When you enable either <code>multiMS</code> or <code>multikvMS</code>, Splunk Enterprise outputs two events for each performance metric it collects. The first event is the average value, and the second is the statistics event. The statistics event has a special sourcetype depending on which output mode you use (<code>perfmonMSStats</code> for <code>multiMS</code> and <code>perfmonMKMSStats</code> for <code>multikvMS</code>) If you do not enable high-performance sampling, the <code>multikvMS</code> output mode is the same as the <code>multikv</code> output mode. The default is <code>single</code>.</td>
</tr>
<tr>
<td>useEnglishOnly</td>
<td>No</td>
<td><strong>Advanced option.</strong> Controls how Splunk Enterprise indexes performance metrics on systems whose locale is not English. Specifically, it dictates which Windows Performance Monitor API to use when it indexes performance metrics on hosts that do not use the English language.</td>
</tr>
</tbody>
</table>
If set to true, Splunk Enterprise collects the performance metrics in English regardless of the system locale. It uses the `PdhAddEnglishCounter()` API to add the counter string. It also disables regular expression and wildcard matching for the object and counter attributes.

If set to false, Splunk Enterprise collects the performance metrics in the system language and expects you to configure the object and counter attributes in that language. It uses the `PdhAddCounter()` API to add the counter string. You can use wildcards and regular expressions, but you must specify valid object, counters, and instances values that are specific to the locale of the operating system.

The default is false.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
</table>
| useWinApiProcStats      | No        | **Advanced option.** When enabled, useWinApiProcStats the Performance Monitor input uses process kernel mode and user mode times to calculate CPU usage for a process. Currently, the input uses the standard Performance Data Helper (PDH) APIs to calculate CPU usage for a process. When this setting is configured to "true", the input uses the `GetProcessTime()` function in the core Windows API to calculate CPU usage for a process, for the following Performance Monitor counters, only:

- Processor Time
- User Time
- Privileged Time

It is a best practice to enable the useWinApiProcStats function on multicore Windows machines.

Identify how many processors are in your system. The total number of cores can be verified on the "system information" page in your Windows deployment.

Each processor core in your system is equal to a maximum processor performance percentage of 100%. So for each core in a multicore windows deployment, 100% is added to the total maximum available processor performance.
percentage. For example, an 8 core windows environment will have a maximum process capability of 800%.

Total processor capability can be validated in your Splunk platform deployment by navigating to **Data inputs > Local performance monitoring** > **Select system**.

the APIs this setting uses are English only. If your Windows machine uses a non-English system locale, you must also set `useEnglishOnly` to true.

See Performance Monitor inputs show maximum values of 100 percent usage for a process on multicore Microsoft Windows machines in the release notes for more information on calculating CPU usage on Windows multicore machines.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Required?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>formatString</code></td>
<td>No</td>
<td><strong>Advanced option.</strong> Controls how Splunk Enterprise formats the output of floating-point values for performance counter events. Windows often prints performance counter events as floating point values. When not formatted, the events print with all significant digits to the right of the decimal point. The <code>formatString</code> attribute controls the number of significant digits that print as part of each event. The attribute uses format specifiers from the C++ <code>printf</code> function. The function includes many kinds of specifiers, depending on how you want to output the event text. A reference with examples can be found at &quot;printf - C++ reference&quot;(<a href="http://www.cplusplus.com/reference/cstdio/printf/">http://www.cplusplus.com/reference/cstdio/printf/</a>) on cplusplus.com. When specifying the format, do not use quotes (&quot;&quot;&quot;). Specify only the valid characters needed to format the string the way you want. The default is <code>%20g</code>.</td>
</tr>
</tbody>
</table>
Collect performance metrics in English regardless of system locale

You can collect performance metrics in English even if the system that Splunk Enterprise runs on does not use the English language.

To do this, use the `useEnglishOnly` attribute in stanzas within `inputs.conf`. There is no way to configure `useEnglishOnly` in Splunk Web.

**Note:** There are caveats to using `useEnglishOnly` in an `inputs.conf` stanza. See Caveats later in this topic.

Examples of performance monitoring input stanzas

Following are some example stanzas that show you how to use `inputs.conf` to monitor performance monitor objects.

```
# Query the PhysicalDisk performance object and gather disk access data for
# all physical drives installed in the system. Store this data in the
# "perfmon" index.
# Note: If the interval attribute is set to 0, Splunk resets the
# interval
# to 1.
[perfmon://LocalPhysicalDisk]
interval = 0
object = PhysicalDisk
counters = Disk Bytes/sec; % Disk Read Time; % Disk Write Time; % Disk
Time
instances = *
disabled = 0
index = PerfMon

# Gather SQL statistics for all database instances on this SQL server.
# 'object' attribute uses a regular expression "\$.\.*" to specify SQL
# statistics for all available databases.
[perfmon://SQLServer_SQL_Statistics]
object = MSSQL\$.\.*:SQL Statistics
counters = *
instances = *

# Gather information on all counters under the "Process" and "Processor"
# Perfmon objects.
# We use ".\.*" as a wild card to match the 'Process' and 'Processor' objects.
[perfmon://ProcessandProcessor]
object = Process.\.*
```
counters = *
instances = *

# Collect CPU processor usage metrics in English only on a French system.
[perfmon://Processor]
object = Processor
instances = _Total
counters = % Processor Time;% User Time
useEnglishOnly = 1
interval = 30
disabled = 0

# Collect CPU processor usage metrics in the French system's native locale.
# Note that you must specify the counters in the language of that locale.
[perfmon://FrenchProcs]
counters = *
disabled = 0
useEnglishOnly = 0
interval = 30
object = Processeur
instances = *

# Collect CPU processor usage metrics. Format the output to two decimal places only.
[perfmon://Processor]
counters = *
disabled = 0
interval = 30
object = Processor
instances = *
formatString = %.20g

**Important information about specifying performance monitor objects in inputs.conf**

*Use all lower case when specifying the `perfmon` keyword*

When you create a performance monitor input in `inputs.conf`, you must use all lower case for the `perfmon` keyword, for example:

<table>
<thead>
<tr>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[perfmon://CPUTime]</code></td>
<td><code>[Perfmon://CPUTime]</code></td>
</tr>
<tr>
<td><code>[PERFMON://CPUTime]</code></td>
<td></td>
</tr>
</tbody>
</table>

If you use capital or mixed-case letters for the keyword, Splunk Enterprise warns of the problem on start-up, and the specified performance monitor input does not run.
**Specify valid regular expressions to capture multiple performance monitor objects**

To specify multiple objects in a single performance monitor stanza, you must use a valid regular expression to capture those objects. For example, to specify a wildcard to match a string beyond a certain number of characters, do not use `*`, but rather `.*`. If the object contains a dollar sign or similar special character, you might need to escape it with a backslash (`\`).

**Values must exactly match what is in the Performance Monitor API if you do not use regular expressions**

When you specify values for the `object`, `counters` and `instances` attributes in `[perfmon://]` stanzas, be sure that those values exactly match those defined in the Performance Monitor API, including case, or the input might return incorrect data, or no data at all. If the input cannot match a performance object, counter, or instance value that you've specified, it logs that failure to `splunkd.log`. For example:

```
01-27-2011 21:04:48.681 -0800 ERROR ExecProcessor - message from "C:\Program Files\Splunk\bin\splunk-perfmon.exe" -noui splunk-perfmon - PerfmonHelper::enumObjectByNameEx: PdhEnumObjectItems failed for object - 'USB' with error (0xc0000bb8): The specified object is not found on the system.
```

Use Splunk Web to add performance monitor data inputs to ensure that you add them correctly.

**Enable remote Windows performance monitoring over WMI**

You can configure remote performance monitoring either in Splunk Web or by using configuration files.

When you collect performance metrics over WMI, you must configure Splunk Enterprise to run as an AD user with appropriate access for remote collection of performance metrics. You must do this before attempting to collect those metrics. Both the machine that runs Splunk Enterprise and the machine(s) Splunk collects performance data from must reside in the same AD domain or forest.

WMI self-throttles by design to prevent denial-of-service attacks. Splunk Enterprise also reduces the number of WMI calls it makes over time as a precautionary measure if these calls return an error. Depending on the size, configuration, and security profile of your network, installing a local forwarder on the host that you want to collect performance metrics might be a better choice. See Considerations for deciding how to monitor remote Windows data in this
WMI-based performance values versus Performance Monitor values

When you gather remote performance metrics through WMI, some metrics return zero values or values that are not in line with values that Performance Monitor returns. A limitation in the implementation of WMI for performance monitor counters causes this problem. This is not an issue with Splunk Enterprise or how it retrieves WMI-based data.

WMI uses the Win32_PerfFormattedData_* classes to gather performance metrics. More info on the specific classes is available at "Win32 Classes" on MSDN.

WMI defines the data structures within these classes as either 32- or 64-bit unsigned integers, depending on the version of Windows you run. The PDH API defines Performance Monitor objects as floating-point variables. This means that you might see WMI-based metrics that appear anomalous, due to rounding factors.

For example, if you collect data on the "Average Disk Queue Length" Performance Monitor counter at the same time you collect the Win32_PerfFormattedData_PerfDisk_PhysicalDisk\AvgDiskQueueLength metric through WMI, the WMI-based metric might return zero values even though the Performance Monitor metric returns values that are greater than zero (but less than 0.5). This is because WMI rounds the value down before displaying it.

If you require additional granularity in your performance metrics, configure the performance monitoring inputs on a universal forwarder on each machine from which you wish to collect performance data. You can then forward that data to an indexer. Data retrieved using this method is more reliable than data gathered remotely using WMI-based inputs.

Configure remote Windows performance monitoring with Splunk Web

Go to the Add New page

You can get there by two routes:

- Splunk Home
- Splunk Settings
By Splunk Settings:

1. Click **Settings** in the upper right corner of Splunk Web.
2. Click **Data Inputs**.
3. Click **Remote performance monitoring**.
4. Click **New** to add an input.

By Splunk Home:

1. Click the **Add Data** link in Splunk Home.
2. Click **Monitor** to monitor performance data from the local Windows machine, or **Forward** to forward performance data from another Windows machine. Splunk Enterprise loads the "Add Data - Select Source" page.

   Forwarding performance data requires additional setup.

3. In the left pane, locate and select **Local Performance Monitoring**.

   **Select the input source**

   *Win32_PerfFormattedData_* classes do not show up as available objects in Splunk Web. If you want to monitor *Win32_PerfFormattedData_* classes, you must add them directly in `wmi.conf`.

   1. In the **Collection Name** field, enter a unique name for this input that you will remember.
   2. In the **Select Target Host** field, enter the host name or IP address of the Windows computer you want to collect performance data from.
   3. Click "Query" to get a list of the performance objects available on the Windows machine you specified in the "Select Target Host" field.
   4. Choose the object that you want to monitor from the **Select Class** list. Splunk Enterprise displays the "Select Counters" and "Select Instances" list boxes.

   You can only add one performance object per data input. This is due to how Microsoft handles performance monitor objects. Many objects enumerate classes that describe themselves dynamically upon selection. This can lead to confusion as to which performance counters and instances belong to which object, as defined in the input. If you need to monitor multiple objects, create additional data inputs for each object.

   5. In the **Select Counters** list box, locate the performance counters you want this input to monitor.
6. Click once on each counter you want to monitor. Splunk Enterprise moves the counter from the "Available counter(s)" window to the "Selected counter(s)" window.

7. To unselect a counter, click on its name in the "Available Items" window. Splunk Enterprise moves the counter from the "Selected counter(s)" window to the "Available counter(s)" window.

8. To select or unselect all of the counters, click on the "add all" or "remove all" links. **Important:** Selecting all of the counters can result in the indexing of a lot of data, possibly more than your license allows.

9. In the **Select Instances** list box, select the instances that you want this input to monitor by clicking once on the instance in the "Available instance(s)" window. Splunk Enterprise moves the instance to the "Selected instance(s)" window.

The "_Total" instance is a special instance, and appears for many types of performance counters. This instance is the average of any associated instances under the same counter. Data collected for this instance can be significantly different than for individual instances under the same counter. For example, when you monitor performance data for the "Disk Bytes/Sec" performance counter under the "PhysicalDisk" object on a host with two disks installed, the available instances include one for each physical disk - "0 C:" and "1 D:" - and the "_Total" instance, which is the average of the two physical disk instances.

10. In the **Polling interval** field, enter the time, in seconds, between polling attempts for the input.

11. Click **Next**.

**Specify input settings**

The **Input Settings** page lets you specify application context, default host value, and index. All of these parameters are optional.

Setting the **Host** only sets the host field in the resulting events. It does not direct Splunk Enterprise to look on a specific host on your network.

1. Select the appropriate **Application context** for this input.
2. Set the **Host** name value. You have several choices for this setting. Learn more about setting the host value in About hosts.
3. Set the **Index** that Splunk Enterprise should send data to. Leave the value as "default", unless you have defined multiple indexes to handle different types of events. In addition to indexes for user data, Splunk Enterprise has a number of utility indexes, which also appear in this dropdown box.
4. Click the green **Review** button.

**Review your choices**

After specifying all your input settings, you can review your selections. Splunk Enterprise lists all options you selected, including the type of monitor, the source, the source type, the application context, and the index.

1. Review the settings.
2. If they do not match what you want, click < to go back to the previous step in the wizard. Otherwise, click **Submit**.

Splunk Enterprise then loads the "Success" page and begins indexing the specified performance metrics.

For more information on getting performance monitor data from remote machines, see Monitor WMI data in this manual.

**Configure remote Windows performance monitoring with configuration files**

wmi.conf controls remote performance monitoring configurations. To set up remote performance monitoring using configuration files, create and/or edit `wmi.conf` in `%SPLUNK_HOME%\etc\system\local`. If you haven't worked with configuration files before, read About configuration files before you begin.

Use Splunk Web to create remote performance monitor inputs unless you do not have access to it. The names of performance monitor objects, counters, and instances must exactly match what the Performance Monitor API defines, including case. Splunk Web uses WMI to get the properly-formatted names, eliminating the potential for typos.

`wmi.conf` contains one stanza for each remote performance monitor object that you want to monitor. In each stanza, you specify the following content.

**Global settings**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Required?</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial_backoff</td>
<td>No</td>
<td>How long, in seconds, to wait before retrying a connection to a WMI provider when an error</td>
<td>5</td>
</tr>
</tbody>
</table>
occurs. If problems persist on connecting to the provider, then the wait time between connection attempts doubles until either it can connect, or until the wait time is greater than or equal to the `max_backoff` attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Required?</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>max_backoff</code></td>
<td>No</td>
<td>The maximum amount of time, in seconds to attempt to reconnect to a WMI provider.</td>
<td>20</td>
</tr>
<tr>
<td><code>max_retries_at_max_backoff</code></td>
<td>No</td>
<td>How many times, after <code>max_backoff</code> seconds has been reached between reconnection attempts with a WMI provider, to continue to attempt to reconnect to that provider.</td>
<td>2</td>
</tr>
<tr>
<td><code>checkpoint_sync_interval</code></td>
<td>No</td>
<td>How long, in seconds, to wait for state data to be flushed to disk.</td>
<td>2</td>
</tr>
</tbody>
</table>

**Input-specific settings**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Required?</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>interval</code></td>
<td>Yes</td>
<td>How often, in seconds, to poll for new data. If this attribute is not present, the input will not run, as there is no default.</td>
<td>N/A</td>
</tr>
<tr>
<td><code>server</code></td>
<td>No</td>
<td>A comma-separated list of one or more valid hosts on which you want to monitor performance.</td>
<td>The local machine</td>
</tr>
<tr>
<td><code>event_log_file</code></td>
<td>No</td>
<td>The names of one or more Windows event log channels to poll. This attribute tells Splunk Enterprise that the incoming data is</td>
<td>N/A</td>
</tr>
<tr>
<td>Attribute</td>
<td>Required?</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>wql</td>
<td>No</td>
<td>A valid Windows Query Language (WQL) statement that specifies the performance objects, counters, and instances you want to poll remotely. This attribute tells Splunk Enterprise to expect data from a WMI provider. Do not use the <code>wql</code> attribute in a stanza that already contains the <code>event_log_file</code> attribute.</td>
<td>N/A</td>
</tr>
<tr>
<td>namespace</td>
<td>No</td>
<td>The namespace in which the WMI provider you want to query resides. The value for this attribute can be either relative (<code>Root\CIMV2</code>) or absolute (<code>\SERVER\Root\CIMV2</code>), but must be relative if you specify the <code>server</code> attribute. Only use the <code>namespace</code> attribute in a stanza that contains the <code>wql</code> attribute.</td>
<td><code>Root\CIMV2</code></td>
</tr>
<tr>
<td>index</td>
<td>No</td>
<td>The desired index to route performance counter data to.</td>
<td><code>default</code></td>
</tr>
<tr>
<td>current_only</td>
<td>No</td>
<td>The characteristics and interaction of WMI-based event collections.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• if <code>wql</code> is defined, this attribute tells Splunk Enterprise whether or not it should expect an event</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Required?</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>notification query</td>
<td></td>
<td>notification query. Set to 1 to tell Splunk to expect an event notification query, and 0 to tell it expect a standard query. See below for additional requirements on WQL and event notification queries.</td>
<td></td>
</tr>
<tr>
<td>event_log_file</td>
<td></td>
<td>if event_log_file is defined, tells Splunk whether or not to only capture events that occur when Splunk is running. Set to 1 to tell Splunk to only capture events that occur when Splunk is running, and 0 to gather events from the last checkpoint or, if no checkpoint exists, the oldest events available.</td>
<td></td>
</tr>
<tr>
<td>disabled</td>
<td>No</td>
<td>Tells Splunk whether or not to gather the performance data defined in this input. Set this to 1 to disable performance monitoring for this stanza, and 0 to enable it.</td>
<td>0</td>
</tr>
</tbody>
</table>

**Examples of using wmi.conf**

The following example of wmi.conf gathers local disk and memory performance metrics and places them into the 'wmi_perfmon' index:

```plaintext
[settings]
initial_backoff = 5
max_backoff = 20
max_retries_at_max_backoff = 2
checkpoint_sync_interval = 2

# Gather disk and memory performance metrics from the local system every second.
# Store event in the "wmi_perfmon" Splunk index.

[WMI:LocalPhysicalDisk]
interval = 1
```

19
wql = select Name, DiskBytesPerSec, 
PercentDiskReadTime, PercentDiskWriteTime, PercentDiskTime from \ 
Win32_PerfFormattedData_PerfDisk_PhysicalDisk
disabled = 0
index = wmi_perfmon

[WMI:LocalMainMemory]
interval = 10
wql = select CommittedBytes, AvailableBytes, 
PercentCommittedBytesInUse, Caption from \ 
Win32_PerfFormattedData_PerfOS_Memory
disabled = 0
index = wmi_perfmon

**Additional information on WQL query statements**

WQL queries must be structurally and syntactically correct. If they are not, you might get undesirable results or no results at all. In particular, when writing event notification queries (by specifying `current_only=1` in the stanza in which a WQL query resides), your WQL statement must contain one of the clauses that specify such a query (`WITHIN`, `GROUP`, and/or `HAVING`). Review this MSDN article on Querying with WQL for additional information.

Splunk Web eliminates problems with WQL syntax by generating the appropriate WQL queries when you use it to create performance monitor inputs.

**Caveats to using the performance monitoring input**

**Increased memory usage during collection of performance metrics**

When you collect data on some performance objects, such as the "Thread" object and its associated counters, you might notice increased memory usage in Splunk. This is normal, as certain performance objects consume more memory than others during the collection process.

**Processor Time counters do not return values of higher than 100**

Due to how Microsoft tallies CPU usage with the `Processor:% Processor Time` and `Process:% Processor Time` counters, these counters do not return a value of more than 100 regardless of the number of CPUs or cores in the system. This is by design - these counters subtract the amount of time spent on the Idle process from 100%.
On non-English installations, the useEnglishOnly attribute has usage limitations

When you edit inputs.conf on a non-English system to enable performance monitoring, there are some limitations to how the useEnglishOnly attribute works.

If you set the attribute to true, you cannot use wildcards or regular expressions for the object and counters attributes. These attributes must contain specific entries based on valid English values as defined in the Performance Data Helper library. You can specify a wildcard for the instances attribute. Here's an example:

```plaintext
[perfmon://Processor]
object = Processor
instances = _Total
counters = % Processor Time;% User Time
useEnglishOnly = 1
interval = 30
disabled = 0

The counters attribute contain values in English even though the system language is not English.

If you set the attribute to false, you can use wildcards and regular expressions for these attributes, but you must specify values based on the operating system’s language. An example of a stanza on a system running in French follows:

```plaintext
[perfmon://FrenchProcs]
counters = *
disabled = 0
useEnglishOnly = 0
interval = 30
object = Processeur
instances = *
```

Note in this example that the object attribute has been set to Processeur, which is the French equivalent of Processor. If you specify English values here, Splunk Enterprise will not find the performance object or instance.

Additional impacts of using the useEnglishOnly attribute

There are additional items to consider when using the attribute.

- When you use Splunk Web to create performance monitor inputs on a non-English operating system, it always specifies useEnglishOnly = false.
Additionally, you can enable, disable, clone, or delete these stanzas within Splunk Web. You cannot, however, edit them in Splunk Web unless the operating system's locale matches the locale specified in the stanza.

You can use Splunk Web to enable, disable, clone, or delete a performance monitor stanza with the `useEnglishOnly` attribute set to true. However, you cannot edit them in Splunk Web unless the system's locale is English.