Monitor Windows Registry data
Monitor Windows Registry data

The Windows Registry is the central configuration database on a Windows machine. Nearly all Windows processes and third-party programs interact with it. Without a healthy Registry, Windows does not run. Splunk Enterprise supports the capture of Windows Registry settings and lets you monitor changes to the Registry in real time.

When a program makes a change to a configuration, it writes those changes to the Registry. Later, when the program runs again, it looks into the Registry to read those configurations. You can learn when Windows programs and processes add, update, and delete Registry entries on your system. When a Registry entry changes, Splunk Enterprise captures the name of the process that made the change, as well as the entire path to the entry being changed.

The Windows Registry input monitor runs as a process called `splunk-regmon.exe`.

If you have Splunk Cloud, you must use the universal forwarder to collect data from the Windows Registry and forward it to your Splunk Cloud deployment.

Why monitor the Registry?

The Registry is probably the most used, yet least understood component of Windows operation. Many programs and processes read from and write to it at all times. When something is not functioning, Microsoft often instructs administrators and users alike to make changes to the Registry directly using the RegEdit tool. The ability to capture those edits, and any other changes, in real time is the first step in understanding the importance of the Registry.

Registry health is very important. Splunk Enterprise tells you when changes to the Registry are made and also if those changes were successful. If programs and processes can't write to or read from the Registry, a system failure can occur. Splunk Enterprise can alert you to problems interacting with the Registry so that you can restore it from a backup and keep your system running.

What do you need to monitor the Registry?

The following table lists the explicit permissions you need to monitor the Registry. You might need additional permissions based on the Registry keys that you want to monitor.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Required permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor the Registry</td>
<td>• Splunk Enterprise or the universal forwarder must be run on Windows</td>
</tr>
<tr>
<td></td>
<td>• The Splunk service must be configured to use the Local System user, or run as a domain user with read access to the Registry hives or keys that you want to monitor</td>
</tr>
</tbody>
</table>

Performance considerations

When you enable Registry monitoring, you specify which Registry hives to monitor: the user hive (represented as `HKEY_USERS` in RegEdit) and/or the machine hive (represented as `HKEY_LOCAL_MACHINE`). The user hive contains user-specific configurations required by Windows and programs, and the machine hive contains configuration information specific to the machine, such as the location of services, drivers, object classes and security descriptors.

Because the Registry plays a central role in the operation of a Windows machine, enabling both Registry paths results in a lot of data for Splunk Enterprise to monitor. To achieve the best performance, filter the amount of Registry data that Splunk Enterprise indexes by configuring `inputs.conf`. 
Similarly, you can capture a baseline snapshot of the current state of your Windows Registry when you first start Splunk Enterprise, and again every time a specified amount of time has passed. The snapshot lets you compare what the Registry looks like at a certain point in time and provides for easier tracking of the changes to the Registry over time.

The snapshot process can be somewhat CPU-intensive, and might take several minutes to complete. You can postpone taking a baseline snapshot until you have narrowed the scope of the Registry entries to those you specifically want Splunk Enterprise to monitor.

**Enable Registry monitoring in 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 **Registry 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 Registry data on the local Windows machine.

**Select the input source**

1. In the left pane, locate and select **Registry monitoring**.
2. In the **Collection Name** field, enter a unique name for the input that you will remember.
3. In the **Registry hive** field, enter the path to the Registry key that you want Splunk Enterprise to monitor.
4. (Optional) If you are not sure of the path, click the **Browse** button to select the Registry key path that you want Splunk Enterprise to monitor. The **Registry hive** window opens and displays the Registry in tree view. Hives, keys and subkeys display as folders, and values display as document icons. The **HKEY_USERS**, **HKEY_CURRENT_USER**, **HKEY_LOCAL_MACHINE**, and **HKEY_CURRENT_CONFIG** hives display as top-level objects. The **HKEY_CLASSES_ROOT** hive is not shown because of the number of subkeys present in the first sublevel of that hive. **To access HKEY_CLASSES_ROOT items, choose HKEY_LOCAL_MACHINE\Software\Classes.**
5. In the **Registry hive** window, choose the desired Registry key by clicking on the name of the key. The qualified key name appears in the **Qualified name** field at the bottom of the window.
6. Click **Select** to confirm the choice and close the window.
7. (Optional) Select **Monitor subnodes** if you want to monitor the child nodes below the starting hive.

The **Monitor subnodes** node determines what Splunk Enterprise adds to the **inputs.conf** file that it creates when you define a Registry monitor input in Splunk Web.

If you use the tree view to select a key or hive to monitor and check **Monitor subnodes**, then Splunk Enterprise adds a **regular expression** to the stanza for the input you are defining. This regular expression (\\\?\.* ) filters out events that do not directly reference the selected key or any of its subkeys.
If you do not check Monitor subnodes, then Splunk Enterprise adds a regular expression to the input stanza which filters out events that do not directly reference the selected key (including events that reference subkeys of the selected key.)

If you do not use the tree view to specify the desired key to monitor, then Splunk Enterprise adds the regular expression only if you have checked Monitor subnodes and have not entered your own regular expression in the Registry hive field.

8. Under **Event types**, select the Registry event types that you want Splunk Enterprise to monitor for the chosen Registry hive:

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td><strong>Set</strong> Splunk Enterprise generates a Set event when a program executes a SetValue method on a Registry subkey, thus setting a value or overwriting an existing value on an existing Registry entry.</td>
</tr>
<tr>
<td>Create</td>
<td><strong>Create</strong> Splunk Enterprise generates a Create event when a program executes a CreateSubKey method within a Registry hive, thus creating a new subkey within an existing Registry hive.</td>
</tr>
<tr>
<td>Delete</td>
<td><strong>Delete</strong> Splunk Enterprise generates a Delete event when a program executes a DeleteValue or DeleteSubKey method. This method either removes a value for a specific existing key, or removes a key from an existing hive.</td>
</tr>
<tr>
<td>Rename</td>
<td><strong>Rename</strong> Splunk Enterprise generates a Rename event when you rename a Registry key or subkey in RegEdit.</td>
</tr>
<tr>
<td>Open</td>
<td><strong>Open</strong> Splunk Enterprise generates an Open event when a program executes an OpenSubKey method on a Registry subkey, such as what happens when a program needs configuration information contained in the Registry.</td>
</tr>
<tr>
<td>Close</td>
<td><strong>Close</strong> Splunk Enterprise generates a Close event when a program executes a Close method on a Registry key. This happens when a program is done reading the contents of a key, or after you make a change to a key’s value in RegEdit and exit the value entry window.</td>
</tr>
<tr>
<td>Query</td>
<td><strong>Query</strong> Splunk Enterprise generates a Query event when a program executes the GetValue method on a Registry subkey.</td>
</tr>
</tbody>
</table>

9. Specify which processes Splunk Enterprise should monitor for changes to the Registry by entering appropriate values in the **Process Path** field. Or, leave the default of .* to monitor all processes.

10. Specify whether or not you want to take a baseline snapshot of the whole Registry before monitoring Registry changes. To set a baseline, click **Yes** under Baseline index.

    The baseline snapshot is an index of your entire Registry, at the time the snapshot is taken. Registry events within the snapshot retain their original indexing timestamps. Scanning the Registry to set a baseline index is a CPU-intensive process and might take some time.

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.

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.
   
   **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.

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 specifying all your 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 Registry nodes.

View Registry change data

To view Registry change data that Splunk Enterprise indexed, go to the Search app and search for events with a source of WinRegistry. An example event, which Group Policy generates when a user logs in to a domain, follows:

3:03:28.505 PM
06/19/2011 15:03:28.505
event_status="(0) The operation completed successfully."
pid=340
process_image="c:\WINDOWS\system32\winlogon.exe"
registry_type="SetValue"
key_path="HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\History\DCName"
data_type="REG_SZ"
data="\\ftw.ad.splunk.com"

Each registry monitoring event contains the following attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event_status</td>
<td>The result of the registry change attempt. This should always be &quot;(0) The operation completed successfully.&quot; If it is not, there might be problems with the Registry that might eventually require a restore from a backup.</td>
</tr>
<tr>
<td>pid</td>
<td>The process ID of the process that attempted to make the Registry change.</td>
</tr>
<tr>
<td>process_image</td>
<td>The name of the process that attempted to make the Registry change.</td>
</tr>
<tr>
<td>registry_type</td>
<td>The type of Registry operation that the process_image attempted to invoke.</td>
</tr>
<tr>
<td>key_path</td>
<td>The Registry key path that the process_image attempted to make a change to.</td>
</tr>
<tr>
<td>data_type</td>
<td>The type of Registry data that the process_image making the Registry change tried to get or set.</td>
</tr>
<tr>
<td>data</td>
<td>The data that the process_image making the Registry change tried to read or write.</td>
</tr>
</tbody>
</table>

Filter incoming Registry events

Windows Registries generate a great number of events due to their near-constant use. This can cause problems with licensing. Splunk Registry monitoring can generate hundreds of megabytes of data per day.

Splunk Windows Registry monitoring uses a configuration file to determine what to monitor on your system, inputs.conf. This file needs to reside in $SPLUNK_HOME\etc\system\local\ on the server that runs Registry monitoring.

inputs.conf contains the specific regular expressions you create to refine and filter the Registry hive paths you want Splunk to monitor.

Each stanza in inputs.conf represents a particular filter whose definition includes:
**Attribute** | **Description**
--- | ---
**proc** | A regular expression containing the path to the process or processes you want to monitor. Default: .*

**hive** | A regular expression that contains the hive path to the entry or entries you want to monitor. Splunk supports the root key value mappings predefined in Windows:

- `\REGISTRY\USER\.*` maps to `HKEY_USERS` or HKU
- `\REGISTRY\USER\_Classes` maps to `HKEY_CLASSES_ROOT` or HKCR
- `\REGISTRY\MACHINE\.*` maps to `HKEY_LOCAL_MACHINE` or HKLM
- `\REGISTRY\MACHINE\SOFTWARE\Classes` maps to `HKEY_CLASSES_ROOT` or HKCR
- `\REGISTRY\MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current` maps to `HKEY_CURRENT_CONFIG` or HKCC
- There is no direct mapping for `HKEY_CURRENT_USER` or HKCU, as the Registry monitor runs in kernel mode. Use `\REGISTRY\USER\.*` (note the period and asterisk at the end) to generate events that contain the security identifier (SID) of the logged-in user.
- Alternatively, you can specify the user whose Registry keys you wish to monitor by using `\REGISTRY\USER\<SID>`, where SID is the SID of the user.

**type** | The subset of event types to monitor. Can be one or more of `delete`, `set`, `create`, `rename`, `open`, `close`, or `query`. The values here must be a subset of the values for `event_types` that you set in `inputs.conf`.

**baseline** | Whether or not to capture a baseline snapshot for that particular hive path. Set to 1 for yes, and 0 for no.

**baseline_interval** | How much time, in seconds, must have elapsed since the last baseline was taken before Splunk Enterprise takes another baseline on startup. For example, if you set `baseline_interval` to 600, then when Splunk Enterprise starts or restarts, it takes a baseline if the existing baseline is more than 600 seconds old. If no baseline exists, then Splunk Enterprise takes a baseline immediately. This setting has no effect if you do not also set `baseline` to 1. The default value is 86,400 seconds (1 day).

**disabled** | Whether or not a filter is enabled. Set to 1 to disable the filter, and 0 to enable it.

### Get a baseline snapshot

When you enable Registry monitoring, you can record a baseline snapshot of the Registry hives the next time Splunk Enterprise starts. By default, the snapshot covers the `HKEY_CURRENT_USER` and `HKEY_LOCAL_MACHINE` hives. It also establishes a timeline for when to retake the snapshot. By default, if the baseline is more than 24 hours old, when Splunk Enterprise next starts, it retakes the baseline snapshot. You can customize this value for each of the filters in `inputs.conf` by setting the value of `baseline_interval`, in seconds.

When you create a baseline snapshot, the snapshot uses the index time of the Registry data, not the snapshot creation time. For example, if a change to a Registry key occurred two years ago, the timestamp for that event will be two years ago, not when the baseline snapshot was created.