



Splunk® Enterprise 6.4.0

Getting Data In

Generated: 5/04/2016 2:49 am

Table of Contents

Introduction.....	1
What Splunk Enterprise can index.....	1
Get started with getting data in.....	3
Is my data local or remote?.....	5
Use forwarders to get data in.....	7
Use apps to get data in.....	8
Configure your inputs.....	9
How Splunk Enterprise handles your data.....	12
 How to get data into Splunk Enterprise.....	 15
How do you want to add data?.....	15
Upload data.....	16
Monitor data.....	17
Forward data.....	18
The Set Sourcetype page.....	20
Prepare your data for previewing.....	26
Modify event processing.....	27
Modify input settings.....	33
Distribute source type configurations.....	35
 Get data from files and directories.....	 38
Monitor files and directories.....	38
Monitor files and directories with Splunk Web.....	41
Monitor files and directories with the CLI.....	43
Monitor files and directories with inputs.conf.....	46
Specify input paths with wildcards.....	52
Whitelist- or blacklist-specific incoming data.....	56
How Splunk Enterprise handles log file rotation.....	59
 Get data from network sources.....	 61
Get data from TCP and UDP ports.....	61
Set up and use HTTP Event Collector.....	71
How Splunk Enterprise handles syslog data.....	78
Send SNMP events to Splunk Enterprise.....	82
 Get Windows data.....	 86
About Windows data and Splunk Enterprise.....	86
How to get Windows data into Splunk Enterprise.....	87
Considerations for deciding how to monitor remote Windows data.....	89

Table of Contents

Get Windows data	
Monitor Active Directory.....	95
Monitor Windows event log data.....	107
Monitor file system changes.....	130
Monitor data through Windows Management Instrumentation (WMI).....	136
Monitor Windows Registry data.....	148
Monitor Windows performance.....	156
Monitor Windows data with PowerShell scripts.....	176
Monitor Windows host information.....	181
Monitor Windows printer information.....	187
Monitor Windows network information.....	191
Get other kinds of data in.....	201
Monitor First In, First Out (FIFO) queues.....	201
Monitor changes to your file system.....	203
Get data from APIs and other remote data interfaces through scripted inputs.....	210
Find more data sources to monitor with crawl.....	217
Configure event processing.....	219
Overview of event processing.....	219
Configure character set encoding.....	219
Configure event line breaking.....	225
Configure event timestamps.....	231
Configure indexed field extraction.....	232
Anonymize data.....	232
Configure timestamps.....	238
How timestamp assignment works.....	238
Configure timestamp recognition.....	240
Configure timestamp assignment for events with multiple timestamps..	251
Specify time zones for timestamps.....	253
Tune timestamp recognition for better indexing performance.....	256
Configure indexed field extraction.....	258
About indexed field extraction.....	258
About default fields (host, source, sourcetype, and more).....	259
Assign default fields dynamically.....	262
Create custom fields at index time.....	263

Table of Contents

Configure indexed field extraction	
Extract fields from files with structured data.....	273
Configure host values.....	285
About hosts.....	285
Set a default host for a Splunk Enterprise server.....	287
Set a default host for a file or directory input.....	289
Set host values based on event data.....	295
Change host values after indexing.....	297
Configure source types.....	299
Why source types matter.....	299
Override automatic source type assignment.....	303
Configure rule-based source type recognition.....	307
List of pretrained source types.....	309
Override source types on a per-event basis.....	314
Create source types.....	317
Manage source types.....	319
Rename source types at search time.....	324
Manage event segmentation.....	326
About event segmentation.....	326
Set the segmentation for event data.....	328
Set search-time event segmentation in Splunk Web.....	331
Improve the data input process.....	332
Use a test index to test your inputs.....	332
Use persistent queues to help prevent data loss.....	334
Troubleshoot the input process.....	336

Introduction

What Splunk Enterprise can index

The first step in using Splunk Enterprise is to feed it data. Once Splunk Enterprise gets some data, it indexes the data and makes it available for searching. Splunk Enterprise transforms your data into a series of **events** that consist of searchable fields. You can massage the data before and after Splunk indexes it, but this is usually not necessary. After the data has been indexed, you can search it, or use it to create charts, reports, alerts, and other interesting output.

What kind of data?

Any kind. In particular, any and all IT streaming, machine, and historical data, such as Windows event logs, web server logs, live application logs, network feeds, system metrics, change monitoring, message queues, archive files, and so on.

Point Splunk Enterprise at a data source. Tell it a bit about the source. That source then becomes a data input. Splunk Enterprise indexes the data stream and transforms it into a series of events. You can view and search those events right away. If the results aren't exactly what you want, you can tweak the indexing process until they are.

The data can be on the same machine as the Splunk Enterprise **indexer (local data)**, or it can be on another machine (**remote data**). You can get remote data into Splunk Enterprise by either using network feeds or by installing Splunk **forwarders** on the hosts where the data originates. For more information on local vs. remote data, see [Where is my data?](#)

Splunk offers **apps** and **add-ons**, with pre-configured inputs for things like Windows- or Linux-specific data sources, Cisco security data, Blue Coat data, and so on. Look on Splunkbase for an app or add-on that fits your needs. Splunk also comes with dozens of recipes for data sources like web server logs, Java 2 Platform, Enterprise Edition (J2EE) logs, or Windows performance metrics. You can get to these from the **Add data** page in Splunk Web. If the recipes and apps don't cover your needs, then you can use the general input configuration capabilities of Splunk Enterprise to specify your particular data source.

For more information on how to configure data inputs for Splunk Enterprise, see [Configure your inputs](#).

Types of data sources

Splunk provides tools to configure many kinds of data inputs, including those that are specific to particular application needs. Splunk also provides the tools to configure any arbitrary data input types. In general, you can categorize Splunk inputs as follows:

- Files and directories
- Network events
- Windows sources
- Other sources

Files and directories

A lot of data comes directly from files and directories. You can use the Splunk Enterprise **files and directories monitor** input processor to get data from files and directories.

To monitor files and directories, see [Get data from files and directories](#).

Network events

Splunk Enterprise can index data from any network port. For example, Splunk can index remote data from `syslog-ng` or any other application that transmits over the TCP protocol. It can also index UDP data, but you should use TCP instead whenever possible for enhanced reliability.

Splunk Enterprise can also receive and index SNMP events, alerts fired off by remote devices.

To get data from network ports, see [Get data from TCP and UDP ports](#) in this manual.

To get SNMP data, see [Send SNMP events to Splunk](#) in this manual.

Windows sources

The Windows version of Splunk Enterprise includes a wide range of Windows-specific inputs. It also provides pages in Splunk System for defining the following Windows-specific input types:

- [Windows Event Log data](#)
- [Windows Registry data](#)
- [WMI data](#)
- [Active Directory data](#)
- [Performance monitoring data](#)

Note: To index and search Windows data on a non-Windows instance of Splunk Enterprise, you must first use a Windows instance to gather the data. See [Considerations for deciding how to monitor remote Windows data](#).

For a more detailed introduction to using Windows data in Splunk Enterprise, see [About Windows data and Splunk Enterprise](#) in this manual.

Other data sources

Splunk Enterprise also supports other kinds of data sources. For example:

- [First-in, first-out \(FIFO\) queues](#)
- [Scripted inputs](#)
Get data from APIs and other remote data interfaces and message queues.
- [Modular inputs](#)
Define a custom input capability to extend the Splunk Enterprise framework.

Get started with getting data in

To get started with getting data into Splunk Enterprise, point it at some data by [configuring an input](#) from the **Add data** page in Splunk Web.

Alternatively, you can download and enable an [app](#), such as the Splunk App for Windows Infrastructure or Splunk App for Unix and Linux.

After you install Splunk Enterprise and either configure the inputs or enable an app, Splunk Enterprise stores and processes the specified data. You can go to either the Search app or the main app page and begin exploring the data that you collected.

- To learn how to add data to Splunk Enterprise, see [How do you want to add data?](#) in this manual.

- To learn how to experiment with adding a test index, see [Use a test index](#).
- To learn about how to add source types in Splunk Enterprise, see "[The Set Sourcetype page](#)."
- To learn what event processing is and how to configure it, see [How Splunk Enterprise handles your data](#).
- To learn how to delete data from Splunk Enterprise, see [Delete indexed data and start over](#).
- To learn about how to configure your inputs with a default index, see [Point your inputs at the default index](#).

Add new inputs

Here is a high-level procedure for adding data.

1. Understand your needs. Questions you can ask yourself include:

- What kind of data do I want Splunk Enterprise to index? See [What Splunk Enterprise can index](#).
- Is there an app for that? See [Use apps to get data in](#).
- Where does the data reside? Is it local or remote? See [Where is my data?](#)
- Should I use forwarders to access remote data? See [Use forwarders to get data in](#).
- What do I want to do with the indexed data? See [What is Splunk knowledge?](#) in the *Knowledge Manager Manual*.

2. Create a test index and add a few inputs. Any data you add to your test index counts against your maximum daily indexing volume for licensing purposes.

3. Preview and modify how Splunk Enterprise indexes your data before committing the data to the test index.

4. Review the test data that you have added with the Search app:

- Do you see the sort of data you were expecting?
- Did the default configurations work well for your events?
- Is data missing or mangled?
- Are the results optimal?

5. If necessary, tweak your input and event processing configurations further until events look the way you want them to.

6. Delete the data from your test index and start over, if necessary.

7. When you are ready to index the data permanently, point your inputs to the default main index.

Repeat this task when you have other inputs to add.

Index custom data

Splunk Enterprise can index any time-series data, usually without additional configuration. If you have logs from a custom application or device, let Splunk Enterprise process it with the default configuration first. If you do not get the results you want, tweak some things to make sure Splunk Enterprise indexes your events correctly.

See "[Overview of event processing](#)" and "How Splunk Enterprise indexes data" so that you can make decisions about how to make Splunk Enterprise work with your data. Here are some issues to consider:

- Are your events multiline? See [Configure event line breaking](#)."
- Is your data in an unusual character set? See "[Configure character set encoding](#)."
- Is Splunk Enterprise unable to determine the timestamps correctly? See "[How Splunk Enterprise extracts timestamps](#)."

Is my data local or remote?

Whether you have just begun to get data into Splunk Enterprise or you have worked with the software for a long time, you might be confused as to what differentiates "local" data from "remote" data in the context of Splunk operations.

The answer to this question depends on a number of things, which include:

- The operating system on which your Splunk Enterprise instance resides.
- The types of data storage that are connected to the Splunk Enterprise instance.
- Whether or not you need to perform any authentication or other intermediate to access the data store that contains the data you want to index.
- Whether or not you run Splunk Cloud or a version of Splunk Enterprise in the cloud.

Local Data

A local resource is a fixed resource that your Splunk Enterprise instance has direct access to. You are able to access a local resource, and whatever it contains, without having to attach, connect, or perform any other intermediate action (such as authentication or mapping a network drive). If your data is on such a resource, the data is considered local.

Some examples of local data include:

- A hard disk or solid state drive installed in a desktop, laptop, or server host.
- A RAM disk.

Remote Data

A remote resource is any resource that does not meet the definition of a "local" resource. Some examples that qualify are:

- Network drives on Windows hosts.
- Active Directory schemas.
- NFS or other network-based mounts on *nix hosts.
- Most cloud-based resources.

Data gathered from these resource endpoints is remote.

Exceptions

Some cases where resources might be considered remote are actually not remote. Here are some examples.

- A host has a volume that has been permanently mounted over a high-bandwidth physical connection such as USB or FireWire. Because the computer can mount the resource at boot time, Splunk Enterprise treats it as a local resource, even though the resource can theoretically be disconnected at a later time.
- A host has a resource that has been permanently mounted over a high-bandwidth network standard such as iSCSI, or to a Storage Area Network over fiber. As the standard treats such volumes as local block devices, such a resource would be considered local.

Use forwarders to get data in

Forwarders are Splunk Enterprise instances that consume data and send it to Splunk Enterprise **indexers** for processing. They require minimal resources and have little impact on performance, so they can usually reside on the machines where the data originates.

For example, if you have a number of Apache Web servers that generate data that you want to search centrally, you can install a Splunk Enterprise indexer and then set up forwarders on the Apache hosts. The forwarders take the Apache data and send it on to the indexer, which then consolidates, stores, and makes it available for searching. Because of their reduced resource footprint, forwarders have minimal performance impact on the Apache servers.

Similarly, you can install forwarders on your employees' Windows desktops. These can send logs and other data to a central Splunk Enterprise instance, where you can view the data as a whole to track malware or other issues. The Splunk App for Windows Infrastructure relies on this kind of deployment.

What forwarders do

Forwarders get data from remote machines. They represent a more robust solution than raw network feeds, with their capabilities for the following actions:

- Tagging of metadata (source, sourcetype, and host)
- Configurable buffering
- Data compression
- SSL security
- Use of any available network ports
- Running scripted inputs locally

Forwarders consume data like any other Splunk Enterprise instance. They can handle exactly the same types of data as an indexer. The difference is that forwarders usually do not index the data themselves. Instead, they get the data and send it on to an indexer, which does the indexing and searching. A single indexer can process data that comes from many forwarders. For detailed information on forwarders, see the *Forwarding Data* or *Universal Forwarder* manuals.

In most Splunk Enterprise deployments, forwarders serve as the primary consumers of data. It is only in single-machine deployments that the indexer might also be the main data consumer. In a large Splunk Enterprise deployment,

you might have hundreds or even thousands of forwarders that consume data and forward it on to a group of indexers for consolidation.

How to configure forwarder inputs

As lightweight instances of Splunk Enterprise, forwarders have limited capabilities by design. For example, most forwarders do not include Splunk Web, which means no web interface is available to set up data inputs. Here are the main ways that you can configure data inputs on a forwarder:

- Specify inputs during initial deployment.
- For Windows forwarders, specify common inputs during the installation process.
- For *nix forwarders, specify inputs directly after installation.
- Use the CLI.
- Edit `inputs.conf`.
- Install an app that contains the inputs you want.
- Use Splunk Web on a full Splunk Enterprise test instance to configure the inputs and then distribute the resulting `inputs.conf` file to the forwarder.

Forwarder Topologies and Deployments

- For information on forwarders, including use cases, typical topologies, and configurations, see About forwarding and receiving in the *Forwarding Data* manual.
- For details on how to deploy the universal forwarder, including how to use the **deployment server** to simplify distribution of configuration files and apps to multiple forwarders, see Example forwarder deployment topologies in the *Universal Forwarder* manual.

Use apps to get data in

Splunk offers **apps** and **add-ons** that extend the capability and simplify the process of getting data into Splunk Enterprise. You can find an app with the data inputs configured. Download apps from Splunkbase.

Apps typically target specific data types and handle everything from configuring the inputs to generating useful views of the data. For example, the Splunk App for Windows Infrastructure provides data inputs, searches, reports, alerts, and dashboards for Windows host management. The Splunk App for Unix and Linux offers the same for Unix and Linux environments. There is a wide range of apps

that handle specific types of application data, such as:

- Splunk IT Service Intelligence
- Splunk App for F5
- Splunk App for Cisco Security
- Splunk App for Websphere Application Server

Further reading for getting and installing apps

Go to Splunkbase to browse through the large set of apps available for download. Check Splunkbase frequently, because new apps get added all the time.

For more information on apps, see What are apps and add-ons? in the *Admin Manual*. In particular, Where to get more apps and add-ons tells you how to download and install apps:.

For information on how to create your own apps, see the *Developing Views and Apps for Splunk Web* manual.

Configure your inputs

To add a new type of data to Splunk Enterprise, tell it a few things about the data by configuring a data input. There are a number of ways to configure data inputs:

- **Apps.** Splunk has a variety of **apps** that offer preconfigured inputs for various data types. For more information, see ["Use apps to get data in"](#).
- **Splunk Web.** You can configure most inputs using the **Splunk Web** data input pages. You can access the **Add Data** landing page from Splunk Home. In addition, when you upload or monitor a file, Splunk Enterprise lets you [preview and make adjustments to](#) how it plans to index the file before doing the indexing.
- **The Splunk Command Line Interface (CLI).** You can use the CLI to configure most types of inputs.
- **The `inputs.conf` configuration file.** When you specify your inputs with Splunk Web or the CLI, Splunk Enterprise saves them in a **configuration file**, `inputs.conf`. You can edit that file directly. Some advanced data input needs might require you to edit it.

In addition, if you configure **forwarders** to send data from outlying machines to a central **indexer**, you can specify some inputs at installation time. See ["Use forwarders to get data in"](#).

Use Splunk Web

You can add data inputs from Splunk Home or the **Settings > Data Inputs** menu:

- From Splunk Home, select **Add Data**. The **Add Data** page appears, with links to recipes for a variety of data input types.
- From anywhere in Splunk Web, select **Settings** in the system bar. Then select **Data inputs** from the **Data** section of the **Settings** pop-up menu. The "Data Inputs" page appears. You can view and manage your existing inputs, as well as add new ones.

The **Add Data** page has options to get data in: [Upload](#), [Monitor](#), and [Forward](#). Click an icon to go to a page to define the data you want to upload, monitor, or forward.

For more help on how to use the "Add Data" page, see [How do you want to add data?](#) in this manual.

How app context determines where Splunk Enterprise writes configuration files

When you add an input through Splunk Web, Splunk Enterprise adds that input to a copy of `inputs.conf`. The app context, that is, the Splunk app you are currently in when you configure the input, determines where Splunk Enterprise writes the `inputs.conf` file.

For example, if you navigated to the Settings page directly from the Search page and then added an input, Splunk Enterprise adds the input to

`$SPLUNK_HOME/etc/apps/search/local/inputs.conf`.

When you add inputs, confirm that you are in the app context that you want to be in. For background on how configuration files work, read [About configuration files](#) in the *Admin* manual.

Use the CLI

You can use the Splunk CLI to configure most inputs. From a shell or command prompt, navigate to the `$SPLUNK_HOME/bin/` directory and use the `./splunk` command. For example, the following command adds `/var/log/` as a data input:

```
splunk add monitor /var/log/
```

The Splunk CLI has built-in help. For the list of CLI commands, type:

```
./splunk help commands
```

Individual commands have their own help pages. To see them, type:

```
./splunk help <command>
```

For information on how to use the CLI to configure a specific input, see the topic for that input. For example, to learn how to use the CLI to configure network inputs, see ["Add a network input using the CLI"](#). For more information on the CLI, including how to get command line help, see "About the CLI" in the *Admin* manual.

Edit inputs.conf

To configure an input with `inputs.conf`, use a text editor. Add a **stanza** for each input. You can add the stanza to the `inputs.conf` file in

`$SPLUNK_HOME/etc/system/local/`, or in your custom application directory (in `$SPLUNK_HOME/etc/apps/<app name>/local`).

You configure the data input by adding attribute/value pairs to its stanza. You can set multiple attributes in an input stanza. If you do not specify a value for an attribute, Splunk Enterprise uses the default value for the attribute. Default values for all `inputs.conf` attributes are in

`$SPLUNK_HOME/etc/system/default/inputs.conf`.

If you have not worked with configuration files, see "About configuration files." before starting to add inputs.

Example inputs.conf stanza

The following example configuration directs Splunk Enterprise to listen on TCP port 9995 for raw data from any remote host. Splunk Enterprise uses the DNS name of the remote host to set the host of the data. It assigns the source type "log4j" and the source "tcp:9995" to the data.

```
[tcp://:9995]
connection_host = dns
sourcetype = log4j
source = tcp:9995
```

For information on how to configure a specific input, see the topic in this manual for that input. For example, to learn how to configure file inputs, see [Monitor files and directories with inputs.conf](#).

The topic for each data input describes the main attributes available for that input. See the `inputs.conf` spec file for the complete list of available attributes, including descriptions of the attributes and several examples.

About source types

As part of the input process, Splunk Enterprise assigns a **source type** to the data. The source type identifies the data format. Splunk Enterprise uses the source type during indexing to format events correctly. It usually assigns the correct source type. For instance, syslog data gets a source type of "syslog". You can substitute either one of the predefined source types or one that you create. You set the source type at the time you configure the input. See "[Why source types matter](#)". The topic "[Override automatic source type assignment](#)" describes source type assignment options.

To learn how to set the source type on a per-event basis, see "[Advanced source type overrides](#)". </!-->

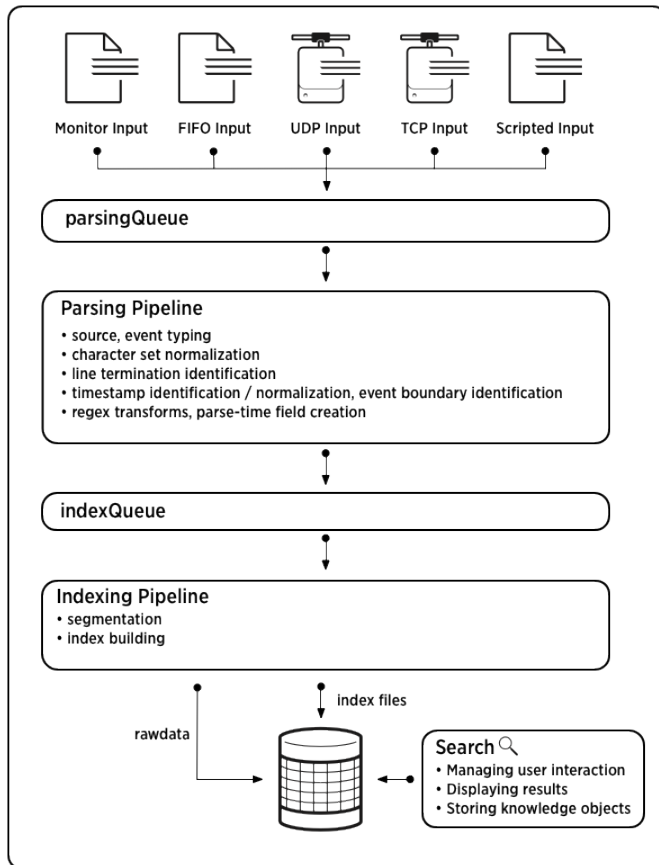
How Splunk Enterprise handles your data

Splunk Enterprise consumes any sort of data and **indexes** it, transforming it into searchable knowledge in the form of **events**. The data pipeline shows the main processes that act on the data during indexing. These processes constitute **event processing**. After the data is processed into events, you can associate the events with **knowledge objects** to enhance their usefulness.

The data pipeline

After a chunk of data enters Splunk Enterprise, it moves through the data pipeline. For a description of the data pipeline, see How data moves through Splunk Enterprise in the *Distributed Deployment Manual*.

This diagram shows the main steps in the data pipeline.



Event processing

Event processing occurs in two stages, parsing and indexing. All data that comes into Splunk Enterprise enters through the **parsing pipeline** as large chunks. During parsing, Splunk Enterprise breaks these chunks into events. It then hands off the events to the **indexing pipeline**, where final processing occurs.

During both parsing and indexing, Splunk Enterprise transforms the data. You can configure most of these processes to adapt them to your needs.

In the parsing pipeline, Splunk Enterprise performs a number of actions, including:

- Extracting a set of **default fields** for each event, including `host`, `source`, and `sourcetype`.
- Configuring **character set encoding**.
- Identifying line termination using **line breaking** rules. You can also modify line termination settings interactively, using the "**Set Sourcetype**" page in

Splunk Web.

- Identifying or creating [timestamps](#). At the same time that it processes timestamps, Splunk Enterprise identifies event boundaries. You can modify timestamp settings interactively, using the "[Set sourcetype](#)" page.
- Anonymizing data, based on your configuration. You can set up Splunk Enterprise to [mask sensitive event data](#) (such as credit card or social security numbers) at this stage.
- [Applying custom metadata](#) to incoming events, based on your configuration.

In the indexing pipeline, Splunk Enterprise performs additional processing, including:

- Breaking all events into [segments](#) that can then be searched. You can determine the level of segmentation, which affects indexing and searching speed, search capability, and efficiency of disk compression.
- Building the index data structures.
- Writing the raw data and index files to disk, where post-indexing compression occurs.

The distinction between parsing and indexing pipelines matters mainly for forwarders. Heavy forwarders can parse data locally and then forward the parsed data on to receiving indexers, where the final indexing occurs. Universal forwarders offer minimal parsing in specific cases such as handling structured data files. Additional parsing occurs on the receiving indexer.

For information about events and what happens to them during the indexing process, see [Overview of event processing](#) in this manual.

Enhance and refine events

After the data has been transformed into events, you can make the events more useful by associating them with knowledge objects, such as event types, field extractions, and reports. For information about managing Splunk knowledge, see the *Knowledge Manager* manual, starting with "What is Splunk knowledge?".

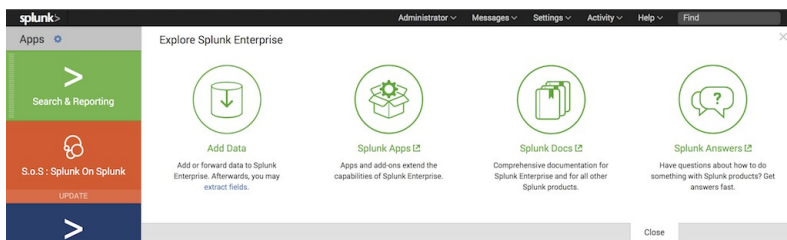
How to get data into Splunk Enterprise

How do you want to add data?

This topic discusses how to add data to Splunk Enterprise with Splunk Web. It describes the ways you can use Splunk Web to get data in

The Add Data page

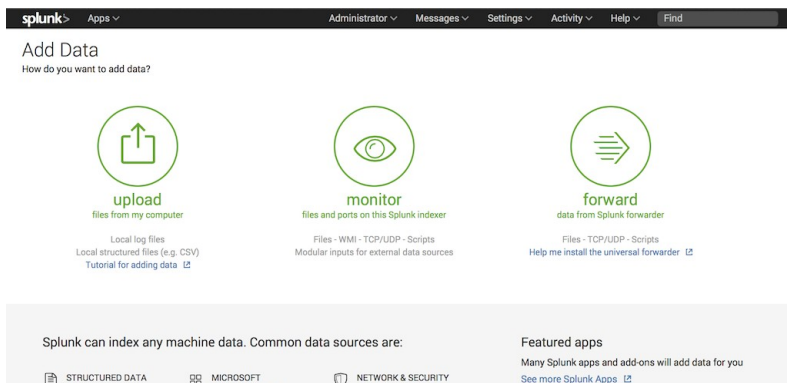
After you log into Splunk Enterprise, the Home page appears:



To add data, click the green **Add Data** button (to the right of the list of apps.) The Add Data page appears.

Note: The Add Data page does not appear if:

- This instance operates as part of a **search head cluster**. See "About search head clustering" in the *Distributed Search* manual.
- This is a Splunk Cloud instance.



There are three options for getting data into your Splunk Enterprise instance with Splunk Web: **Upload**, **Monitor**, and **Forward**.

Upload

The Upload option lets you upload a file or archive of files into Splunk Enterprise for indexing. When you click Upload, Splunk Enterprise goes to a page that starts the upload process. See [Upload data](#).

Monitor

The Monitor option lets you monitor one or more files, directories, network streams, scripts, Event Logs (on Windows hosts only), performance metrics, or any other type of machine data that the Splunk Enterprise instance has access to. When you click Monitor, Splunk Enterprise loads a page that starts the monitoring process. See [Monitor data](#).

Forward

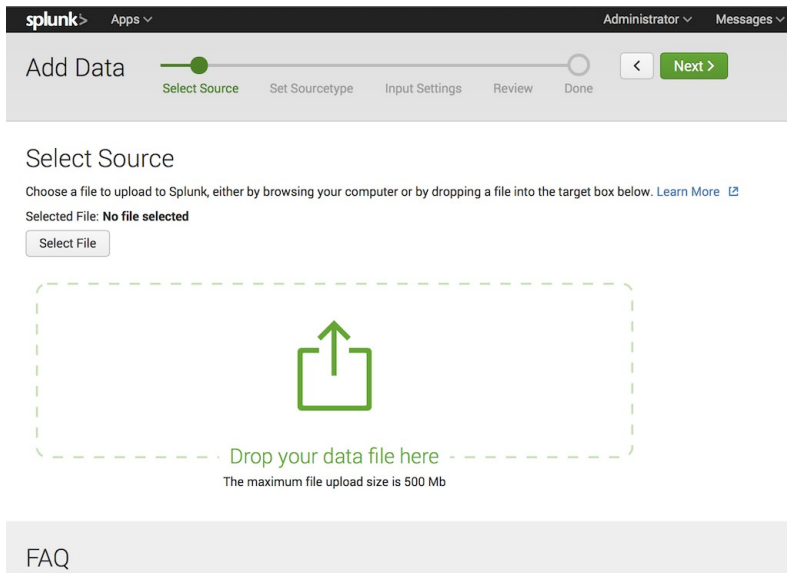
The Forward option lets you receive data from **forwarders** into this Splunk Enterprise instance. When you click on the "Forward" button, Splunk Enterprise takes you to a page that starts the data collection process from forwarders. See [Forward data](#).

Note: The Forward option requires additional configuration. Use it in a single-instance Splunk Enterprise environment only.

Upload data

This topic explains the page that Splunk Enterprise loads when you select the "Upload" button on the "Add data" page.

The "Upload" page



This page lets you upload data through one of the following methods:

- Drag the file you want Splunk Enterprise to index directly from your desktop and drop it into the "Drop your data file here" area on the page.

or

- In the upper left of the screen, click **Select File** and select the file that you want to index.

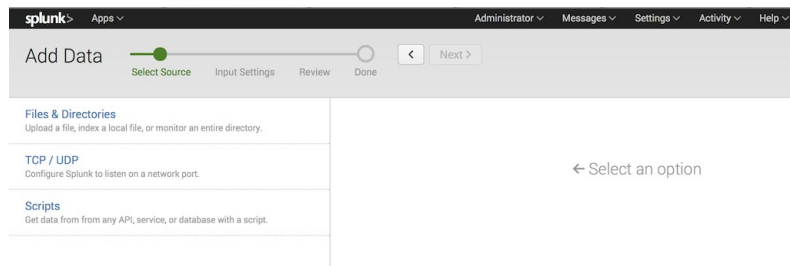
Splunk Enterprise then loads the file and processes it, depending on what type of file it is. After it has completed loading, you can then click the green **Next** button on the upper right to proceed to the [next step](#) in the "Add data" process.

Note: Windows Event Log (.evt) and Windows Event Log XML (.evtx) files that have been exported from another host do not work with the upload feature. This is because those files contain information that is specific to the host that generated them. Other hosts won't be able to process the files in their unaltered form. See [Index exported event log \(.evt or .evtx\) files](#) in this manual for more information about the constraints for working with these kinds of files.

Monitor data

This topic explains the page that Splunk Enterprise loads when you select the **Monitor** button on the Add data page.

The Monitor page



When you access the "Monitor" page, choose the type of data that you want Splunk Enterprise to monitor. Splunk Enterprise lists default inputs first, followed by forwarded inputs, and then any modular inputs you have installed on the instance.

Splunk Enterprise shows only sources that it can monitor. See the [list of data sources](#) for more information. If you do not see the data source that you want to monitor, consider that:

Add a data input

- Some data sources are available only on certain operating systems. For example, Windows data sources only available on hosts that run Windows.
- The user you logged into Splunk Enterprise with might not have permissions to add data or see the data source you want to add.

1. Select a source from the left pane by clicking it once. The page updates based on the source you selected. For example, if you select "Files & Directories", the page updates with a field to enter a file or directory name and specify how Splunk Enterprise should monitor the file or directory.

2. Follow the on-screen prompts to complete the selection of the source object that you want Splunk Enterprise to monitor.

3. Click **Next** to proceed to the [next step](#) in the Add data process.

Forward data

This topic explains the Select Forwarders page that Splunk Enterprise loads when you click the **Forward** button on the Add data page.

Use this page only if you have a single instance of Splunk Enterprise acting as an indexer and deployment server. If you have multiple hosts that perform indexing, see [About deployment server and forwarder management](#) in the *Updating Splunk Enterprise Instances* manual.

Prerequisites

The "Forward Data" page requires that you configure at least one forwarder to connect to the instance as a deployment client. This lets you configure data inputs on the forwarder and have the forwarder send the data it collects to this instance.

If you have not configured a forwarder, the page notifies you that no deployment clients have been found.

To configure a light or heavy forwarder as a deployment client, see [Configure deployment clients](#) in the *Updating Splunk Enterprise Instances* manual. To configure a universal forwarder as a deployment client, see [Configure the universal forwarder as a deployment client](#) in the *Universal Forwarder* manual. On Windows hosts, you can configure the forwarder as a deployment client during installation.

The Select Forwarders page

When you select "Forward Data" from the "Add Data" page, the following page appears:

The screenshot shows the 'Select Forwarders' page in the Splunk web interface. At the top, there's a navigation bar with 'splunk>' and 'Apps' dropdown. Below it, a breadcrumb trail shows 'Add Data' followed by a progress bar with steps: 'Select Forwarders' (active), 'Select Source', 'Input Settings', 'Review', and 'Done'. A 'Next >' button is visible. The main heading is 'Select Forwarders' with a subtext: 'Create or select a server class for data inputs. Use this page only in a single-instance Splunk environment.' Below this, a note says: 'To enable forwarding of data from deployment clients to this instance, set the output configurations on your forwarders. [Learn More](#)'. The 'Select Server Class' section has two tabs: 'New' and 'Existing'. Under 'New', there are two columns: 'Available host(s)' and 'Selected host(s)'. The 'Available host(s)' column has a list with 'WINDOWS Docs-W2R8-AD'. The 'Selected host(s)' column is empty. At the bottom, there's a text input field for 'New Server Class Name'. A footer section contains an 'FAQ' link and a question: '> How do I create sourcetypes for data originating from Forwarders?'.

You can define **server classes** and add forwarders to those classes. Server classes are logical groupings of hosts based on things such as architecture or host name.

This page only displays forwarders that you configured to forward data and act as deployment clients to this instance. If you have not configured any forwarders, the page warns you of this.

1. In **Select Server Class**, click one of the options:

- **New** to create a new server class, or if an existing server class does not match the group of forwarders that you want to configure an input for.
- **Existing** to use an existing server class.

2. In the **Available host(s)** pane, choose the forwarder(s) that you want this instance to receive data from. The forwarders move from the **Available host(s)** pane to the **Selected host(s)** pane.

Note: A server class must contain hosts of a certain platform. You cannot, for example, put Windows and *nix hosts in the same server class.

3. (Optional) You can add all of the hosts by clicking the **add all** link, or remove all hosts by selecting the **remove all** link.

4. If you chose **New** in "Select server class", enter a unique name for the server class that you will remember. Otherwise, select the server class you want from the drop-down list.

5. Click **Next**. The "Select Source" page shows source types that are valid for the forwarders that you selected.

6. Select the data sources that you want the forwarders to send data to this instance.

7. Click **Next** to proceed to the [Set Sourcetype](#) page.

The Set Sourcetype page

The "Set Sourcetype" page lets you improve **event processing** by previewing how Splunk Enterprise indexes your data. Use this page to confirm that Splunk Enterprise indexes your data as you want it to appear.

Preview data prior to indexing

The Set Sourcetype page appears after you use the ["Upload"](#) or ["Monitor"](#) pages to specify a single file as a source of data.

On the "Set Sourcetypes" page, you can make adjustments to how Splunk Enterprise indexes your data. You can adjust and improve the indexing process interactively so that when Splunk Enterprise performs the actual indexing, your **event data** ends up stored in the format you want.

Assign the correct source type to your data

The "Set sourcetype" page helps you apply the correct **source type** to your incoming data. The source type is one of the **default fields** that Splunk Enterprise assigns to all incoming data, and determines how Splunk Enterprise formats the data during indexing. By assigning the correct source type to your data, the indexed version of the data (the **event data**) will look the way you want it to, with correct **timestamps** and **event** breaks.

Splunk Enterprise comes with a large number of predefined source types and attempts to assign the correct source type to your data based on its format. In some cases, you might need to manually select a different predefined source type to the data. In other cases, you might need to create a new source type with customized event processing settings.

The page displays how Splunk Enterprise will index the data based on the application of a predefined source type. You can modify the settings interactively and save those modifications as a new source type.

Use the Set Sourcetype page to:

- See what your data will look like without any changes, using the default event processing configuration that Splunk Enterprise automatically applies.
- Apply a different source type to see whether or not that offers results more to your liking.
- Modify settings for timestamps and event breaks to improve the quality of the indexed data and save the modifications as a new source type.
- Create a new source type from scratch.

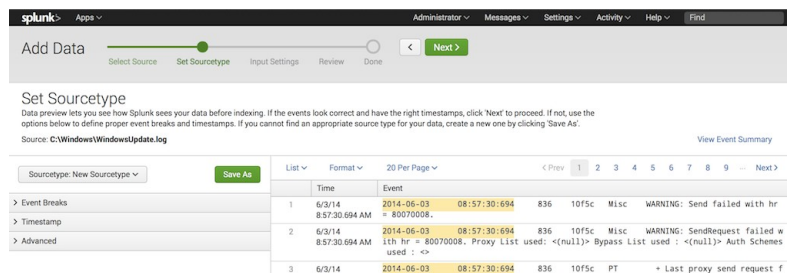
The page saves any new source types to a `props.conf` file that you can later distribute across the indexers in your deployment, so that the source types are available globally. See ["Distribute source type configurations"](#).

For information on source types, see "[Why source types matter](#)" in this manual.

Use the "Set Sourcetypes" page

When the Set Sourcetype page loads, Splunk Enterprise chooses a source type based on the data you specified. You can accept that recommendation or change it.

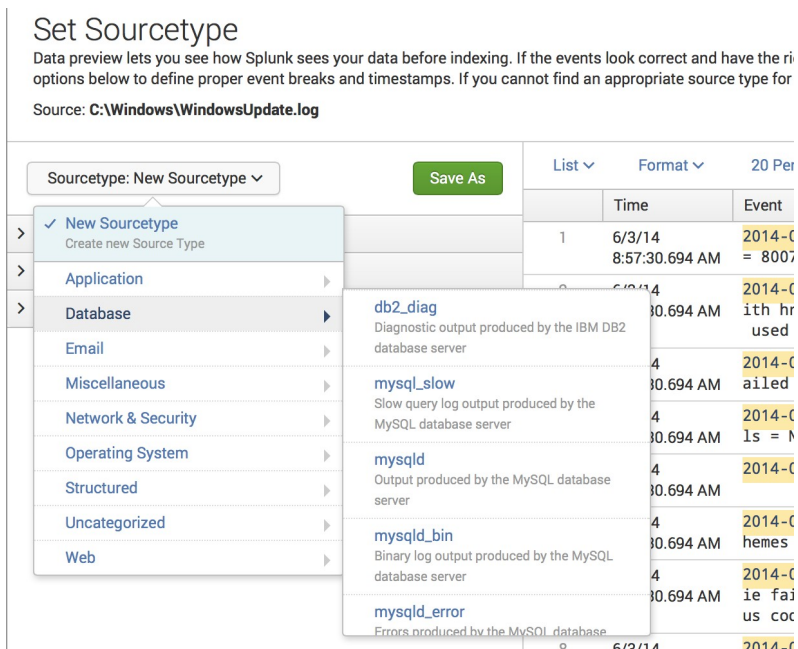
Here is an example of the "Set Sourcetype" page:



1. Look at the preview pane on the right side of the page to see how Splunk Enterprise will index the data. Review event breaks and time stamps.
2. (Optional) View the event summary by clicking on the **View event summary** link on the right. Splunk Enterprise displays the event summary in a new window. See "[View event summary](#)".
3. If the data appears the way that you want, then proceed to Step 5. Otherwise, choose from one of the following options:
 - ◇ **Choose an existing source type** to change the data formatting. See "[Choose an existing source type](#)."
 - ◇ **Adjust time stamps, delimiters, and line breaking manually**, then save the changes as a new source type. See [Adjust time stamps and event line breaks](#)."
4. After making the changes, return to Step 1 to preview the data again.
5. After you are satisfied with the results, click "Next" to proceed to the [Input Settings](#) page.

Choose an existing source type

If the data does not appear the way that you want, see whether or not an existing source type fixes the problem.



1. Click the **Sourcetype: <sourcetype>** button to see a list of source type categories. Under each category is a list of source types within that category.

Note: If Splunk Enterprise can detect a source type, it displays that source type in the "Sourcetype <sourcetype>" button. If not, it displays "Sourcetype: System Defaults" instead.

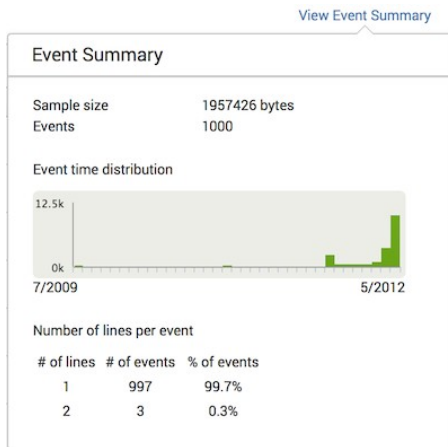
2. Hover over the category that best represents your data. As you do, the source types under that category appear in a pop-up menu to the right.

3. Select the source type that best represents your data. Splunk Enterprise updates the data preview pane to show how the data looks under the new source type.

Note: You might need to scroll to see all source types in a category.

4. Review your data again.

View event summary



You can see a summary of the events within the data sample that Splunk Enterprise collected by clicking the "View Event Summary" link on the right side of the page. This summary shows the following information:

- The size of the sample data, in bytes, that Splunk Enterprise gathered.
- The number of events that were present in the sample.
- The chart that represents the distribution of the events over time. Splunk Enterprise uses date stamps within the file to determine how to display this chart.
- A breakdown of the number of lines each event in the sample took up.

Adjust time stamps and event breaks

If you choose an existing source type without success, then you can manually adjust how Splunk Enterprise processes timestamps and event line breaks for the incoming data.

To manually adjust time stamp and event line breaking parameters, use the **Event Breaks**, **Timestamp**, **Delimited Settings**, and **Advanced** drop-down tabs on the left pane of the "Set Sourcetypes" page. The preview pane updates as you make changes to the settings.

For more information about how to adjust time stamps and event breaks, see ["Modify event processing"](#).

Note: Some tabs appear only if Splunk Enterprise detects that a file is a certain type, or if you select a specific source type for a file.

1. Click the **Event breaks** tab. The tab displays the **Break type** buttons, which control how Splunk Enterprise should line break the file into events.

- **Auto:** Splunk Enterprise detects event breaks based on the location of the time stamp.
- **By Line:** Splunk Enterprise breaks every line into a single event.
- **Regex?:** Splunk Enterprise uses the regular expression that you specify to determine line breaking.

Note: The Event breaks tab appears when Splunk Enterprise cannot determine how to line break the file, or if you select a source type that does not have line breaking defined.

2. Click the **Timestamps** tab. The tab expands to show options for extraction. Select from one of the following options:

- ◇ **Auto:** Splunk Enterprise handles time stamp extraction automatically, usually by looking in the file for timestamp events.
- ◇ **Current time:** Splunk Enterprise applies the current time to all events it finds.
- ◇ **Advanced:** Splunk Enterprise lets you specify the time zone, the timestamp format (in a specific format known as `strptime()`), and/or any fields that comprise the timestamp.

3. Click the **Delimited settings** tab to display delimiting options.

Delimited settings

Field delimiter: (comma),

Quote character: (double quote) "

File preamble:

A regular expression that instructs Splunk to ignore these preamble lines within the file.

Field names: Auto Line... Custom... Regex...

- ◇ **Field delimiter:** The delimiting character that Splunk Enterprise should use to define structured data files, such as comma-separated value (CSV) files.
- ◇ **Quote character:** The character that Splunk Enterprise uses to determine when something is in quotes.
- ◇ **File preamble:** A regular expression that tells Splunk Enterprise to ignore one or more preamble lines (lines that don't contain any actual data) in the structured data file.
- ◇ **Field Names:** How Splunk Enterprise should determine field names: Automatically, based on line number, based on a

comma-separated list, or through a regular expression.

After the results look the way you want, save your changes as a new **source type**, which you can then apply to the data as Splunk Enterprise indexes it.

Note: The Delimited settings tab appears only when Splunk Enterprise detects that you want to import a structured data file, or you select a source type for structured data (such as `csv`).

4. Click the **Advanced** tab to display fields that let you enter attribute/value pairs that get committed directly to the `props.conf` configuration file.

Caution: The "Advanced" tab requires advanced knowledge of Splunk Enterprise, and changes made here might negatively impact the indexing of your data. Consider consulting a member of Splunk Professional Services for help in configuring these options.

Make configuration changes in the Advanced tab

1. Click a field to make edits directly to `props.conf` entries that Splunk Enterprise generates automatically based on the choices you made previously.

2. Click on the X to the right of an attribute/value field pair to delete that pair.

3. Click **New setting** to create a new attribute/value field pair and specify a valid attribute and value for `props.conf`.

4. Click **Apply settings** to commit the changes to the `props.conf` file.

Prepare your data for previewing

This topic describes how to prepare your data to be viewed in the Splunk Enterprise "Set sourcetype" page.

The "[Set Sourcetype](#)" page works on single files only, and can only access files that are on the Splunk Enterprise instance or have been uploaded there. Although it does not directly process network data or directories of files, you can work around those limitations.

Preview network data

You can direct some sample network data into a file, which you can then either upload or add as a file monitoring input. Several external tools can do this. On *nix, the most popular tool is `netcat`.

For example, if you listen for network traffic on UDP port 514, you can use `netcat` to direct some of that network data into a file:

```
nc -lu 514 > sample_network_data
```

For best results, run the command inside a shell script that has logic to kill `netcat` after the file reaches a size of 2MB. By default, Splunk Enterprise reads only the first 2MB of data from a file when you preview it.

After you have created the "sample_network_data" file, you can add it as an input, preview the data, and assign any new source types to the file.

Preview directories of files

If all the files in a directory are similar in content, then you can preview a single file and be confident that the results will be valid for all files in the directory. However, if you have directories with files of heterogeneous data, preview a set of files that represents the full range of data in the directory. Preview each type of file separately, because specifying any wildcard causes Splunk Enterprise to disable the "Set Sourcetype" page.)

File size limit

Splunk Enterprise displays the first 2MB of data from a file in the "Set Sourcetypes" page. In most cases, this amount provides a sufficient sampling of your data. To sample a larger quantity of data, change the `max_preview_bytes` attribute in `limits.conf`. Alternatively, you can edit the file to reduce large amounts of similar data, so that the remaining 2MB of data contains a representation of all the types of data in the original file.

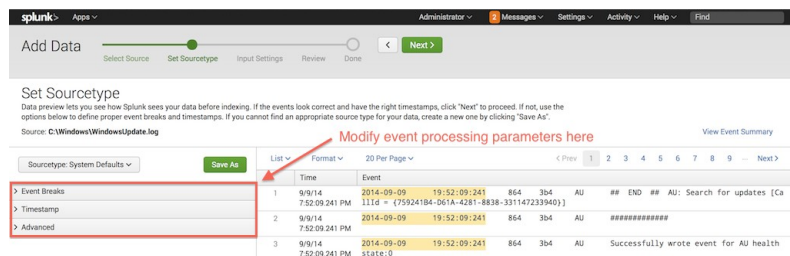
Modify event processing

You can change the event processing settings and save the improved settings as a new source type.

1. View the event data, as described in the "Set Sourcetype" page.
2. Modify the event processing settings.
3. Review the effect of your changes and iterate until you are satisfied.
4. Save the modified settings as a new source type.
5. Apply the new source type to any of your inputs.

Modify the event processing settings

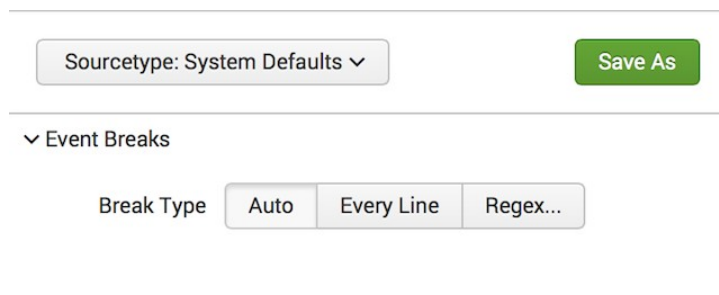
To create the new source type, use the event-breaking and time stamp parameters, then save the source type.



On the left side of the "Set Sourcetypes" page, there are collapsible tabs and links for the three types of adjustments that you can perform:

- **Event Breaks.** Adjust the way that Splunk Enterprise breaks the data into events.
- **Timestamps.** Adjust the way Splunk Enterprise determines event timestamps.
- **Advanced mode.** Edit `props.conf` directly.

Event breaks



To modify event break parameters, click on the **Event Breaks** bar to expand it. The bar opens to display the following buttons:

- **Auto.** Splunk Enterprise performs event breaking based on where it finds timestamps in the data.
- **Every line.** Splunk Enterprise considers every line a single event.
- **Regex...** Click this button to specify a regular expression that Splunk Enterprise uses to break data into events.

See "[Configure event linebreaking](#)".

For a primer on regular expression syntax and usage, see [Regular-Expressions.info](#). You can test your regular expression by using it in a search with the rex search command. Splunk Enterprise also maintains a list of useful third-party tools for writing and testing regular expressions.

Timestamps

Timestamp

Extraction

Auto

Current time

Advanced...

Time zone

Auto

Timestamp format

A string in `strptime()` format that helps Splunk recognize timestamps. [Learn More](#)

Timestamp prefix

Timestamp is always prefaced by a regex pattern eg:
`\d+abc123\d[2,4]`

Lookahead

128

Timestamp never extends more than this number of characters into the event, or past the Regex if specified above.

Timestamp fields

Specify all the fields which constitute the timestamp. ex:
`field1,field2,...,fieldn`

To modify time stamp recognition parameters click the **Timestamps** tab to expand it. The tab opens to reveal these options:

For **Extraction**, you can choose one of these options:

- **Auto.** Splunk Enterprise automatically locates the timestamp.
- **Current Time.** Splunk Enterprise uses the current time on the local instance.
- **Advanced.** Splunk provides additional advanced parameters to adjust.

The "Advanced" parameters are:

- **Timezone.** The time zone that you want to use for the events.
- **Timestamp format.** A string that represents the [time stamp format](#) you expect Splunk Enterprise to use when searching for time stamps within the data.
- **Timestamp prefix.** A regular expression that represents the characters that appear before a time stamp.
- **Lookahead.** The number of characters that Splunk Enterprise should look into the event (or, for the regular expression that you specified in "Timestamp prefix") for the time stamp.

Note: If you specify a timestamp format in the "Timestamp format" field and the timestamp is not located at the very start of each event, you must also specify a prefix in the **Timestamp prefix** field. Otherwise, Splunk Enterprise can not process the formatting instructions, and every event will contain a warning about the inability to use `strptime`. (It's possible that you still end up with a valid timestamp, based on how Splunk Enterprise attempts to recover from the problem.)

For information on configuring timestamps, see ["Configure timestamps"](#).

Advanced




To modify advanced parameters, click the **Advanced** tab to expand it. The tab shows options that let you specify source type properties by editing the underlying `props.conf` file.

You can add or change source type properties by specifying attribute/value pairs. See `props.conf` for details on how to set these properties.

The "Advanced" box shows the current, complete set of properties for the selected source type:

- Settings generated by changes made in the **Event Breaks** or **Timestamps** tabs (after you click **Apply**).
- Pre-existing settings for a source type that was either auto-detected or manually selected when you first previewed the file.
- Settings you apply from the **Additional settings** text box (after you click **Apply settings**).

Advanced

Name	Value	
SHOULD_LINEMERGE	true	
NO_BINARY_CHECK	true	
disabled	false	

New setting

Copy to clipboard

Apply settings

For information on how to set source type properties, see "props.conf" in the Configuration file reference. See also "[How Splunk Enterprise extracts timestamps](#)" and "[event linebreaking](#)."

How Splunk Enterprise combines settings

The settings changes you make in **Advanced mode** take precedence. For example, if you alter a timestamp setting using the **Timestamps** tab and also make a conflicting timestamp change in **Advanced mode**, the **Advanced mode** change wins.

Starting with highest precedence, here is how Splunk Enterprise combines any adjustments with the underlying default settings:

- Advanced mode changes
- Event Breaks/Timestamps changes
- Settings for the underlying source type, if any
- Default system settings for all source types

Also, if you return to the Event Breaks or Timestamps tabs after making changes in **Advanced mode**, the changes will not be visible from those tabs.

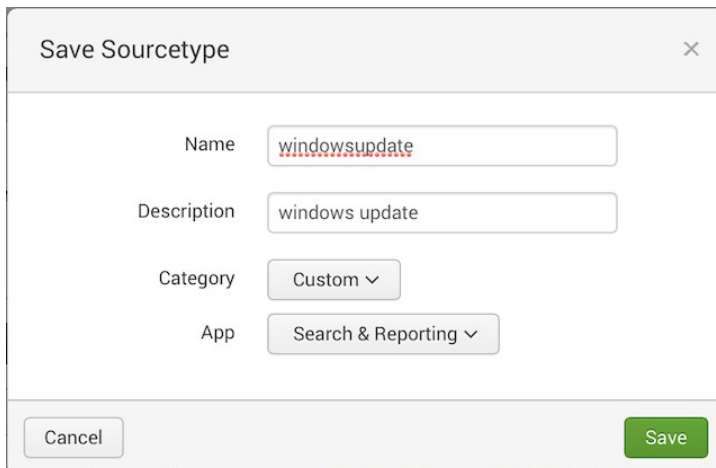
Review your changes

When you are ready to view the effect of your changes, select **Apply settings**. Splunk Web refreshes the screen, so you can review the effect of your changes on the data.

To make further changes using any of the three adjustment methods available again, select **Apply changes** to view the effect of the changes on your data.

Save modifications as a new source type

To save the changes as a new source type, click "Save As" next to the "Sourcetype" button. Splunk Web displays a dialog box where you can name your new source type, choose the category in which it should be shown in the "Sourcetype" button dialog, and the application context it should use.



The screenshot shows a 'Save Sourcetype' dialog box. It has a title bar with the text 'Save Sourcetype' and a close button (X). The main area contains four input fields: 'Name' with the value 'windowsupdate', 'Description' with the value 'windows update', 'Category' with a dropdown menu showing 'Custom', and 'App' with a dropdown menu showing 'Search & Reporting'. At the bottom, there are 'Cancel' and 'Save' buttons.

1. Enter the **Name** of the new source type.
2. Enter the **Description** of the source type.
3. Select the **Category** in which the source type should appear when you click "Sourcetype".
4. Select the **App** for which the new source type should be used.
5. Click **Save** to save the source type and return to the Set Sourcetypes page.

Next steps

You have several options after you save the source type:

1. (Optional) Click **Next** to apply the source type to your data and proceed to the **Input settings** page.
2. (Optional) Click the "<" button to go back and choose a new file to upload or monitor.
3. (Optional) Click the **Add data** text to return to the beginning of the Add Data wizard.

Modify input settings

This topic discusses the "Input Settings" page that appears after you configure the data source in the Set Sourcetypes page.

After you select the source (or set your source type when uploading or monitoring a single file), the following page appears:

The screenshot shows the 'Input Settings' page in the Splunk interface. At the top, there's a navigation bar with 'Add Data' and a progress bar indicating the current step is 'Input Settings'. Below this, the page is titled 'Input Settings' with a subtitle 'Optionally set additional input parameters for this data input as follows:'. The page is divided into four main sections, each with a title, a brief description, and configuration options:

- Source type**: Describes the source type as a default field for indexing. Options include 'Select' and 'New' buttons, a 'Source Type' text input, a 'Source Type Category' dropdown menu, and a 'Source Type Description' text input.
- App context**: Describes application contexts as folders within a Splunk instance. Options include an 'App Context' dropdown menu with 'Search & Reporting' selected.
- Host**: Describes the host value as the name of the machine where the event originates. Options include a 'Method' dropdown menu with 'IP', 'DNS', and 'Custom' options.
- Index**: Describes the index where data is stored. This section is partially visible at the bottom of the screenshot.

You can specify additional parameters for your data input, such as its source type, its application context, its host value, and the index where data from the input should be stored.

The following input settings are available:

Configure source type

You can specify the source type that Splunk Enterprise applies to your data with the "Source type" setting. This setting appears when:

- You specify a directory as a data source.
- You specify a network input as a data source.
- You specify a data source that has been forwarded from another Splunk Enterprise instance.

If your data source does not meet these criteria, then the "Source type" setting does not appear.

To specify a source type, click one of the buttons:

- **Select:** Applies the source type you specify to the data. When you click "Select", a drop-down appears.
- **New:** Adds a new source type to Splunk Enterprise. When you click "New", two text fields and a drop-down appear.

To choose an existing source type:

1. From the "Select Source Type" drop-down, choose the category that best represents the source type you want.
2. Choose the source type from the pop-up list that appears.

To add a new source type:

1. In the "Source Type" text field, enter the name of the new source type.
2. Choose a category for the source type in the "Source Type Category" drop down.
3. In the "Source Type Description" field, enter the description for the source type.

Configure app context

The **Application Context** setting determines the context in which the input should collect data. Application contexts improve manageability of input and source type definitions. Splunk Enterprise loads all app contexts based on precedence rules. See Configuration file precedence in the *Admin* manual.

Select the application context you want this input to operate within by clicking the drop-down list and selecting the application context you want.

Configure host value

Splunk Enterprise tags each event that it indexes with a host value. You can configure what Splunk Enterprise tags events with by specifying how it should do so.

- **IP:** Uses the IP address of the host from which the event originates.
- **DNS:** Resolves the host value through Domain Name Services (DNS). Splunk Enterprise tags the events with the host name that it gets through name resolution.
- **Custom:** Uses the host value you assign in the "Host field value" text box that appears when you select this option.

Index

The "Index" setting determines which index that it should store the events for this input. To use the default index, leave the drop-down list set to "Default". To choose another index, click the drop-down list and select the index you want the data to go to by clicking the selection in the list. If the index you want to send the data to is not in the list, and you have permissions to create indexes, you can create a new index by clicking the **Create a new index** button.

After you make your selections, click **Next** to proceed to the final step of the "Add Data" process.

Distribute source type configurations

You can use either the "[Set source type](#)" or [source type management](#) pages in Splunk Web to create new source types, which you can then assign to inputs from specific files or directories, or for network inputs. Either of these pages saves a new source type to a `props.conf` configuration file on the local Splunk Enterprise instance. You can then distribute this file to other Splunk Enterprise instances so that they recognize the new source type.

You can use a new source type in a distributed environment where you have **forwarders** consuming data and then sending the data to indexers.

To install this new source type, follow these high-level steps:

1. Distribute the `props.conf` file that contains the source type definition to the `$SPLUNK_HOME/etc/system/local` directory on indexers that you want to index data with the source type you created.
2. Use the new source type when you define an input on forwarders that send data to those indexers.

When a forwarder sends data tagged with the new source type to an indexer, the indexer can correctly process it into events.

Data preview props.conf file

When you create a source type in the "Set Sourcetype" page, Splunk Enterprise saves the source type definition as a stanza in a `props.conf` file in the app that you selected when you saved the source type. If you later create additional source types, Splunk Enterprise saves the additional source types to the same `props.conf` file.

For example, if you selected the "Search and Reporting" app, the file resides in `$SPLUNK_HOME/etc/apps/search/local/props.conf`. The only exception is the "System" app: If you choose that app when saving the source type, the file resides in `$SPLUNK_HOME/etc/system/local..`

Note: A Splunk Enterprise instance might have multiple versions of some configuration files, in several directories. At run-time, Splunk Enterprise combines the contents of configuration files according to a set of precedence rules. For background on how configuration files work, see [About configuration files and Configuration file precedence](#).

Distribute props.conf to other indexers

After you create source types, you can distribute `props.conf` to another Splunk Enterprise instance. That instance can then index any incoming data that you tag with the new source type.

Best practice is to place the configuration file in its own app directory on the target Splunk Enterprise instance; for example,
`$SPLUNK_HOME/etc/apps/custom_sourcetype/local/.`

To distribute configuration files to other Splunk instances, you can use the Splunk Enterprise **deployment server** or another distribution tool. See the

Updating Splunk Instances manual.

Note: Splunk Enterprise uses the source type definitions in `props.conf` to parse incoming data into events. For this reason, you can only distribute the file to a Splunk Enterprise instance that performs **parsing** (either an indexer or a **heavy forwarder**.)

Specify the new source type in forwarder inputs

Forwarders (with the exception of the heavy forwarder) do not have Splunk Web. This means that you must configure their inputs through the CLI or the `inputs.conf` configuration file. When you specify an input in that file, you can also specify its source type. For information on `inputs.conf`, read the section on `inputs.conf` in the Configuration file reference.

To tag a forwarder input with a new source type, add the source type to the input stanza in `inputs.conf`. For example:

```
[tcp://:9995]
sourcetype = new_network_type
```

Confirm that all of the indexers that the forwarder sends data to have copies of the `props.conf` file that contains the source type definition for "new_network_type". When the forwarder sends data to the indexers, they can identify the new source type and correctly format the data.

Get data from files and directories

Monitor files and directories

Splunk Enterprise has three file input processors: **monitor**, `MonitorNoHandle`, and `upload`.

You can use **monitor** to add nearly all your data sources from files and directories. However, you might want to use `upload` to add one-time inputs, such as an archive of historical data.

On hosts that run Windows Vista or Windows Server 2008 and later, use `MonitorNoHandle` to monitor files which the system rotates automatically. `MonitorNoHandle` works only on Windows hosts.

Add inputs to `monitor` or `upload` using any of these methods:

- [Splunk Web](#)
- [The CLI](#)
- [inputs.conf](#)

You can add inputs to `MonitorNoHandle` using either the CLI or `inputs.conf`.

Use the "Set Sourcetype" page to see how Splunk Enterprise will index data from a file. See "[The "Set Sourcetype" page](#)" for details.

How the monitor processor works in Splunk Enterprise

Specify a path to a file or directory and the Splunk Enterprise monitor processor consumes any new data written to that file or directory. This is how you can monitor live application logs such as those coming from Web access logs, Java 2 Platform Enterprise Edition (J2EE) or .NET applications, and so on.

Splunk Enterprise monitors and indexes the file or directory as new data appears. You can also specify a mounted or shared directory, including network file systems, as long as Splunk Enterprise can read from the directory. If the specified directory contains subdirectories, Splunk Enterprise recursively examines them for new files, as long as the directories can be read.

You can include or exclude files or directories from being read by using **whitelists** and **blacklists**.

If you disable or delete a monitor input, Splunk Enterprise does not stop indexing the files that the input references. It only stops checking those files again. To stop all in-process data indexing, you must stop and restart the Splunk Enterprise server.

How Splunk Enterprise handles monitoring of files during restarts

When you restart Splunk Enterprise, it continues processing files where it left off. It first checks for the file or directory specified in a monitor configuration. If the file or directory is not present on start, Splunk Enterprise checks for it every 24 hours from the time of the last restart. Splunk Enterprise also scans subdirectories of monitored directories continuously.

Monitor inputs may overlap. So long as the stanza names are different, Splunk Enterprise treats them as independent stanzas and files matching the most specific stanza will be treated in accordance with its settings.

How Splunk Enterprise monitors archived files

If Splunk Enterprise encounters an archived file (such as a `.tar` or `.zip` file, it decompresses the file before indexing its contents. Splunk Enterprise can handle the following files:

- `.tar`
- `.gz`
- `.bz2`
- `.tar.gz` and `.tgz`
- `.tbz` and `.tbz2`
- `.zip`
- `.z`

If you add new data to an existing archive file, Splunk Enterprise reindexes the entire file, not just the new data. This can result in event duplication.

How Splunk Enterprise monitors files that the operating system rotates on a schedule

Splunk Enterprise detects log file rotation and does not process renamed files that it has already indexed (with the exception of `.tar` and `.gz` archives. See ["Log file rotation"](#) in this manual).

How Splunk Enterprise monitors nonwritable Windows files

Windows can prevent Splunk from reading open files. If you need to read files while they are being written to, you can use the `monitorNoHandle` input.

Restrictions to the kinds of files that Splunk Enterprise can monitor

Splunk Enterprise cannot monitor a file whose path exceeds 1024 characters.

Splunk Enterprise does not index files with a `.splunk` filename extension because it expects files with that extension to be metadata information files. If you need to index files with a `.splunk` extension, use the `add oneshot` CLI command.

Why use upload or batch?

To index a static file once, select **Upload** in Splunk Web. Splunk Enterprise will only monitor the file once.

You can also use the CLI `add oneshot` or `spool` commands for the same purpose. See "[Use the CLI](#)" for details.

Use the `batch` input type in `inputs.conf` to load files once and destructively. By default, the Splunk batch processor is located in `$SPLUNK_HOME/var/spool/splunk`. If you move a file into this directory, Splunk indexes it and then deletes it.

Note: For best practices on loading file archives, see "How to index different sized archives" on the Community Wiki.

Why use MonitorNoHandle?

This Windows-only input lets you read files on Windows systems as Windows writes to them. It does this by using a kernel-mode filter driver to capture raw data as it gets written to the file. Use this input stanza on files which get locked open for writing. You can use this input stanza on a file which the system locks open for writing, such as the Windows DNS server log file.

Note: `MonitorNoHandle` only works on Windows Vista or Windows Server 2008 and later operating systems. You can only monitor single files with `MonitorNoHandle`. You can not monitor directories. If a file you choose to monitor already exists, Splunk does not index its current contents, only new information that comes into the file as it gets written to.

Monitor files and directories with Splunk Web

You can use Splunk Web to add inputs from files and directories.

Go to the Add New page

You add an input from the Add Data page in Splunk Web.

You can get there by two routes:

- Splunk Home
- Splunk Settings

Splunk Settings:

1. Click **Settings** in the upper right corner of Splunk Web.
2. In the Data section of the Settings pop-up, click **Data Inputs**.
3. Click **Files & Directories**.
4. Click **New** to add an input.

Splunk Home:

1. Click **Add Data** in Splunk Home.
2. Click **Upload** to upload a file, **Monitor** to monitor a file, or **Forward** to forward a file.

Note: Forwarding a file requires additional setup. See the following topics in the *Forwarding Data* manual:

- "Universal forwarder deployment overview" if you work with universal forwarders.
- "Enable forwarding on a Splunk Enterprise instance" if you work with heavy and light forwarders.

Select the input source

1. To add a file or directory input, click **Files & Directories**.

2. In the **File or Directory** field, specify the full path to the file or directory.

To monitor a shared network drive, enter the following: <myhost>/<mypath> (or \\<myhost>\<mypath> on Windows). Make sure Splunk Enterprise has read access to the mounted drive, as well as to the files you want to monitor.

3. Choose how you want Splunk Enterprise to monitor the file:

- **Continuously Monitor.** Sets up an ongoing input. Splunk Enterprise monitors the file continuously for new data.
- **Index Once.** Copies a file on the server into Splunk Enterprise.

4. Click **Next**. If you specified a directory in the "File or Directory" field, Splunk Enterprise refreshes the screen to show fields for "whitelist" and "blacklist". These fields let you specify regular expressions that Splunk Enterprise then uses to match files for inclusion or exclusion. Otherwise, Splunk Enterprise proceeds to the "Set Sourcetype" page where you can preview how Splunk Enterprise proposes to index the events. See "[Whitelist or blacklist specific incoming data.](#)"

Preview your data and set its source type

When you add a new file input, Splunk Enterprise lets you set the **source type** of your data and preview how it will look once it has been indexed. This lets you ensure that the data has been formatted properly and make any necessary adjustments.

For information about this page, see "[The Set Sourcetype page.](#)"

If you skip previewing the data, the **Input Settings** page appears.

Note: You cannot preview directories or archived files.

Specify input settings

You can specify application context, default host value, and index in the **Input Settings** page. All parameters are optional.

1. Select the appropriate **Application context** for this input.
2. Set the **Host** name value.

Note: **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 for this input. Leave the value as "default", unless you have defined multiple indexes and want to use one of those instead.

4. Click **Review** to review all of the choices you have made.

Review your choices

After you specifying all input settings, review your selections. Splunk Enterprise lists all options you selected, including but not limited to 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 the gray < button to go back to the previous step in the wizard. Otherwise, click **Submit**. The "Success" page appears and Splunk Enterprise begins indexing the specified file or directory.

Monitor files and directories with the CLI

Monitor files and directories via the Splunk Enterprise Command Line Interface (CLI). To use the CLI, navigate to the `$SPLUNK_HOME/bin/` directory from a command prompt or shell, and use the `splunk` command in that directory.

The CLI has built-in help. Access the main CLI help by typing `splunk help`. Individual commands have their own help pages as well. Access that help by typing `splunk help <command>`.

CLI commands for input configuration

The following commands are available for input configuration using the CLI:

Command	Command syntax	Action
add monitor	<code>add monitor</code> <code>[-source] <source></code> <code>[-parameter value]</code> <code>...</code>	Monitor inputs from <source>.
edit monitor	<code>edit monitor</code> <code>[-source] <source></code> <code>[-parameter value]</code> <code>...</code>	Edit a previously added monitor input for <source>.

remove monitor	<code>remove monitor [-source] <source></code>	Remove a previously added monitor input for <source>.
list monitor	<code>list monitor</code>	List the currently configured monitor inputs.
add oneshot	<code>add oneshot <source> [-parameter value] ...</code>	Copy the file <source> directly into Splunk. This uploads the file once, but Splunk Enterprise does not continue to monitor it. You cannot use the <code>oneshot</code> command against a remote Splunk Enterprise instance. You also cannot use the command with either recursive folders or wildcards as a source. Specify the exact source path of the file you want to monitor.
spool	<code>spool <source></code>	Copy the file <source> into Splunk Enterprise using the sinkhole directory. Similar to add oneshot , except that the file spools from the sinkhole directory, rather than being added immediately.

CLI parameters for input configuration

Change the configuration of each data input type by setting additional parameters. Parameters are set via the syntax: `-parameter value`.

Note: You can only set one `-hostname`, `-hostregex` or `-hostsegmentnum` per command.

Parameter	Required?	Description
<source>	Yes	Path to the file or directory to monitor/upload for new input. Unlike the other parameters, the syntax for this parameter can be the value itself. It does not have to follow a parameter flag. You can use either of <code>./splunk monitor <source></code> or <code>./splunk monitor -source <source></code> .
<code>sourcetype</code>	No	Specify a <code>sourcetype</code> field value for events from the input source.
<code>index</code>	No	Specify the destination index for events from

		the input source.
<code>hostname</code> or <code>host</code>	No	Specify a host name to set as the host field value for events from the input source. These parameters are functionally equivalent.
<code>hostregex</code> or <code>host_regex</code>	No	Specify a regular expression to use to extract the host field value from the source key. These parameters are functionally equivalent.
<code>hostsegmentnum</code> or <code>host_segment</code>	No	An integer, which determines what "/" separated segment of the path to set as the host field value. If set to 3, for example, the third segment of the path is used. These parameters are functionally equivalent.
<code>rename-source</code>	No	Specify a value for the "source" field to be applied to data from this file.
<code>follow-only</code>	No	Set to true or false. Default is false. When set to true, Splunk Enterprise reads from the end of the source (like the "tail -f" Unix command). This parameter is not available for <code>add oneshot</code> .

Example 1: Monitor files in a directory

The following example shows how to monitor files in `/var/log/`.

Add `/var/log/` as a data input:

```
./splunk add monitor /var/log/
```

Example 2: Monitor windowsupdate.log

The following example shows how to monitor the Windows Update log file where Windows logs automatic updates, sending the data to an index called "newindex".

Add `C:\Windows\windowsupdate.log` as a data input:

```
./splunk add monitor c:\Windows\windowsupdate.log -index newindex
```

Example 3: Monitor Internet Information Server (IIS) logging

This example shows how to monitor the default location for Windows IIS logging.

Add C:\windows\system32\LogFiles\W3SVC as a data input:

```
./splunk add monitor c:\windows\system32\LogFiles\W3SVC
```

Example 4: Upload a file

This example shows how to upload a file into Splunk. Splunk Enterprise consumes the file only once. It does not monitor it continuously.

Upload /var/log/applog (C:\Program Files\AppLog\log.txt on Windows) directly into Splunk Enterprise with the `add oneshot` command:

Unix	Windows
<pre>./splunk add oneshot /var/log/applog</pre>	<pre>.\splunk add oneshot C:\Program Files\AppLog\log.txt</pre>

You can also upload a file through the sinkhole directory with the `<code>spool` command:

Unix	Windows
<pre>./splunk spool /var/log/applog</pre>	<pre>.\splunk spool C:\Program Files\AppLog\log.txt</pre>

The result is the same with either command.

The `add oneshot` and `spool` commands can not search recursively and do not support any wildcard usage. You must use an exact file path as the source.

Monitor files and directories with inputs.conf

To configure an input to Splunk Enterprise, add a **stanza** to `inputs.conf` in `$SPLUNK_HOME/etc/system/local/`, or your own custom application directory in `$SPLUNK_HOME/etc/apps/`.

You can set multiple attributes in an input stanza. If you do not specify a value for an attribute, Splunk Enterprise uses the default for that attribute, as defined in `$SPLUNK_HOME/etc/system/default/inputs.conf`.

For more information about configuration files, see "About configuration files".

Configuration settings

Use the following attributes in both `monitor` and `batch` input stanzas.

Attribute	Description	Default
<code>host = <string></code>	Sets the host key to a static initial value for this stanza. The input processor uses the key during parsing and indexing to set the host field and uses the field during searching. Splunk Enterprise prepends the <code><string></code> with <code>host::</code> .	the IP address or fully-qualified domain name of the host where the data originated.
<code>index = <string></code>	<p>Sets the index where events from this input will be stored. Splunk Enterprise prepends the <code><string></code> with <code>index::</code>.</p> <p>For more information about the index field, see "How indexing works" in the <i>Managing Indexers and Clusters</i> manual.</p>	<code>main</code> or whatever you set the default index to
<code>sourcetype = <string></code>	<p>Sets the sourcetype key/field for events from this input. Explicitly declares the source type for this data, as opposed to letting Splunk Enterprise determine it automatically. This is important both for searchability and for applying the relevant formatting for this type of data during parsing and indexing.</p> <p>Sets the sourcetype key initial value. Splunk Enterprise uses the key during parsing and indexing to set the source type field and uses the source type</p>	Splunk Enterprise picks a source type based on various aspects of the data. There is no default.

	<p>field during searching. Splunk Enterprise prepends the <code><string> with sourcetype::</code>.</p> <p>For more information about source types, see "Why source types matter", in this manual.</p>	
<pre>queue = parsingQueue indexQueue</pre>	<p>Specifies where the input processor should deposit the events that it reads. Set to "parsingQueue" to apply <code>props.conf</code> and other parsing rules to your data. Set to "indexQueue" to send your data directly into the index.</p>	parsingQueue
<pre>_TCP_ROUTING = <tcpout_group_name>, <tcpout_group_name>, ...</pre>	<p>Specifies a comma-separated list of tcpout group names. Use this attribute to selectively forward your data to specific indexers by specifying the tcpout groups that the forwarder should use when forwarding the data.</p> <p>Define the tcpout group names in <code>outputs.conf</code> in <code>[tcpout:<tcpout_group_name>]</code> stanzas.</p>	the groups present in 'defaultGroup' in <code>[tcpout]</code> stanza in <code>outputs.conf</code>
<pre>host_regex = <regular expression></pre>	<p>A regular expression that extracts host from the file name of each input. Specifically, Splunk Enterprise uses the first group of the regular expression as the host.</p>	the default "host =" attribute, if the regular expression fails to match
<pre>host_segment = <integer></pre>	<p>Sets the segment of the path as the host, using <code><integer></code> to determine the segment. For example, if <code>host_segment = 2</code>, <code>host</code> becomes the second segment of the path. Path segments are separated by the</p>	the default "host =" attribute, if the value is not an integer, or is less than 1

'/' character.

Monitor syntax and examples

Monitor input stanzas direct Splunk Enterprise to watch all files in the `<path>` (or `<path>` itself if it represents a single file). You must specify the input type and then the path, so put three slashes in the path if the path includes the root directory.

You can use wildcards for the path. See "[Specify input paths with wildcards](#)" in this manual.

```
[monitor://<path>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

The following are additional attributes you can use when defining monitor input stanzas:

Attribute	Description	Default
<code>source = <string></code>	Sets the source field for events from this input. Do not override unless absolutely necessary. Consider use of source types, tagging, and search wildcards instead. The input layer usually provides a more accurate string to aid in problem analysis and investigation by accurately recording the file from which the data was retrieved. Splunk Enterprise prepends the <code><string></code> with <code>source::.</code>	the input file path
<code>crcSalt = <string></code>	Forces Splunk Enterprise to consume files that have matching CRCs (cyclic redundancy checks). By default, the software only performs CRC checks against the first few lines of a file. This behavior prevents indexing of the same file twice, even though you might have renamed it, such as with rolling log files. However, because the CRC counts only the first few lines of the file, it is possible for legitimately different files to have matching CRCs, particularly if they have identical	N/A

	<p>headers.)</p> <p>If set, Splunk Enterprise adds <code>string</code> to the CRC. If set to <code><SOURCE></code>, Splunk Enterprise adds the full source path to the CRC. This ensures that each file being monitored has a unique CRC.</p> <p>Use caution with this attribute for rolling log files. It can lead to the log file being re-indexed after it has rolled.</p> <p>This setting is case sensitive.</p>	
<code>ignoreOlderThan = <time_window></code>	<p>Causes the input to stop checking files for updates if their modification time (mtime) has passed the <code><time_window></code> threshold. This improves the speed of file tracking operations when monitoring directory hierarchies with large numbers of historical files (for example, when active log files share a directory with old files that no longer get writes).</p> <p>Splunk Enterprise does not index files whose modification time falls outside <code><time_window></code> when it first attempts to monitor the file.</p> <p>You must specify <code><number><unit></code>. For example, "7d" indicates one week. Valid units are "d" (days), "h" (hours), "m" (minutes), and "s" (seconds).</p>	<p>0 (disabled)</p>
<code>followTail = 0 1</code>	<p>If set to 1, monitoring begins at the end of the file (like <code>*nix tail -f</code>). This only applies to files the first time Splunk Enterprise attempts to monitor them. After that, Splunk Enterprise keeps track of the file using its internal file position records.</p>	<p>0</p>
<code>whitelist = <regular expression></code>	<p>If set, Splunk Enterprise only monitors files whose names match the specified regular expression.</p>	<p>N/A</p>
<code>blacklist = <regular</code>	<p>If set, Splunk Enterprise does NOT monitor files whose names match the specified regular</p>	<p>N/A</p>

expression>	expression.	
<pre>alwaysOpenFile = 0 1</pre>	<p>If set to 1, Splunk Enterprise opens a file to check if it has already been indexed. This is only useful for files that don't update their modification time.</p> <p>Use this attribute for monitoring files on Windows, and mainly for Internet Information Server (IIS) logs.</p> <p>Caution: Use of this attribute increases load and slows down indexing.</p>	N/A
<pre>recursive = true false</pre>	If set to <code>false</code> , Splunk Enterprise does not look into subdirectories that it finds within a monitored directory.	true
<pre>time_before_close = <integer></pre>	The modification time delta required before Splunk Enterprise can close a file on End-of-file (EOF). Tells the system not to close files that have been updated in the past <code><integer></code> seconds.	3
<pre>followSymlink = true false</pre>	If <code>false</code> , Splunk Enterprise ignores symbolic links that it finds within a monitored directory.	true

Example 1. To load anything in `/apache/foo/logs` or `/apache/bar/logs`, etc.

```
[monitor:///apache/.../logs]
```

Example 2. To load anything in `/apache/` that ends in `.log`.

```
[monitor:///apache/*.log]
```

MonitorNoHandle syntax and examples

On Windows systems only, use the `MonitorNoHandle` stanza to monitor files without using Windows file handles. This allows you to read special log files like Windows's DNS server log files.

You must specify a valid path to a file when you use `MonitorNoHandle`. You cannot specify a directory. If you specify a file that already exists, Splunk Enterprise does not index the existing data in the file. It only indexes new data that the system writes to the file.

You can only configure `monitorNoHandle` using `inputs.conf` or the CLI. you cannot configure it in Splunk Web.

```
[MonitorNoHandle://<path>]
<attributel> = <val1>
<attribute2> = <val2>
...
```

Batch syntax and examples

Use batch to set up a one time, destructive input of data from a source. For continuous, non-destructive inputs, use **monitor**. Remember, after the batch input is indexed, Splunk **deletes** the file.

```
[batch://<path>]
move_policy = sinkhole
<attributel> = <val1>
<attribute2> = <val2>
...
```

When you define batch inputs, you must include the attribute, `move_policy = sinkhole`. This loads the file destructively. Do not use the batch input type for files that you do not want to delete after indexing.

Example: This example batch loads all files from the directory `system/flight815/`, but does not recurse through any subdirectories under it:

```
[batch://system/flight815/*]
move_policy = sinkhole
```

Note: To ensure that new events are indexed when you copy over an existing file with new contents, set the `CHECK_METHOD = modtime` attribute in `props.conf` for the source. This checks the modification time of the file and re-indexes it when it changes. Be aware that the entire file will be re-indexed, which can result in duplicate events.

Specify input paths with wildcards

Input path specifications in `inputs.conf` do not use regular expressions (regexes) but rather Splunk-defined wildcards. This topic discusses how to specify these wildcards in a path in `inputs.conf`. To specify wildcards, you must use `inputs.conf` to specify file and directory monitor inputs.

Wildcard overview

A wildcard is a character that you can substitute for one or more unspecified characters when searching text or selecting multiple files or directories. In Splunk Enterprise, you can use wildcards to specify the input path for a file or directory monitor input.

Wildcard	Description	Reg. Exp. equivalent	Example(s)
...	<p>The ellipsis wildcard recurses through directories and any number of levels of subdirectories to find matches.</p> <p>If you specify a folder separator (for example, <code>//var/log/.../file</code>), it does not match the first folder level, only subfolders.</p>	.*	<p><code>/foo/.../bar.log</code> matches the files <code>/foo/1/bar.log</code>, <code>/foo/2/bar.log</code>, <code>/foo/1/2/bar.log</code>, etc., but does not match <code>/foo/bar.log</code>, or <code>/foo/3/notbar.log</code></p> <p>Because a single ellipse recurses through all folders and subfolders, <code>/foo/.../bar.log</code> matches the same as <code>/foo/.../.../bar.log</code>.</p>
*	<p>The asterisk wildcard matches anything in that specific folder path segment.</p> <p>Unlike "...", "*" does not recurse through subfolders.</p>	[^/]*	<p><code>/foo/*/bar</code> matches the files <code>/foo/bar</code>, <code>/foo/1/bar</code>, <code>/foo/2/bar</code>, etc., but does not match <code>/foo/1/2/bar</code>.</p> <p><code>/foo/m*r/bar</code> matches <code>/foo/mr/bar</code>, <code>/foo/mir/bar</code>, <code>/foo/moor/bar</code>, etc.</p> <p><code>/foo/*.log</code> matches all files with the <code>.log</code> extension, such as <code>/foo/bar.log</code>. It does not match <code>/foo/bar.txt</code> or <code>/foo/bar/test.log</code>.</p> <p>A single period (.) is not a wildcard, and is the regular expression equivalent of <code>\.</code></p>

For more specific matches, combine the `...` and `*` wildcards. For example, `/foo/.../bar/*` matches any file in the `/bar` directory within the specified path.

Wildcards and regular expression metacharacters

When determining the set of files or directories to monitor, Splunk Enterprise splits elements of a monitoring stanza into segments. Segments are blocks of text between directory separator characters ("`/`" or "`\`") in the stanza definition. If you specify a monitor stanza that contains segments with both wildcards and regular expression metacharacters (such as `(`, `)`, `[`, `]`, and `|`), those characters behave differently depending on where the wild card is in the stanza.

If a monitoring stanza contains a segment with regular expression metacharacters before a segment with wildcards, Splunk Enterprise treats the metacharacters literally, as if you wanted to monitor files or directories with those characters in the file or directory names. For example:

```
[monitor://var/log/log(a|b).log]
```

monitors the `/var/log/log(a|b).log` file. Splunk Enterprise does not treat the `(a|b)` as a regular expression because no wildcards are present.

```
[monitor://var/log()/log*.log]
```

monitors all files in the `/var/log()/` directory that begin with `log` and have the extension `.log`. Splunk Enterprise does not treat the `()` as a regular expression because it is in the segment before the wildcard.

If the regular expression metacharacters occur within or after a segment that contains a wildcard, Splunk Enterprise treats the metacharacters as a regular expression and matches files to monitor accordingly. For example:

```
[monitor://var/log()/log(a|b)*.log]
```

monitors all files in the `/var/log()/` directory that begin with either `loga` or `logb` and have the extension `.log`. Splunk does not treat the first set of `()` as a regular expression because the wild card is in the following segment. The second set of `()` does get treated as a regular expression because it is in the same segment as the wildcard `'*'`.

```
[monitor://var/.../log(a|b).log]
```

monitors all files in any subdirectory of the `/var/` directory named `loga.log` and `logb.log`. Splunk treats `(a|b)` as a regular expression because of the wildcard

'...' in the previous stanza segment.

```
[monitor:///var/.../log[A-Z0-9]*.log]
```

monitors all files in any subdirectory of the `/var/` directory that:

- begin with `log`, then
- contain a single capital letter (from A-Z) or number (from 0-9), then
- contain any other characters, then
- end in `.log`.

Splunk Enterprise treats `[A-Z0-9]*` as a regex because of the wildcard '...' in the previous stanza segment.

Input examples

To monitor `/apache/foo/logs`, `/apache/bar/logs`, `/apache/bar/1/logs`:

```
[monitor:///apache/.../logs/*]
```

To monitor `/apache/foo/logs`, `/apache/bar/logs`, but not `/apache/bar/1/logs` or `/apache/bar/2/logs`:

```
[monitor:///apache/*/logs]
```

To monitor any file directly under `/apache/` that ends in `.log`:

```
[monitor:///apache/*.log]
```

To monitor any file under `/apache/` under any level of subdirectory that ends in `.log`:

```
[monitor:///apache/.../*.log]
```

The "..." followed by a folder separator will imply that the wildcard level folder will be excluded.

```
[monitor:///var/log/.../*.log]
```

the tailing logic will become `'^\\var\\log\\.*/[^']*\\.log$'`

Therefore, `/var/log/subfolder/test.log` will match, but `/var/log/test.log` will not match and be excluded. To monitor all files in all folders use:

```
[monitor:///var/log/]
```

```
whitelist=\.log$
```

```
recurse=true
```

```
#true by default
```

Wildcards and whitelisting

Splunk Enterprise defines whitelists and blacklists with standard Perl-compatible Regular Expression (PCRE) syntax.

When you specify wildcards in a file input path, Splunk Enterprise creates an implicit `whitelist` for that stanza. The longest wildcard-free path becomes the monitor stanza, and Splunk Enterprise translates the wildcards into regular expressions.

Splunk Enterprise anchors the converted expression to the right end of the file path, so that the entire path must be matched.

For example, if you specify

```
[monitor:///foo/bar*.log]
```

Splunk Enterprise translates this into

```
[monitor:///foo/]
whitelist = bar[^\.]*\.log$
```

On Windows, if you specify

```
[monitor://C:\Windows\foo\bar*.log]
```

Splunk Enterprise translates it into

```
[monitor://C:\Windows\foo\]
whitelist = bar[^\.]*\.log$
```

Note: In Windows, `whitelist` and `blacklist` rules do not support regular expressions that include backslashes. Use two backslashes (`\\`) to escape wildcards.

Whitelist- or blacklist-specific incoming data

Use **whitelist** and **blacklist** rules to explicitly tell Splunk Enterprise which files to consume when **monitoring** directories. You can also apply these settings to `batch` inputs. When you define a whitelist Splunk Enterprise indexes only the files in that list. When you define a blacklist, Splunk Enterprise ignores the files in that list and consumes everything else. Define whitelists and blacklists in the input stanza in `inputs.conf`.

It is not necessary to define both a whitelist and a blacklist in a stanza. They are independent settings. If you do define both and a file matches both, Splunk Enterprise does not index that file as `blacklist` overrides `whitelist`.

Whitelist and blacklist rules use regular expression syntax to define the match on the file name/path. They must be contained within a configuration stanza, for example `[monitor://<path>]`. Splunk Enterprise ignores whitelists and blacklists outside of stanzas.

To learn more about how to build regular expressions, visit the [Regular-expressions.info](http://regular-expressions.info) (<http://regular-expressions.info>) website.

When you define whitelist and blacklist entries, you must use exact regular expression syntax.

Route and filter data

Instead of whitelisting or blacklisting your data inputs, you can filter specific events and send them to different queues or indexes. You can also use [the crawl feature](#) to predefine files you want Splunk to index or not index when they get added to your file system.

Whitelist (allow) files

To define the files you want to exclusively index, add the following line to your `monitor` stanza in the `/local/inputs.conf` file for the app this input was defined in:

```
whitelist = <your_custom_regex>
```

For example, to monitor only files with the `.log` extension:

```
[monitor:///mnt/logs]
  whitelist = \.log$
```

You can whitelist multiple files in one line, using the `"|"` (OR) operator. For

example, to whitelist filenames that contain `query.log` OR `my.log`:

```
whitelist = query\.log$|my\.log$
```

Or, to whitelist exact matches:

```
whitelist = /query\.log$|/my\.log$
```

Note: The "\$" anchors the regex to the end of the line. There is no space before or after the "|" operator.

Blacklist (ignore) files

To define the files you want Splunk Enterprise to exclude from indexing, add the following line to your `monitor` stanza in the `/local/inputs.conf` file **for the app this input was defined in**:

```
blacklist = <your_custom_regex>
```

If you create a `blacklist` line for each file you want to ignore, Splunk activates only the last filter.

To ignore and not monitor only files with the `.txt` extension:

```
[monitor:///mnt/logs]
  blacklist = \.txt$
```

To ignore and not monitor all files with either the `.txt` extension OR the `.gz` extension (note that you use the "|" for this):

```
[monitor:///mnt/logs]
  blacklist = \.(?:txt|gz)$
```

To ignore entire directories beneath a monitor input refer to this example:

```
[monitor:///mnt/logs]
  blacklist = archive|historical|\.bak$
```

This example tells Splunk Enterprise to ignore all files under `/mnt/logs/` within the `archive` or `historical` directories and all files ending in `*.bak`.

To ignore files whose names contain a specific string, you can do:

```
[monitor:///mnt/logs]
  blacklist = 2009022[89]file\.txt$
```

This example ignores the `webserver20090228file.txt` and `webserver20090229file.txt` files under `/mnt/logs/`.

How Splunk Enterprise handles log file rotation

Splunk Enterprise recognizes when a file that it is monitoring (such as `/var/log/messages`) has been rolled by the operating system (`/var/log/messages1`) and will not read the rolled file a second time.

The monitoring processor picks up new files and reads the first 256 bytes of the file. The processor then hashes this data into a begin and end cyclic redundancy check (CRC), which functions as a fingerprint representing the file content. Splunk Enterprise uses this CRC to look up an entry in a database that contains all the beginning CRCs of files Splunk Enterprise has seen before. If successful, the lookup returns a few values, but the important ones are a **seekAddress**, meaning the number of bytes into the known file that Splunk Enterprise has already read, and a **seekCRC** which is a fingerprint of the data at that location.

Using the results of this lookup, Splunk Enterprise can attempt to categorize the file.

There are three possible outcomes of a CRC check:

- No matching record for the CRC from the file beginning in the database. This indicates a new file. Splunk Enterprise picks it up and consume its data from the start of the file. Splunk Enterprise updates the database with the new CRCs and Seek Addresses as it consumes the file.
- A matching record for the CRC from the file beginning in the database, the content at the Seek Address location matches the stored CRC for that location in the file, and the size of the file is larger than the Seek Address that Splunk Enterprise stored. While Splunk Enterprise has seen the file before, data has been added since it was last read. Splunk Enterprise opens the file, seeks to Seek Address--the end of the file when Splunk last finished with it--and starts reading from there. Splunk Enterprise only reads the new data.
- A matching record for the CRC from the file beginning in the database, but the content at the Seek Address location does not match the stored CRC at that location in the file. Splunk Enterprise has read some file with the same initial data, but either some of the material that it read has been modified in place, or it is in fact a wholly different file which begins with the

same content. Because the database for content tracking is keyed to the beginning CRC, it has no way to track progress independently for the two different data streams, and further configuration is required.

Because the CRC start check runs against only the first 256 bytes of the file by default, it is possible for non-duplicate files to have duplicate start CRCs, particularly if the files are ones with identical headers. To handle such situations you can:

- Use the `initCrcLength` attribute in `inputs.conf` to increase the number of characters used for the CRC calculation, and make it longer than your static header.
- Use the `crcSalt` attribute when configuring the file in `inputs.conf`, as described in "[Monitor files and directories with inputs.conf](#)" in this manual. The `crcSalt` attribute, when set to `<SOURCE>`, ensures that each file has a unique CRC. The effect of this setting is that Splunk Enterprise assumes that each path name contains unique content.

Do not use `crcSalt = <SOURCE>` with rolling log files, or any other scenario in which logfiles get renamed or moved to another monitored location. Doing so prevents Splunk Enterprise from recognizing log files across the roll or rename, which results in the data being reindexed.

Get data from network sources

Get data from TCP and UDP ports

You can configure Splunk Enterprise to accept an input on any TCP or UDP port. Splunk Enterprise consumes any data that arrives on these ports. Use this method to capture data from network services such as syslog (default port is UDP 514). You can also set up the netcat service and bind it to a port.

TCP is the network protocol that underlies the Splunk Enterprise data distribution scheme. It is the recommended protocol for sending data from any remote host to your Splunk Enterprise server. Splunk Enterprise can index remote data from `syslog-ng` or any other application that transmits via TCP.

Splunk Enterprise supports monitoring over UDP, but you should use TCP to send network data instead whenever possible. UDP is not desirable as a transport because, among other reasons, it does not guarantee delivery of network packets.

When you monitor TCP network ports, the user Splunk Enterprise runs as must have access to the port you want to monitor. On many Unix operating systems, by default, you must run Splunk Enterprise as the root user to listen directly on a port below 1024.

See [Working with UDP connections](#) on the Splunk Community Wiki for recommendations if you must send network data with UDP.

Confirm how your network device handles external monitoring before you use the network monitoring input

Before you begin monitoring the output of a network device with the Splunk Enterprise network monitor, confirm how the device interacts with external network monitors.

If you configure TCP logging on some network devices, such as a Cisco Adaptive Security Appliance (ASA), and the device cannot connect to the monitor, it might cause reduced performance or stop logging, or worse. By default, the Cisco ASA stops accepting incoming network connections when it encounters network congestion or connectivity problems.

Add a network input using Splunk Web

To add inputs from network ports using Splunk Web:

Go to the Add New page

You can get there through two routes.

By Splunk Settings:

1. Click **Settings**.
2. Click **Data Inputs**.
3. Choose **TCP** or **UDP**.
4. Click **New** to add an input.

By Splunk Home:

1. Click the **Add Data** link in Splunk Home.
2. Click **Monitor** to monitor a network port on the local machine, or **Forward** to receive network data from another machine.

Note: Forwarding a file requires additional setup.

3. If you selected **Forward**, choose or create the group of forwarders you want this input to apply to.
4. Click **Next**.

Specify the network input

1. In the left pane, click **TCP / UDP** to add an input.
2. Click the **TCP** or **UDP** button to choose between a TCP or UDP input.
3. In the **Port** field, enter a port number.
4. In the **Source name override** field, enter a new source name to override the default source value, if necessary.

Note: Consult Splunk Support before changing the "Source name override" value.

5. If this is a TCP input, specify whether this port should accept connections from all hosts or only one host in the **Only accept connections from** field. If you only want the input to accept connections from one host, enter the host name or IP address of the host. You can use wildcards to specify hosts.

6. Click **Next** to continue to the **Input Settings** page.

Specify input settings

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

1. Set the **Source type**. This is a default field that Splunk Enterprise adds to events and uses to determine processing characteristics, such as timestamps and event boundaries.

2. Set the **Host** name value. You have several choices:

- **IP.** Sets the input processor to rewrite the host with the IP address of the remote server.
- **DNS.** Sets the host to the DNS entry of the remote server.
- **Custom.** Sets the host to a user-defined label.

Learn more about setting the host value in ["About hosts"](#).

Note: **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 for this input. 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 the 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 are not 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 network input.

Add a network input using the CLI

To access the Splunk Enterprise CLI, navigate to the `$SPLUNK_HOME/bin/` directory and use the `./splunk` command.

If you get stuck, the CLI has help. Access the main CLI help by typing `splunk help`. Individual commands have their own help pages as well and can be accessed by typing `splunk help <command>`.

The following CLI commands are available for network input configuration:

Command	Command syntax	Action
add	<code>add tcp udp <port> [-parameter value] ...</code>	Add inputs from <port>.
edit	<code>edit tcp udp <port> [-parameter value] ...</code>	Edit a previously added input for <port>.
remove	<code>remove tcp udp <port></code>	Remove a previously added data input.
list	<code>list tcp udp [<port>]</code>	List the currently configured monitor.

The <port> is the port number on which to listen for data. The user you run Splunk as must have access to this port.

You can modify the configuration of each input by setting any of these additional parameters:

Parameter	Required?	Description
-----------	-----------	-------------

<code>sourcetype</code>	No	Specify a sourcetype field value for events from the input source.
<code>index</code>	No	Specify the destination index for events from the input source.
<code>hostname</code>	No	Specify a host name to set as the host field value for events from the input source.
<code>remotehost</code>	No	Specify an IP address to exclusively accept data from.
<code>resolvehost</code>	No	Set to true or false (T F). Default is False. Set to true to use DNS to set the host field value for events from the input source.
<code>restrictToHost</code>	No	Specify a host name or IP address that this input should accept connections from only.

Examples

- Configure a UDP input to watch port 514 and set the source type to "syslog":

```
./splunk add udp 514 -sourcetype syslog
```

- Set the UDP input host value via DNS. Use `auth` with your username and password:

```
./splunk edit udp 514 -resolvehost true -auth admin:changeme
```

For information on best practices for using UDP, see [Best practices for configuring Syslog input in the Community Wiki](#).

Change restricted hosts on a TCP network input

If you decide to only accept connections from a specific host when you create a TCP input, once you save that input, you can neither change nor remove that host later, either from Splunk Web or the CLI.

To change or remove the restricted host of a port, you must first delete the input that contains the old restricted host. Then, you must add a new input that either contains the new restricted host or has no restriction.

Add a network input using inputs.conf

To add an input, add a stanza for it to inputs.conf in

`$SPLUNK_HOME/etc/system/local/`, or your own custom application directory in `$SPLUNK_HOME/etc/apps/`. If you have not worked with Splunk's configuration files before, read *About configuration files* in the *Admin* manual before you begin.

You can set any number of attributes and values following an input type. If you do not specify a value for one or more attributes, Splunk Enterprise uses the defaults that are preset in `$SPLUNK_HOME/etc/system/default/` (noted below).

Configure a TCP input

```
[tcp://<remote server>:<port>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

Tells Splunk Enterprise to listen to `<remote server>` on `<port>`. If `<remote server>` is blank, Splunk Enterprise listens to all connections on the specified port.

Attribute	Description	Default
<code>host = <string></code>	Sets the host key/field to a static value for this stanza. Also sets the host key initial value. Splunk Enterprise uses the key during parsing and indexing, in particular to set the host field. It also uses the host field at search time. The <code><string></code> is prepended with 'host::'.	the IP address or fully-qualified domain name of the host where the data originated.
<code>index = <string></code>	Sets the index where Splunk Enterprise should store the events from this input. The <code><string></code> is prepended with 'index::'.	<code>main</code> or whatever you set the default index to
<code>sourcetype = <string></code>	Sets the sourcetype key/field for events from this input. Also declares the source type for this data, instead of letting Splunk Enterprise determine it. This is	Splunk Enterprise picks a source type based on various aspects of the data. There is no hard-coded default.

	<p>important both for searchability and for applying the relevant formatting for this type of data during parsing and indexing.</p> <p>Sets the sourcetype key initial value. Splunk Enterprise uses the key during parsing and indexing, in particular to set the source type field during indexing. Splunk Enterprise uses the source type field used search time.</p> <p>The <code><string></code> is prepended with 'sourcetype::'.</p>	
<pre>source = <string></pre>	<p>Sets the source key/field for events from this input. The <code><string></code> is prepended with 'source::'.</p> <p>Note: Do not override the source key unless absolutely necessary. The input layer provides a more accurate string to aid in problem analysis and investigation by recording the file from which the data was retrieved. Consider use of source types, tagging, and search wildcards before overriding this value.</p>	The input file path
<pre>queue = parsingQueue indexQueue</pre>	<p>Specifies where the input processor should deposit the events that it reads.</p> <p>Set to "parsingQueue" to apply <code>props.conf</code> and other parsing rules to your data. Set to "indexQueue" to send your data directly into the index.</p>	parsingQueue
<pre>connection_host = ip dns none</pre>	"ip" sets the host to the IP address of the remote server.	ip

	<p>"dns" sets the host to the DNS entry of the remote server.</p> <p>"none" leaves the host as specified.</p>	
--	---	--

Configure a TCP input over SSL

```
[tcp-ssl:<port>]
```

Use this stanza type if you receive encrypted, unparsed data from a forwarder or third-party system. Set `<port>` to the port on which the forwarder or third-party system is sending unparsed, encrypted data.

Configure a UDP input

```
[udp://<remote server>:<port>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

This type of input stanza is similar to the TCP type, except that it listens on a UDP port.

- If you specify `<remote server>`, the specified port only accepts data from that host.
- If you specify nothing for `<remote server>` - `[udp://<port>]` - the port accepts data sent from any host.

Attribute	Description	Default
<code>host = <string></code>	Sets the host key/field to a static value for this stanza. Also sets the host key initial value. Splunk Enterprise uses this key during parsing and indexing, in particular to set the host field. It also uses the host field at search time. The <code><string></code> is prepended with 'host::'.	the IP address or fully-qualified domain name of the host where the data originated.
<code>index = <string></code>	Sets the index where Splunk Enterprise should	<code>main</code> or whatever you set the default index to

	store events from this input. The <code><string></code> is prepended with 'index::'.	
<code>sourcetype = <string></code>	<p>Sets the sourcetype key/field for events from this input. Also declares the source type for this data, as opposed to letting Splunk Enterprise determine it. This is important both for searchability and for applying the relevant formatting for this type of data during parsing and indexing.</p> <p>Sets the sourcetype key initial value. Splunk Enterprise uses the key during parsing and indexing, in particular to set the source type field during indexing. It also uses the source type field used search time.</p> <p>The <code><string></code> is prepended with 'sourcetype::'.</p>	Splunk Enterprise picks a source type based on various aspects of the data. There is no hard-coded default.
<code>source = <string></code>	<p>Sets the source key/field for events from this input. The <code><string></code> is prepended with 'source::'.</p> <p>Note: Do not override the source key unless absolutely necessary. The input layer provides a more accurate string to aid in problem analysis and investigation by recording the file from which the data was retrieved. Consider</p>	The input file path

	use of source types, tagging, and search wildcards before overriding this value.	
<code>queue = parsingQueue indexQueue</code>	Sets where the input processor should deposit the events that it reads. Set to "parsingQueue" to apply <code>props.conf</code> and other parsing rules to your data. Set to "indexQueue" to send your data directly into the index.	parsingQueue
<code>_rcvbuf = <integer></code>	Sets the receive buffer for the UDP port, in bytes. If the value is 0 or negative, Splunk Enterprise ignores the value.	1,572,864 unless the value is too large for an OS. In this case, Splunk Enterprise halves the value from this default continuously until the buffer size is at an acceptable level.
<code>no_priority_stripping = true false</code>	<p>Sets how Splunk Enterprise handles receiving syslog data.</p> <p>If you set this attribute to true, Splunk Enterprise does not strip the <priority> syslog field from received events.</p> <p>Depending on how you set this attribute, Splunk Enterprise also sets event timestamps differently. When set to true, Splunk Enterprise honors the timestamp as it comes from the source. When set to false, Splunk Enterprise assigns events the local</p>	false (Splunk Enterprise strips <priority>.)

	time.	
<code>no_appending_timestamp</code> <code>= true false</code>	<p>Sets how Splunk Enterprise applies timestamps and hosts to events.</p> <p>If you set this attribute to true, Splunk Enterprise does not append a timestamp and host to received events.</p> <p>Note: Do not set this attribute if you want to append timestamp and host to received events.</p>	false (Splunk Enterprise appends timestamps and hosts to events)

UDP packets and line merging

Splunk Enterprise does not index each UDP packet as an independent event. Instead, it performs event merging on the data stream and merges events together if they don't have a clear timestamp.

You can avoid this problem by editing the underlying source type in `props.conf` and setting the `SHOULD_LINEMERGE` attribute to `false`. This keeps Splunk Enterprise from merging packets together.

Answers

Have questions? Visit [Splunk Answers](#) and see what answers the Splunk community has about questions UDP inputs, TCP inputs, and inputs in general,

Set up and use HTTP Event Collector

HTTP Event Collector (EC) is an endpoint that lets you send application events into Splunk Enterprise using the HTTP or Secure HTTP (HTTPS) protocols. Event Collector uses an authentication model based on tokens that you generate. You then configure a logging library or HTTP client with this token to send data to EC in a specific format. This process eliminates the need for a forwarder when sending application events.

EC was created with application developers in mind, so that all it takes is a few lines of code added to an app for the app to send data. Also, EC is token-based, so you never need to hard-code your Splunk Enterprise credentials in your app or supporting files.

EC runs as a separate app in Splunk Enterprise called `splunk_httpinput` and stores its input configuration there in

`$SPLUNK_HOME/etc/apps/splunk_httpinput/local.`

About Event Collector Tokens

Tokens are entities that let logging agents and clients connect to the HTTP Event Collector endpoint. Each token has a token value: a 32-bit number that agents and clients use to authenticate their connections to EC. When they connect, they present this token value. If EC has the token value configured and it is active, EC accepts the connection and the agent can then begin delivering its payload of application events in JavaScript Object Notation (JSON) format.

EC receives the events and Splunk Enterprise indexes them based on the configuration of the token that the agent used to connect. Splunk Enterprise uses the source, source type, and index that was specified in the token. If a forwarding output group configuration exists, Splunk Enterprise then forwards the application events to other indexers as the output group defines them.

Configure HTTP Event Collector in Splunk Web

Enable HTTP Event Collector

Before you can use Event Collector to receive events through HTTP, you must enable it. You enable EC through the "Global Settings" dialog box in the EC management page.

1. From the system bar, click **Settings > Data Inputs**.
2. On the left side of the page, click **HTTP Event Collector**. The EC management page loads.
3. In the upper right corner, click **Global Settings**.

The screenshot shows the 'Edit Global Settings' dialog box. It contains the following settings:

- All Tokens:** A toggle with 'Enabled' selected and 'Disabled' as an alternative.
- Default Source Type:** A dropdown menu currently showing '--Select Source Type--'.
- Default Index:** A dropdown menu currently showing 'Default'.
- Default Output Group:** A dropdown menu currently showing 'None'.
- Use Deployment Server:** An unchecked checkbox.
- Enable SSL:** A checked checkbox.
- HTTP Port Number:** A text input field containing the value '8088'.

At the bottom of the dialog are two buttons: 'Cancel' and 'Save'.

4. In the **All Tokens** toggle button, select **Enabled**.

5. To set the source type for all EC tokens, select a category from the **Default Source Type** drop-down, then select the source type you want. You can also type in the name of the source type in the text field above the drop-down before choosing the source type.

6. To set the default index for all EC tokens, choose an index in the **Default Index** drop-down.

7. To set the default forwarding output group for all EC tokens, choose an output group from the **Default Output Group** drop-down.

8. To use a deployment server to handle configurations for EC tokens, click the **Use Deployment Server** check box.

9. To have EC listen and communicate over HTTPS rather than HTTP, click the **Enable SSL** checkbox.

10. To specify the port number that EC listens on, enter a number in the **HTTP Port Number** field.

Note: To ensure that proper communication happens between logging agents and EC, confirm that no firewall blocks the port number specified in the **HTTP Port Number** field, either on the agents, the Splunk Enterprise instance that hosts EC, or in between.

11. To save your settings, click **Save**. The dialog box disappears and Splunk Enterprise saves the global settings and returns you to the EC management page.

Create an Event Collector token

To use the HTTP Event Collector, you must configure at least one token. The token is what clients and agents use when they connect to Event Collector to send data.

1. From the Settings menu, select **Add Data**.
2. Select **monitor**, and then in the left pane, select **HTTP Event Collector**. The right pane populates with fields for EC end point.
3. In the **Name** field, enter a name for the token that describes its purpose and that you will remember.
4. (Optional) In the **Source name override** field, enter a name for a source that Splunk Enterprise should assign to events that this end point generates.
5. (Optional) In the **Description** field, enter a description for the input.
6. (Optional) In the **Output Group** field, select an existing forwarder output group by picking it in the drop-down list.

Note: Define output groups in `outputs.conf`. See [Configure forwarders with outputs.conf](#). You can also set up forwarding in Splunk Web, which generates a default output group called `default-autolb-group`.

7. (Optional) If you want to enable **indexer acknowledgment** for this token, click the **Enable indexer acknowledgment** checkbox.

Note: Indexer acknowledgement is verification from the indexer that events have been indexed. Indexer acknowledgement in HTTP Event Collector is not the same indexer acknowledgement capability described in *Protect against loss of in-flight data in the Splunk Enterprise Forwarding Data* manual. For more information about indexer acknowledgement in HTTP Event Collector, see [Enable indexer acknowledgement](#).

8. Click **Next**. The **Input Settings** page displays.
9. Make edits to source type and confirm the index you want Splunk Enterprise to send EC events to. See "[Modify input settings](#)."
10. Click **Review**. Confirm that all settings for the end point are what you want. If you need to change settings, click the gray < button at the top of the page.

11. If all settings are what you want, click **Next**. The success page loads and displays the token value that Event Collector generated. You can copy this token value from the displayed field and paste it into another document for reference later. See "[About Event Collector tokens.](#)"

Modify an Event Collector token

Dialog box titled "Edit Token: JSON stream inbound".

Fields and options:

- Description: optional
- Source: optional
- Set Source Type: Entered sourcetype
- Source Type:
- Select Allowed Indexes (optional):
 - Available indexes: history, main, pas, pas_keycard, summary
 - Selected indexes: main
 - Buttons: add all, remove all
 - Note: Select indexes that clients will be able to select from.
- Default Index: main
- Output Group (optional): None
- Enable indexer acknowledgement: ☒

Buttons: Cancel, Save

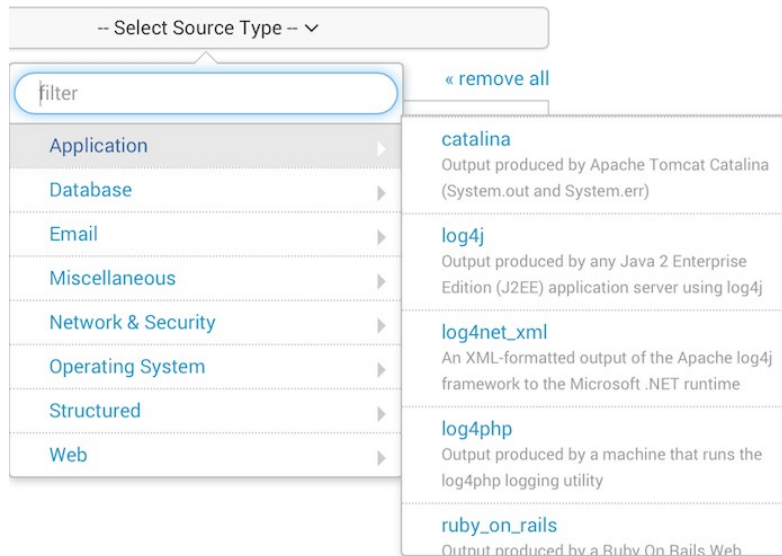
You can make changes to an EC token after you have created it. Visit the EC management page and edit a token to change any of its characteristics, including its name, description, default source type, default index, and output group.

To change the properties of a token:

1. Go to the EC management page. From the **Settings** menu, select **Data Inputs**.
2. Select **HTTP Event Collector**.
3. Locate the token that you want to change in the list.
4. In the **Actions** column for that token, click **Edit**. You can also click the link to the token name.
5. Edit the description of the token by entering updated text in the **Description** field.
6. (Optional) Update the source value of the token by entering text in the **Source**

field.

7. (Optional) Choose a different source type by selecting it in the **Source Type** drop-down. First choose a category, then select a source type in the pop-up menu that appears. You can also type in the name of the source type in the text box at the top of the drop-down.



8. (Optional) Choose a different index by selecting it in the **Available Indexes** pane of the **Select Allowed Indexes** control. The index moves to the **Selected Indexes** pane of the control.

9. (Optional) Choose a different output group from the **Output Group** drop-down.

10. (Optional) Choose whether or not you want indexer acknowledgment enabled for the token.

11. Click **Save**. The dialog closes and Splunk Enterprise returns you to the EC management page.

Delete an Event Collector token

You can also delete an EC token if you don't plan to use it any more. Deleting an EC token does not affect other EC tokens, nor does it disable the EC endpoint.

Caution: You cannot undo this action. Agents that use this token to send data to Splunk Enterprise will no longer be able to authenticate with the token. You must generate a new token and change the agent

configuration to use the new token value.

To delete an EC token:

1. Go to the EC management page. From the **Settings** menu, select **Data Inputs**.
2. Select **HTTP Event Collector**.
3. Locate the token that you want to delete in the list.
4. In the **Actions** column for that token, click **Delete**.
5. In the Delete Token dialog, click **Delete**. Splunk Enterprise deletes the token and returns you to the EC management page.

Enable and disable Event Collector tokens

You can enable or disable a single EC token from within the EC management page. Changing the status of one token does not change the status of other tokens. To enable or disable all tokens, use the Global Settings dialog. See ["Enable the HTTP Event Collector."](#)

To toggle the active status of an EC token:

1. Go to the EC management page.
2. Locate the token whose status you want to toggle.
3. In the **Actions** column for that token, click the **Enable** link (if the token is active) or the **Disable** link (if the token is inactive.) The token status toggles immediately and the link changes to **Enable** or **Disable** based on the changed token status.

Making use of HTTP Event Collector from a developer perspective

You have several options within your developer environment for using HTTP Event Collector. You can use our Java, JavaScript (Node.js) and .NET logging libraries, which are compatible with popular logging frameworks. Or you can make an HTTP request using your favorite HTTP client and send your JSON-encoded events.

Making an HTTP call with the command line using a `curl` command in your operating system is an easy way to test this out.

Example:

Note: This POST request is made to port 8088 and uses HTTPS for transport. The port and HTTP protocol settings can be configured independently of settings for any other servers in your deployment.

JSON

The following cURL statement uses an example HTTP Event Collector token (B5A79AAD-D822-46CC-80D1-819F80D7BFB0), and uses `https://localhost` as the hostname. Replace these values with your own before executing this statement.

JSON Request

```
curl -k https://localhost:8088/services/collector/event -H
"Authorization: Splunk B5A79AAD-D822-46CC-80D1-819F80D7BFB0" -d
'{"event": "hello world"}'
```

Note: the key "event" is required.

JSON Response

```
{"text": "Success", "code": 0}
```

More information

You can find more developer-related content about using HTTP Event Collector in the Splunk Developer Portal. For a complete walkthrough of using HTTP Event Collector, see "HTTP Event Collector walkthrough".

How Splunk Enterprise handles syslog data

This topic describes how Splunk Enterprise handles data that it receives when you have it listen on a UDP network port for syslog data.

Splunk Enterprise can act as a syslog server or a syslog message sender. It should not be substituted for such a server in regular use. This is because Splunk Enterprise modifies syslog data by default as part of the indexing process (it assigns a timestamp and a host to the event.)

If you must retain raw syslog data (for example, a data retention policy requires access to untouched events), then consider using a tool such as `syslog-ng` to simultaneously save the raw data to a log file and forward events into Splunk. This gives you the added advantage of indexing the log file into Splunk Enterprise later if you want.

See the diagrams later in this topic for a description of how Splunk Enterprise handles syslog events over UDP.

How Splunk Enterprise handles syslog inputs

When you configure a UDP network input to listen to a syslog in Splunk Enterprise, any syslog events that arrive through the input receive a timestamp and connected host field. Splunk Enterprise prepends these fields to each event before indexing.

You can change this behavior by setting the `no_appending_timestamp` attribute in `inputs.conf`.

If the data contains a syslog header, Splunk Enterprise strips it out unless you set the `no_priority_stripping` attribute in the stanza.

Splunk Enterprise does not modify TCP packets in this fashion. If you send syslog data over TCP, Splunk Enterprise does not strip priority information from the events. It does, however, prepend a host name and time stamp to the event unless you tell it not to.

How Splunk Enterprise handles syslog outputs

Splunk Enterprise can also forward events to another syslog server. When it does, it prepends the priority information to the event so that the downstream syslog server can translate the events properly.

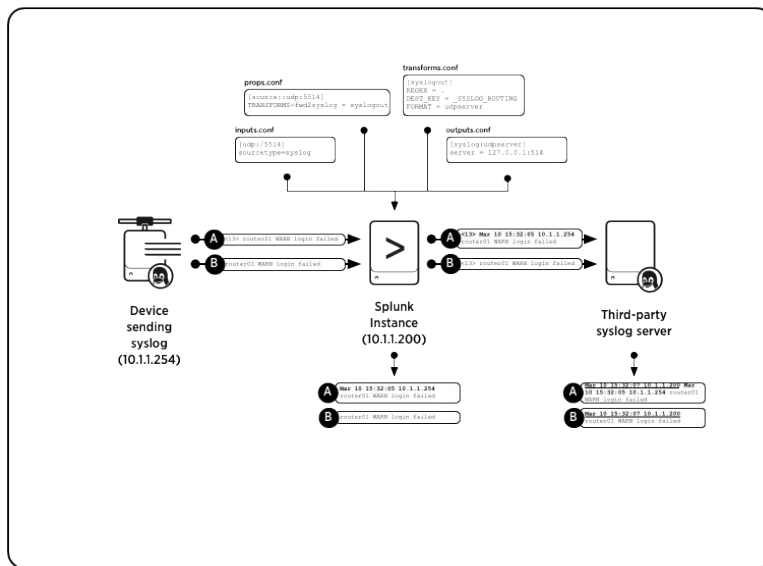
When the event reaches the downstream syslog server, that host prepends a timestamp, priority, and connected host name, which is the Splunk Enterprise instance.

You can also prepend a timestamp and host name to the event at the time you forward the event to the syslog server.

For information on configuring routing, filtering, and usage of source types, see *Route and filter data* in the *Forwarding Data* manual and the `props.conf` spec file in the *Admin* manual.

How Splunk Enterprise moves syslog events when you configure it to use syslog source type

The following diagram shows how Splunk Enterprise moves two syslog messages from one syslog server to another. In the diagram, Splunk Enterprise listens on a UDP network port and indexes incoming events. On the other side, the same instance forwards events to a second, third-party syslog server.



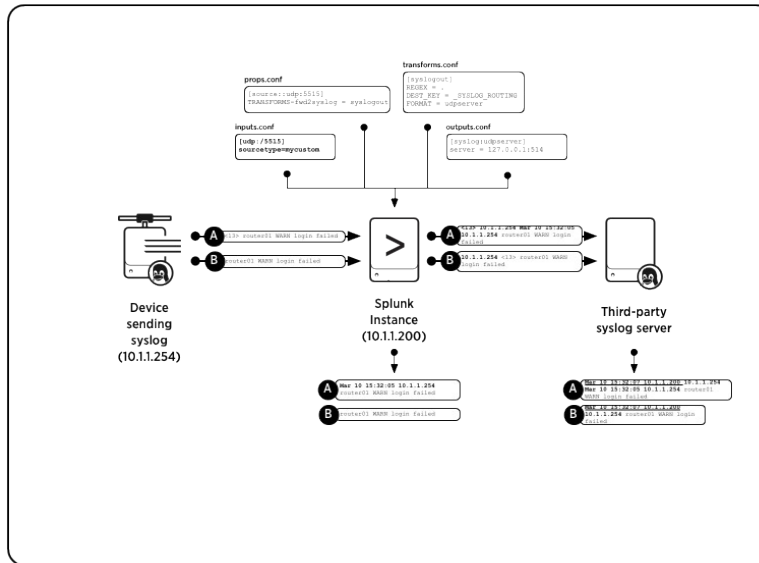
In the diagram, Message A originates as a syslog event and Message B originates as a similar event that does not have priority information associated with it. Upon receipt, Splunk Enterprise tags the events with a timestamp and the host that generated the event.

If you configured the instance as a forwarder, Splunk Enterprise then transforms the events by adding a priority header (that you specify in `outputs.conf`) before it forwards the events on to the syslog server. Once they arrive at the syslog server, that server prepends timestamp and host data to the events as it received them from the Splunk Enterprise instance.

How Splunk Enterprise moves syslog events when you configure a custom source type

In this diagram, Splunk Enterprise has been configured to use a non-syslog source type.

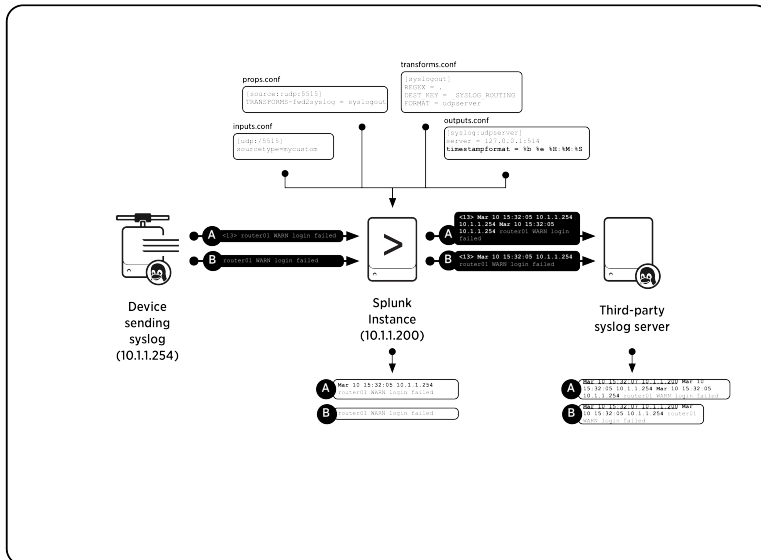
The initial Messages A and B are identical to the first example. In this example, Splunk Enterprise prepends the event with an originating host name or IP address.



How Splunk Enterprise moves syslog events when you configure it with timestamping

You can also configure Splunk Enterprise to add timestamps to syslog events when you forward those events. You could time stamp the events when you don't want the downstream server to add its own timestamp. The following diagram shows the required attribute and depicts how Splunk Enterprise deals with the data.

The initial Messages A and B are identical to the first and second examples. Splunk Enterprise prepends the events with a timestamp and an originating host name or IP address.



Send SNMP events to Splunk Enterprise

Simple Network Management Protocol (SNMP) traps are alerts that remote devices send out. This topic describes how to receive and index SNMP traps at the Splunk Enterprise indexer.

Note: The procedures shown in this topic (for both *nix and Windows) are examples only. You can accomplish the task of sending SNMP traps to Splunk Enterprise in a number of ways. For example, instead of using Net-SNMP, you can use other tools, such as Snare or SNMPGate, to write SNMP traps to file storage for monitoring by Splunk Enterprise.

How to index SNMP traps

The most effective way to index SNMP traps is to first write them to a file on the Splunk Enterprise server. Then, configure Splunk Enterprise to monitor the file.

To consume SNMP trap data:

1. Configure the remote devices to send their traps directly to the Splunk Enterprise instance IP address. The default port for SNMP traps is `udp:162`.
2. Write the SNMP traps to a file on the Splunk Enterprise instance, as described in ["Write SNMP traps to a file on the Splunk Enterprise server."](#)

3. Configure Splunk Enterprise to monitor the file, as described in "[Monitor files and directories](#)".

Note: This topic does not cover SNMP polling, which is a way to query remote devices.

Write SNMP traps to a file on the Splunk Enterprise instance

Use your favorite SNMP software to write the SNMP traps to a file. For information about available SNMP software, visit the SNMP portal (<http://www.snmpLink.org>) website.

*For *nix*

On *nix, you can use the Net-SNMP project `snmptrapd` binary to write SNMP traps to a file.

Before you install `snmptrapd` on your system, see the local documentation for the version of `snmptrapd` that comes with your distribution of *nix. See also the manual page for `snmptrapd`.

The simplest configuration is:

```
# snmptrapd -Lf /var/log/snmp-traps
```

Note: Versions 5.3 and later of `snmptrapd` apply access control checks to all incoming notifications instead of accepting and logging them automatically (even if no explicit configuration was provided). If you run `snmptrapd` without suitable access control settings, then it does not process those traps. You can avoid this by specifying:

```
# snmptrapd -Lf /var/log/snmp-traps --disableAuthorization=yes
```

To see the version of `snmptrapd`, run `snmptrapd --version` from the command prompt.

Troubleshoot problems with SNMP

If you experience problems sending SNMP traps to Splunk Enterprise, consider that:

- UDP port 162 is a privileged network port. If you need to use this port, then you must run `snmptrapd` as root.

- You can use the `-f` flag to keep `snmptrapd` in the foreground while testing.
- You can use the `-Lo` flags instead of `-Lf` to log to standard output.
- You can use the `snmptrapd` command to generate an example trap, as in:

```
# snmptrap -v2c -c public localhost 1 1
```

For Windows

To log SNMP traps to a file on Windows.

1. Download and install the latest version of `NET-SNMP` for Windows from the `NET-SNMP` website.

Note: The OpenSSL library must not be installed on the system because it conflicts with `NET-SNMP`.

2. Register `snmptrapd` as a service using the script included in the `NET-SNMP` install.

3. Edit `C:\usr\etc\snmp\snmptrapd.conf`:

```
snmpTrapdAddr [System IP]:162
authCommunity log [community string]
```

4. The default log location is `C:\usr\log\snmptrapd.log`

Use Management Information Bases (MIBs)

Management Information Bases (MIBs) provide a map between numeric object IDs (OIDs) reported by the SNMP trap and a textual human readable form. Though `snmptrapd` can work without any MIB files at all, it won't display the results in exactly the same way.

The vendor of the device you receive SNMP traps from can provide a specific MIB. For example, all Cisco device MIBs can be located using the online Cisco SNMP Object Navigator.

To add a new MIB file:

1. Download and copy the MIB file into the MIB search directory. On the *nix version of `Net-SNMP`, the default location is `/usr/local/share/snmp/mibs`. You can set a different directory by providing the `-m` argument to `snmptrapd`.

2. Instruct `snmptrapd` to load the MIB(s) by passing a colon-separated list to the `-m` argument.

Note:

- If you add a leading '+' character for the parameters in the `-m` argument, `snmptrapd` loads the MIB in addition to the default list, instead of overwriting the list.
- The special keyword `ALL` tells `snmptrapd` to load all MIB modules in the MIB directory.

For example, to load all MIB modules in the MIB directory:

```
snmptrapd -m +ALL
```

Get Windows data

About Windows data and Splunk Enterprise

Splunk Enterprise can index many different kinds of Windows data. This data can be pretty much anything: an Event Log channel, the Registry, or Active Directory. You also have available the standard set of Splunk Enterprise inputs, such as files and directories, the network monitoring inputs, and scripted inputs.

The following specialized inputs are available only on Windows installations of Splunk Enterprise:

- **Windows Event Logs.** [Monitor events](#) generated by the Windows Event Log service on any available event log channel on the machine. You can collect events on the local machine or remotely by using either a universal forwarder or Windows Management Instrumentation (WMI).
- **Performance monitoring.** [Collect performance data](#) on Windows machines with Splunk Enterprise and then alert or report on that data. Any performance counter that is available in Performance Monitor is also available to Splunk Enterprise. You can monitor performance locally or remotely through a universal forwarder or WMI.
- **Remote monitoring over WMI.** Splunk Enterprise can [use WMI](#) to access event log and performance data on remote machines.
- **Registry monitoring.** You can [monitor changes to the local Windows Registry](#) using the Registry monitoring capability. You can use a universal forwarder to gather Registry data from remote machines.
- **Active Directory monitoring.** Splunk Enterprise can audit any [changes to the Active Directory](#) including changes to user, group, machine, and group policy objects. You can forward Active Directory data to another Splunk Enterprise server.

The Splunk App for Windows Infrastructure

The Splunk App for Windows Infrastructure provides data inputs, searches, reports, alerts, and dashboards for Windows server and desktop management.

You can monitor, manage, and troubleshoot Windows operating systems from one place. The app includes inputs for CPU, disk I/O, memory, event logs, configurations, and user data, plus a web-based setup UI for indexing Windows event logs.

Initial considerations for deploying Splunk Enterprise on Windows

When you install and deploy Splunk Enterprise on Windows, consider the following:

- **Authentication.** To perform any operations on remote Windows machines in your network, Splunk Enterprise must run as a user with credentials to access those machines. Make these credentials available before deploying. See ["Considerations for deciding how to monitor remote Windows data."](#)
- **Disk bandwidth.** Splunk Enterprise indexers require lots of disk I/O bandwidth, particularly when indexing large amounts of data. Make sure that you configure any installed antivirus software to avoid monitoring Splunk Enterprise directories or processes, because such scans significantly reduce performance.
- **Shared hosts.** Before you install Splunk Enterprise on a host that runs other services, such as Exchange, SQL Server, or a hypervisor, see Introduction to capacity planning for Splunk Enterprise in the *Capacity Planning* manual.

The most efficient way to gather data from any Windows server is to [install universal forwarders](#) on the hosts that you want to gather data. Universal forwarders use limited resources. In some cases, such as Registry monitoring, you must use a forwarder, because you cannot collect Registry data over WMI.

How to get Windows data into Splunk Enterprise

Splunk Enterprise lets you collect many different kinds of Windows data.

When you download and install Splunk Enterprise on a Windows machine, you can collect the following Windows statistics:

- [Windows Event Logs](#)

- File system changes
- Active Directory
- data over the Windows Management Instrumentation (WMI) infrastructure
- Registry data
- Performance metrics
- Host information
- Print information
- Network information

You can collect all of these types of data only on Windows hosts. Other operating systems cannot collect Windows data directly. You can forward Windows data from Windows hosts to Splunk Enterprise instances that do not run Windows.

Use Splunk Web to collect Windows data

Nearly all Windows inputs let you collect Windows data by using the Splunk Web interface. The exception is the `MonitorNoHandle` input, which you must set up by using a configuration file.

1. Log into your Splunk Enterprise instance.
2. Click **Settings** in the upper right corner, then click **Data inputs**. The **Data inputs** page appears.
3. Find the input that you want to add in the list of available inputs by clicking **Add new** in the Actions column for the input.
4. Follow the instructions in the subsequent pages for the input type you select.
5. Click **Save**.

Splunk Enterprise begins collecting the data immediately in most cases.

Use configuration files to collect Windows data

In cases where you cannot use Splunk Web to create and enable data inputs, such as when you use a universal forwarder to collect the data, you must use configuration files. Using configuration files offers more control and configurability than Splunk Web does in many cases. Some inputs can only be configured using configuration files.

Note: The universal forwarder installer on Windows lets you configure some Windows inputs at installation time.

1. From a command prompt or PowerShell window, go to the `%SPLUNK_HOME%\etc\system\local` directory.
2. Edit `inputs.conf` in this directory. You might need to create this file.
3. Add inputs to the `inputs.conf` file by defining input stanzas.
4. Save the file and close it.
5. Restart Splunk Enterprise. The software reloads the configuration files and begins collecting data based on the new configuration.

Considerations for deciding how to monitor remote Windows data

This topic discusses the considerations you must take when using Splunk Enterprise to gather remote Windows data.

Remote Windows data overview

Splunk Enterprise collects remote Windows data for indexing in one of two ways:

- from Splunk forwarders
- via Windows Management Instrumentation (WMI)

Use a forwarder to collect remote Windows data

Use a universal forwarder to get remote Windows data whenever possible. The universal forwarder has these advantages:

- It uses minimal network and disk resources on the installed machines.
- You can install it as a non-privileged user, whereas you require administrative access for WMI.
- If you install it as the Local System user, then it has administrative access to the machine and requires no authentication to get data from there, as WMI does.
- It scales well in large environments and is easy to install. You can install it manually, with either a Microsoft deployment tool like System Center Configuration Manager (SCCM) or a third party distribution solution such as Puppet or IBM BigFix.

After you install a universal forwarder, it gathers information locally and sends it to a Splunk Enterprise indexer. You tell the forwarder what data to gather either during the installation process or later, by distributing configuration updates manually or with a **deployment server**. You can also install add-ons into the universal forwarder.

There are some drawbacks to using the universal forwarder, depending on your network configuration and layout. See "Forwarders versus remote collection through WMI" in this topic.

Use WMI to collect remote Windows data

The Windows Management Instrumentation (WMI) framework lets Splunk Enterprise collect virtually any kind of data from remote Windows machines. In this configuration, Splunk Enterprise runs as a user that you specify at installation (or later on, in the Services control panel).

This configuration:

- Gives Splunk Enterprise as much access to the network as the specified account has for remote access.
- Lets indexers collect data from remote Windows machines across the enterprise and place that data into a central repository.
- Is ideal for small to medium-sized networks with at least one indexer in each network segment

There are some caveats to this method of collection. See [Forwarders versus WMI](#) in this topic.

Also, while Active Directory (AD) monitoring does not use WMI, it has the same authentication considerations as data inputs that do use it. For information on how Splunk Enterprise monitors AD, see [Monitor Active Directory](#) in this manual.

Considerations for getting data over WMI

When collecting remote Windows data over WMI, consider the following:

Authentication for remote Windows data

Windows requires authentication for remote operations. Failure to understand how Splunk Enterprise interacts with Windows over the network can lead to suboptimal search results, or none at all. This section provides guidelines on security for collecting remote Windows data.

When you install Splunk Enterprise, you can specify that it run as the Local System user, or another user. This choice has ramifications for both installation and data collection.

The user you tell Splunk Enterprise to run as determines the kind of data it can retrieve from remote machines. To get the data you want, you must provide an appropriate level of permission to this user.

In most cases, configure the Splunk Enterprise user account with "least-permissive" access to the data sources you want to collect. This entails:

- Adding the user to various domain security groups.
- Making changes to the access control lists of various AD objects, depending on the data sources you need to access.

If your AD domain security policy enforces password changes regularly, you must also:

- Confirm that either the Splunk Enterprise user password never expires, or that you manually change the password before it expires, as defined by the password policy.
- Restart Splunk services that run as that account on all hosts in your network, once you change the password.

You should also assign the Splunk Enterprise account the "Deny log on locally" user rights assignment in Local Security Policy to prevent the user from logging in interactively to workstations. This method gives you more control and is more secure than handing out domain administrator access.

Individual Getting Data In topics in this manual that deal with remote access to Windows machines contain additional information and recommendations on how to configure the user Splunk Enterprise runs as for least-permissive access. Review the "Security and remote access considerations" section on those pages.

Use managed system accounts to access Windows data

On recent versions of Windows Server, you can use managed service accounts (MSAs) to address challenges with password expiry. See Managed service accounts on Windows Server 2008 and Windows 7 in the *Installation* manual.

Network and I/O usage considerations

Monitor network bandwidth usage closely, especially in networks with slow or thin WAN links. For this reason alone, universal forwarders are a better choice for large remote data collection operations.

Disk bandwidth is a concern as well. Anti-virus scanner drivers and drivers that intermediate between Splunk Enterprise and the operating system should always be configured to ignore the Splunk Enterprise directory and processes, regardless of the type of installation.

Splunk forwarders versus WMI

Use a universal forwarder to get data in from a remote Windows host. A universal forwarder offers the most types of data sources, provides more detailed data (for example, in performance monitoring metrics), minimizes network overhead, and reduces operational risk and complexity. It is also more scalable than WMI in many cases.

In circumstances where you collect data remotely (such as when corporate or security policy restricts code installation, or there are performance or interoperability concerns,) you can use the native WMI interface to collect event logs and performance data.

These are the main areas of tradeoff between WMI and forwarders:

- Performance
- Deployment
- Management

Performance

With respect to performance, a forwarder is a better choice when:

- You collect local event logs or flat files. A forwarder requires less CPU and performs basic precompression of the data in an effort to reduce network overhead.
- You want to collect data from a machine without having to worry about authentication. When you install a forwarder as the Local System user, it has administrative access to the machine, letting you collect any data from it.
- You want to collect data from busy hosts such as AD domain controllers or machines that consistently experience periods of high utilization, such as

Exchange, SQL Server/Oracle, VMWare, Hyper-V, or SharePoint servers. This is because WMI might have problems keeping up with the amount of data these services generate. WMI polling is best-effort by design, and Splunk Enterprise also throttles WMI calls to prevent unintentional denial-of-service attacks.

- You are concerned about CPU and network utilization. Forwarders use as little of these resources as possible, while WMI uses more CPU and network resources to transfer data.
- You are concerned about scalability. Universal forwarders scale very well. Heavy forwarders do not scale as well as universal forwarders, but both types of forwarder scale considerably better than WMI.

WMI is a better choice when you have concerns about memory usage on a system with high memory utilization. Because forwarders have more polling options available, and reside on the local machine while collecting data, they use more memory than WMI does.

Deployment

A forwarder is a better choice for deployment when:

- You have control of the base build of the OS, as is the case when you create system images.
- You have many data sources to collect, particularly if the data requires transformation of any kind.

Note: Except for a few cases, you cannot use a universal forwarder to process data before it reaches the indexer. If you need to make any changes to your data before you index it, you must use a heavy forwarder.

WMI is a better choice when:

- You don't have control of the base OS build, or you don't have domain administrator access, or local administrator privileges on the machines from which you want to get data.
- You want or need only a limited set of data from a large number of hosts (for example, CPU data for usage billing).

A common deployment scenario is to first test using remote polling, then add successful or useful data inputs to your forwarder configurations later, or when you do large scale forwarder installations.

Management

Both mechanisms offer logging and alerting to advise if a host comes on or offline or is unreachable. To prevent an unintentional denial of service attack, the WMI polling service in Splunk Enterprise polls less frequently over time if it cannot contact a host, and eventually stops polling unreachable hosts altogether. Do not use remote polling over WMI for machines that are frequently offline, such as laptops or dynamically provisioned virtual machines.

The table shows a list of data sources and indicates which data collection type(s) are appropriate for each data source.

Data sources and collection methods

Data source	Local forwarder	WMI
Event logs	Yes	Yes*
Performance	Yes	Yes
Registry	Yes	No
Active Directory	Yes	No
Log files	Yes	Yes**
Crawl	Yes	No

** For remote event log collection, you must know the name of the event log you want to collect. On local forwarders, you have the option to collect all logs, regardless of name.*

*** Splunk Enterprise supports remote log file collection using the "\\SERVERNAME\SHARE" syntax; however, you must use CIFS (Common Internet File System, or Server Message Block) as your application layer file access protocol, and Splunk Enterprise must have at least read access to both the share and the underlying file system.*

Search Windows data on a non-Windows instance of Splunk Enterprise

You can index and search your Windows data on a non-Windows instance of Splunk, but you must first use a Windows instance of Splunk Enterprise to get the Windows data. You can do this by installing a Splunk forwarder onto the Windows computer and configuring it to forward Windows data to the non-Windows instance of Splunk Enterprise.

There are two ways to proceed:

- Set up forwarders locally on each Windows machine that you want data. These forwarders can send the Windows data to the non-Windows receiving instance.
- Set up a forwarder on a separate Windows machine. The forwarder can use WMI to collect data from all the Windows machines in the environment and then forward the combined data to a non-Windows receiving instance of Splunk.

Monitor Active Directory

Active Directory (AD) is an integral part of any Windows network. The Active Directory database (known as the NT Directory Service (NTDS) database) is the central repository for user, computer, network, device and security objects in an AD domain or forest. When you make a change to Active Directory, such as adding or deleting a user, member server or domain controller, those changes are recordable. Splunk Enterprise lets you alert and monitor those changes in real time.

You can configure AD monitoring to watch changes to your Active Directory forest, and collect user and machine metadata. You can use this feature combined with dynamic list lookups to decorate or modify events with any information available in AD.

After you have configured Splunk to monitor your Active Directory, it takes a baseline snapshot of the AD schema. It uses this snapshot to establish a starting point against which to monitor.

The AD monitoring input runs as a separate process called `splunk-admon.exe`. It runs once for every Active Directory monitoring input defined in Splunk.

Why monitor Active Directory?

If you are charged with maintaining the integrity, security and health of your Active Directory, then you are concerned with what is happening with it day to day. Splunk Enterprise lets you see what has changed in your AD, who or what made the changes, and when they were made.

You can transform this data into reports for corporate security compliance or forensics. You can also use the data retrieved for intrusion alerts for immediate

response. Additionally, you can create health reports with the data indexed for future AD infrastructure planning activities, such as assignment of operations master roles, AD replicas, and global catalogs across domain controllers (DCs).

What do you need to monitor Active Directory?

The following table lists the explicit permissions you need to monitor an Active Directory schema.

Activity	Required permissions
Monitor an Active Directory schema	<ul style="list-style-type: none">* Splunk Enterprise must run on Windows* Splunk Enterprise must run as a domain user* The user Splunk Enterprise runs as must have read access to all AD objects that you want to monitor

Considerations for monitoring Active Directory

To get the best results out of monitoring AD with Splunk Enterprise, understand the following:

- This feature is only available with Splunk Enterprise on Windows. You cannot monitor AD changes from a *nix version of Splunk Enterprise. You can forward AD data gathered from a Windows version of Splunk Enterprise to a *nix indexer).
- The AD monitoring process can run under a full Splunk instance or within any kind of forwarder.
- The machine that monitors changes to AD must belong to the domain or forest you want to monitor.
- The user Splunk runs as must be part of the domain too. The permissions that the user has determine which parts of AD Splunk can monitor.

For more information, see [Considerations for deciding how to monitor remote Windows data](#) in this manual. For information on deciding which user Splunk should run as at installation time, see Choose the user Splunk should run as in the *Installation* manual.

Configure Active Directory monitoring

You can configure AD monitoring either in Splunk Web or by editing configuration files. More options, such as the ability to configure monitors for multiple DCs, are available when using configuration files.

Configure AD 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 **Active Directory 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 Active Directory on the local Windows machine.

Select the input source

1. In the left pane, locate and select **Active Directory monitoring**.
2. In the **Collection name** field, type in a unique name for the input that you will remember.
3. Optionally, in the **Target domain controller** field, enter the host name or IP address of the domain controller you want to use to monitor AD.
4. Optionally, in the **Starting node** field, type in the Active Directory node you would like the input to begin monitoring from. You must specify the Lightweight Directory Access Protocol format, for example, `DC=Splunk-Docs,DC=com`.

You can click the **Browse** button to browse through a list of available Active Directory nodes to browse through a list of available AD domains.

5. Check the 'Monitor Subtree' button if you want Splunk Enterprise to monitor all sub-nodes of the node you entered in the "Starting node" field.

6. 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.

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"](#).

Note: **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 the green **Review** button.

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 Active Directory node.

Configure AD monitoring with configuration files

The `inputs.conf` configuration file controls Active Directory monitoring configurations. Edit copies of `inputs.conf` in the

`%SPLUNK_HOME%\etc\system\local` directory. If you edit them in the default directory, Splunk overwrites any changes you make when you upgrade. For more information about configuration file precedence, see "Configuration file precedence" in this manual.

1. Open `%SPLUNK_HOME%\etc\system\local\inputs.conf` for editing. You might need to create this file if it does not exist.

2. Add the appropriate AD monitoring stanzas and settings.

By default, when you enable AD monitoring inputs, Splunk gathers AD change data from the first domain controller that it can attach to. If that is acceptable, no further configuration is necessary.

inputs.conf settings

`inputs.conf` contains one stanza for each AD monitoring input, with a header like the following:

```
[admon://<name of stanza>]
```

In each stanza, you can specify:

Attribute	Required?	Description	Default
<code>targetDc</code>	Yes	<p>The unique name of the domain controller you want Splunk to use for AD monitoring.</p> <p>Specify a unique name for this attribute if:</p> <ul style="list-style-type: none">• You have a very large AD and you only want to monitor information in a particular Organizational Unit (OU), subdomain, etc.• You have a specific (read-only) domain controller that can be used for monitoring purposes in a high security environment.• You have multiple domains or forests in with transitive trusts established, and want to target a different tree than the one where the host that runs Splunk Enterprise resides.• You want to configure multiple AD monitoring inputs to target multiple domain	n/a

		<p>controllers. For example, to monitor AD replication across a distributed environment.</p> <p>To target multiple DCs, add another <code>[admon://<uniquename>targetDc]</code> stanza for a target in that tree.</p>	
<code>startingNode</code>	No	<p>A fully qualified Lightweight Directory Access Protocol (LDAP) name (for example: <code>"LDAP://OU=Computers,DC=ad,DC=splunk,DC=com"</code>) that specifies where in the AD tree that Splunk Enterprise should begin its indexing. The software starts there and enumerates down to sub-containers, depending on the configuration of the <code>monitorSubtree</code> attribute.</p> <p>The value of <code>startingNode</code> must be within the scope of the DC you are targeting for Splunk to get AD data.</p>	The highest root domain in the tree that Splunk Enterprise can access.
<code>monitorSubtree</code>	No	How much of the target AD container to index. A value of 0 means to index only the target container, and not traverse into subcontainers within that container. A value of 1 means to enumerate all sub-containers and domains that it has access to.	1 (monitor all domains that Splunk Enterprise has access to)
<code>baseline</code>	No	Whether or not the input enumerates all existing available AD objects when it first runs. A value of 0 means not to set a baseline. A value of 1 means to set a baseline.	1 (set the baseline.)
<code>index</code>	No	The index to route AD monitoring data to.	the 'default' index.
<code>disabled</code>	No	Whether or not the Splunk should run the input. A value of 0 tells Splunk that the input is enabled, and a value of 1 tells Splunk that the input is disabled.	0 (enabled).

Example AD monitoring configurations

The following are examples of how to use `inputs.conf` to monitor desired portions of your AD network.

To index data from the top of the AD directory:

```
#Gather all AD data that this server can see
```

```
[admon://NearestDC]
targetDc =
startingNode =
```

To use a DC that is at a higher root level than an OU you want to target for monitoring:

```
# Use the pri01.eng.ad.splunk.com domain controller to get all AD
metadata for
# the Computers OU in this forest. We want schema data for the entire AD
tree, not
# just this node.
```

```
[admon://DefaultTargetDc]
targetDc = pri01.eng.ad.splunk.com
startingNode = OU=Computers,DC=eng,DC=ad,DC=splunk,DC=com
```

To monitor multiple domain controllers:

```
# Get change data from two domain controllers (pri01 and pri02) in the
same AD tree.
# Index both and compare/contrast to ensure AD replication is occurring
properly.
```

```
[admon://DefaultTargetDc]
targetDc = pri01.eng.ad.splunk.com
startingNode = OU=Computers,DC=eng,DC=ad,DC=splunk,DC=com
```

```
[admon://SecondTargetDc]
targetDc = pri02.eng.ad.splunk.com
startingNode = OU=Computers,DC=eng,DC=ad,DC=splunk,DC=com
```

Sample AD monitoring output

When the Splunk AD monitoring utility runs, it gathers AD change events. Each change event is indexed as an event in Splunk Enterprise. You can view these events as they come into Splunk in the Search app.

There are several types of AD change events that Splunk Enterprise can index. Examples of these events follow. Some of the content of these events has been obscured or altered for publication purposes.

Update event

When an AD object changes, Splunk Enterprise generates this type of event. The software logs this change as type `admonEventType=Update`.

2/1/10
3:17:18.009 PM

02/01/2010 15:17:18.0099
dcName=stuff.splunk.com
admonEventType=Update

Names:

objectCategory=CN=Computer,CN=Schema,CN=Configuration
name=stuff2
displayName=stuff2
distinguishedName=CN=stuff2,CN=Computers

Object Details:

sAMAccountType=805306369
sAMAccountName=stuff2
logonCount=4216
accountExpires=9223372036854775807
objectSid=S-1-5-21-3436176729-1841096389-3700143990-1190
primaryGroupID=515
pwdLastSet=06:30:13 pm, Sat 11/27/2010
lastLogon=06:19:43 am, Sun 11/28/2010
lastLogoff=0
badPasswordTime=0
countryCode=0
codePage=0
badPwdCount=0
userAccountControl=4096
objectGUID=blah
whenChanged=01:02.11 am, Thu 01/28/2010
whenCreated=05:29.50 pm, Tue 11/25/2008
objectClass=top|person|organizationalPerson|user|computer

Event Details:

uSNChanged=2921916
uSNCreated=1679623
instanceType=4

Additional Details:

isCriticalSystemObject=FALSE
servicePrincipalName=TERMSRV/stuff2|TERMSRV blah
dNSHostName=stuff2.splunk.com
operatingSystemServicePack=Service Pack 2
operatingSystemVersion=6.0 (6002)

```
operatingSystem=Windows Vista? Ultimate
localPolicyFlags=0
```

Delete event

Splunk Enterprise generates this event type when an AD object has been marked for deletion. The event type is similar to `admonEventType=Update`, except that it contains the `isDeleted=True` key/value pair at the end of the event.

```
2/1/10
3:11:16.095 PM
```

```
02/01/2010 15:11:16.0954
dcName=stuff.splunk.com
admonEventType=Update
Names:
```

```
name=SplunkTest
```

```
DEL:blah
```

```
distinguishedName=OU=SplunkTest\0ADEL:blah,CN=Deleted
```

```
Objects
```

```
DEL:blah
```

```
Object Details:
```

```
objectGUID=blah
whenChanged=11:31.13 pm, Thu 01/28/2010
whenCreated=11:27.12 pm, Thu 01/28/2010
objectClass=top|organizationalUnit
```

```
Event Details:
```

```
uSNChanged=2922895
uSNCreated=2922846
instanceType=4
```

```
Additional Details:
```

```
dScorePropagationData=20100128233113.0Z|20100128233113.0Z|2010012823311
lastKnownParent=stuff
'''isDeleted=TRUE'''
```

Sync event

When AD monitoring inputs are configured, Splunk Enterprise tries to capture a baseline of AD metadata when it starts. Splunk Enterprise generates event type `admonEventType=Sync`, which represents the instance of one AD object and all its field values. Splunk Enterprise tries to capture all of the objects from the last recorded Update Sequence Number (USN).

When you restart either Splunk Enterprise or the `splunk-admon.exe` process, the software logs an extra 'sync' event. This is normal.

2/1/10
3:11:09.074 PM

02/01/2010 15:11:09.0748

dcName=ftw.ad.splunk.com

admonEventType=Sync

Names:

name=NTDS Settings
distinguishedName=CN=NTDS
Settings,CN=stuff,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration
cn=NTDS Settings
objectCategory=CN=NTDS-DSA,CN=Schema,CN=Configuration,DC=ad,DC=splunk,DC=com
fullPath=LDAP://stuff.splunk.com/<GUID=bla bla bla>
CN=NTDS Settings

Object Details:

whenCreated=10:15.04 pm, Tue 02/12/2008
whenChanged=10:23.00 pm, Tue 02/12/2008
objectGUID=bla bla bla
objectClass=top,applicationSettings,nTDSDSA
classPath=nTDSDSA

Event Details:

instanceType=4

Additional Details:

systemFlags=33554432
showInAdvancedViewOnly=TRUE
serverReferenceBL=CN=stuff,CN=Domain System Volume
(SYSVOL share),CN=File Replication Service,CN=System
options=1
msDS-hasMasterNCs=DC=ForestDnsZones|DC=DomainDnsZones|CN=Schema,CN=Configuration
msDS-HasInstantiatedNCs=
msDS-HasDomainNCs=blah
msDS-Behavior-Version=2
invocationId=bla bla bla
hasMasterNCs=CN=Schema,CN=Configuration|CN=Configuration
dScorePropagationData=
dMDLocation=CN=Schema,CN=Configuration
nTSecurityDescriptor=NT AUTHORITY\Authenticated Users
SchemaName=LDAP://stuff.splunk.com/schema/nTDSDSA

Schema event

When you restart Splunk Enterprise after configuring it for AD monitoring, it generates a schema type event: `admonEventType=schema`. This event shows the definitions of every object in the Active Directory structure. The available, required and optional fields are listed for each AD object. Failure to see all of these fields can indicate a problem with Active Directory.

02/01/2010 15:11:16.0518

dcName=LDAP://stuff.splunk.com/
admonEventType=schema
className=msExchProtocolCfgSMTPIPAddress
classCN=ms-Exch-Protocol-Cfg-SMTP-IP-Address
instanceType=MandatoryProperties
nTSecurityDescriptor=MandatoryProperties
objectCategory=MandatoryProperties
objectClass=MandatoryProperties
adminDescription=OptionalProperties
adminDisplayName=OptionalProperties
allowedAttributes=OptionalProperties
allowedAttributesEffective=OptionalProperties
allowedChildClasses=OptionalProperties
allowedChildClassesEffective=OptionalProperties
bridgeheadServerListBL=OptionalProperties
canonicalName=OptionalProperties
cn=OptionalProperties
createTimeStamp=OptionalProperties
description=OptionalProperties
directReports=OptionalProperties
displayName=OptionalProperties
displayNamePrintable=OptionalProperties
distinguishedName=OptionalProperties
dSASignature=OptionalProperties
dSCorePropagationData=OptionalProperties
extensionName=OptionalProperties
flags=OptionalProperties
fromEntry=OptionalProperties
frsComputerReferenceBL=OptionalProperties
fRSMemberReferenceBL=OptionalProperties
fSMORoleOwner=OptionalProperties
heuristics=OptionalProperties
isCriticalSystemObject=OptionalProperties
isDeleted=OptionalProperties
isPrivilegeHolder=OptionalProperties
lastKnownParent=OptionalProperties
legacyExchangeDN=OptionalProperties
managedObjects=OptionalProperties
masteredBy=OptionalProperties
memberOf=OptionalProperties
modifyTimeStamp=OptionalProperties
mS-DS-ConsistencyChildCount=OptionalProperties
mS-DS-ConsistencyGuid=OptionalProperties
msCOM-PartitionSetLink=OptionalProperties
msCOM-UserLink=OptionalProperties
msDFSR-ComputerReferenceBL=OptionalProperties
msDFSR-MemberReferenceBL=OptionalProperties
msDS-Approx-Immed-Subordinates=OptionalProperties
msDs-masteredBy=OptionalProperties
msDS-MembersForAzRoleBL=OptionalProperties
msDS-NCReplCursors=OptionalProperties
msDS-NCReplInboundNeighbors=OptionalProperties

msDS-NCReplOutboundNeighbors=OptionalProperties
msDS-NonMembersBL=OptionalProperties
msDS-ObjectReferenceBL=OptionalProperties
msDS-OperationsForAzRoleBL=OptionalProperties
msDS-OperationsForAzTaskBL=OptionalProperties
msDS-ReplAttributeMetaData=OptionalProperties
msDS-ReplValueMetaData=OptionalProperties
msDS-TasksForAzRoleBL=OptionalProperties
msDS-TasksForAzTaskBL=OptionalProperties
msExchADCGlobalNames=OptionalProperties
msExchALObjectVersion=OptionalProperties
msExchHideFromAddressLists=OptionalProperties
msExchInconsistentState=OptionalProperties
msExchIPAddress=OptionalProperties
msExchTurfList=OptionalProperties
msExchUnmergedAttsPt=OptionalProperties
msExchVersion=OptionalProperties
msSFU30PosixMemberOf=OptionalProperties
name=OptionalProperties
netbootSCPBL=OptionalProperties
nonSecurityMemberBL=OptionalProperties
objectGUID=OptionalProperties
objectVersion=OptionalProperties
otherWellKnownObjects=OptionalProperties
ownerBL=OptionalProperties
partialAttributeDeletionList=OptionalProperties
partialAttributeSet=OptionalProperties
possibleInferiors=OptionalProperties
proxiedObjectName=OptionalProperties
proxyAddresses=OptionalProperties
queryPolicyBL=OptionalProperties
replicatedObjectVersion=OptionalProperties
replicationSignature=OptionalProperties
replPropertyMetaData=OptionalProperties
replUpToDateVector=OptionalProperties
repsFrom=OptionalProperties
repsTo=OptionalProperties
revision=OptionalProperties
sDRightsEffective=OptionalProperties
serverReferenceBL=OptionalProperties
showInAddressBook=OptionalProperties
showInAdvancedViewOnly=OptionalProperties
siteObjectBL=OptionalProperties
structuralObjectClass=OptionalProperties
subRefs=OptionalProperties
subSchemaSubEntry=OptionalProperties
systemFlags=OptionalProperties
unmergedAtts=OptionalProperties
url=OptionalProperties
uSNChanged=OptionalProperties
uSNCreated=OptionalProperties
uSNSALastObjRemoved=OptionalProperties

```
USNIntersite=OptionalProperties
uSNLastObjRem=OptionalProperties
uSNSource=OptionalProperties
wbemPath=OptionalProperties
wellKnownObjects=OptionalProperties
whenChanged=OptionalProperties
whenCreated=OptionalProperties
wWWHomePage=OptionalProperties
```

Answers

Have questions? Visit [Splunk Answers](#) and see what questions and answers the Splunk community has around monitoring AD with Splunk.

Monitor Windows event log data

Windows generates log data during the course of its operation. The Windows Event Log service handles nearly all of this communication. It gathers log data published by installed applications, services and system processes and places them into event log channels. Programs such as Microsoft Event Viewer subscribe to these log channels to display events that have occurred on the system.

Splunk Enterprise also supports the monitoring of Windows event log channels. It can monitor event log channels and files stored on the local machine, and it can collect logs from remote machines.

The event log monitor runs as an input processor within the `splunkd` service. It runs once for every event log input that you define in Splunk Enterprise.

Why monitor event logs?

Windows event logs are the core metric of Windows host operations - if there is a problem with your Windows system, the Event Log service has logged it. Splunk Enterprise indexing, searching, and reporting capabilities make your logs accessible.

What do you need to monitor event logs?

Activity:	Required permissions:
Monitor local event logs	Splunk Enterprise must run on Windows Splunk Enterprise must run as the Local System user to read

	all local event logs
Monitor remote event logs	<p>A universal forwarder must run on the Windows host from which you want to collect event logs</p> <p>OR</p> <p>Splunk Enterprise must run on Windows</p> <p>AND</p> <p>Splunk Enterprise must run as a domain or remote user with read access to Windows Management Instrumentation (WMI) on the target host</p> <p>AND</p> <p>The user Splunk Enterprise runs as must have read access to the desired event logs</p>

Security and remote access considerations

Splunk Enterprise collects event log data from remote machines using either WMI or a forwarder. Splunk recommends using a universal forwarder to send event log data from remote machines to an indexer. See *The universal forwarder* in the *Universal Forwarder* manual for information about how to install, configure and use the forwarder to collect event log data.

To install forwarders on your remote machines to collect event log data, you can install the forwarder as the Local System user on these machines. The Local System user has access to all data on the local machine, but not on remote machines.

To use WMI to get event log data from remote machines, you must ensure that your network and Splunk instances are properly configured. You cannot install the Splunk platform as the Local System user, and the user you install with determines the event logs Splunk software sees. See [Security and remote access considerations](#) in the [Monitor WMI-based data](#) topic in this manual for additional information on the requirements you must satisfy in order for the Splunk platform to collect remote data properly using WMI.

By default, Windows restricts access to some event logs depending on which version of Windows you run. In particular, the Security event logs by default can only be read by members of the local Administrators or global Domain Admins groups.

Collect event logs from a remote Windows machine

You have several choices to collect data from a remote Windows host:

Use a universal forwarder

You can install a universal forwarder on the Windows host and instruct it to collect event logs. You can do this manually, or use a deployment server to manage the forwarder configuration.

For specific instructions to install the universal forwarder, see [Install a Windows universal forwarder from an installer in the Universal Forwarder manual](#).

1. On the Windows host that you want to collect Windows Event Logs, download the universal forwarder software from Splunk.
2. Run the universal forwarder installation package to begin the installation process.
3. When the installer prompts you, configure a receiving indexer.
4. When the installer prompts you to specify inputs, enable the event log inputs by checking the "Event logs" checkbox.
5. Complete the installation procedure.
6. On the receiving indexer, use Splunk Web to search for the event log data. An example search string follows:

```
host=<name of remote Windows host> sourcetype=Wineventlog
```

Use WMI

If you choose to collect event logs using WMI, you must install Splunk Enterprise with an Active Directory domain user. If the selected domain user is not a member of the Administrators or Domain Admins groups, then you must configure event log security to give the domain user access to the event logs.

To change event log security for access to the event logs from remote machines, you must:

- Have administrator access to the host from which you are collecting event logs.
- Understand how the Security Description Definition Language (SDDL) ([external link](#)) works, and how to assign permissions with it.

If you run Windows Vista, Windows 7, Windows Server 2008/2008 R2 or Server 2012 R2, use the `wevtutil` utility to set event log security.

See [Considerations for deciding how to monitor remote Windows data](#) for information on collecting data from remote Windows machines.

1. Download Splunk Enterprise instance onto a Windows host.
2. Double-click the installer file to begin the installation.
3. When the installer prompts you to specify a user, choose **Domain user**.
4. On the next installer pane, enter the domain user name and password that Splunk Enterprise should use when it runs.
5. Follow the prompts to complete installation of the software.
6. Once the software has installed, log into the instance.
7. Use Splunk Web to add the remote event log input, as described in [Configure remote event log monitoring](#).

Anomalous host names visible in event logs on some systems

On Windows Vista and Server 2008 R2 systems, you might see some event logs with randomly-generated host names. This is the result of those systems logging events before the user has named the system, during the OS installation process.

This anomaly occurs only when you collect logs from the above-mentioned versions of Windows remotely over WMI.

Use Splunk Web to configure event log monitoring

To get local Windows event log data, point your Splunk instance at your Event Log service:

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 event log collection**.

4. Click **New** to add an input.

By Splunk Home:

1. Click the **Add Data** link in Splunk Home.
2. Click **Monitor** to monitor Event Log data on the local Windows machine, or **Forward** to forward Event Log data from another Windows machine. Splunk Enterprise loads the "Add Data - Select Source" page.
3. If you selected **Forward**, choose or create the group of forwarders you want this input to apply to. See "[Forward data](#)" in this manual.
4. Click **Next**.

Select the input source

1. In the left pane, select **Local Event Logs**
2. In the **Select Event Logs** list box, choose the Event Log channels you want this input to monitor.
3. Click once on each Event Log channel you want to monitor. Splunk Enterprise moves the channel from the "Available items" window to the "Selected items" window.
4. To unselect a channel, click on its name in the "Available Items" window. Splunk Enterprise moves the channel from the "Selected items" window to the "Available items" window.
5. To select or unselect all of the event logs, click on the "add all" or "remove all" links. **Important:** Selecting all of the channels can result in the indexing of a lot of data, possibly more than your license allows.
6. 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"](#).

Note: **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 you specify 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 Event Log channels.

Configure remote event log monitoring

The process for configuring remote event log monitoring is nearly identical to the process for monitoring local event logs.

1. Follow the instructions to get to the Add New page, as described in [Go to the Add New page](#).

2. In the left pane, locate and select **Remote Event Logs**.

3. In the **Event Log collection name** field, enter a unique name for this input that you will remember.

4. In the **Choose logs from this host** field, enter the host name or IP address of the machine that contains the Event Log channels you want to monitor.

5. Click the **Find logs** button to refresh the page with a list of available Event Log channels on the host you entered.

6. Click once on each Event Log channel you want to monitor. Splunk Enterprise moves the channel from the "Available items" window to the "Selected items" window.

7. To unselect a channel, click on its name in the "Available Items" window. Splunk Enterprise moves the channel from the "Selected items" window to the "Available items" window.

8. To select or unselect all of the event logs, click on the "add all" or "remove all" links.

Caution: Selecting all of the channels can result in the indexing of a lot of data, possibly more than your Splunk license can support.

9. In the **Collect the same set of logs from additional hosts** field, enter host names or IP addresses of additional machines that contain the Event Logs you selected previously. Separate multiple hosts with commas.

10. Click the green **Next** button.

11. Follow the instructions to specify input settings, as described in "[Specify input settings](#)."

12. Follow the instructions to review your choices, as described in "[Review your choices](#)."

Use inputs.conf to configure event log monitoring

Edit `inputs.conf` to configure event log monitoring. For information on configuring data inputs with `inputs.conf`, see "[Configure your inputs](#)" in this manual.

Note: You can always review the defaults for a configuration file by looking at the examples in `%SPLUNK_HOME%\etc\system\default` or at the spec file in the Admin Manual.

To enable event log inputs by editing `inputs.conf`:

1. Using Notepad or a similar editor, open

`%SPLUNK_HOME%\etc\system\local\inputs.conf` for editing. You might need to

create this file if it does not exist.

2. Enable Windows event log inputs by adding input stanzas that reference Event Log channels.

3. Save the file and close it.

4. Restart Splunk Enterprise.

The next section describes the available configuration values for event log monitoring.

Event log monitor configuration values

Windows event log (*.evt) files are in binary format. You cannot monitor them like you do a normal text file. The `splunkd` service monitors these binary files by using the appropriate APIs to read and index the data within the files.

Splunk Enterprise uses the following stanzas in `inputs.conf` to monitor the default Windows event logs:

```
# Windows platform specific input processor.
[WinEventLog://Application]
disabled = 0
[WinEventLog://Security]
disabled = 0
[WinEventLog://System]
disabled = 0
```

Monitor non-default Windows event logs

You can also configure Splunk Enterprise to monitor non-default Windows event logs. Before you can do this, you must import them to the Windows Event Viewer. After you import the logs, you can add them to your local copy of `inputs.conf`, as follows:

```
[WinEventLog://DNS Server]
disabled = 0
[WinEventLog://Directory Service]
disabled = 0
[WinEventLog://File Replication Service]
disabled = 0
```

Use the "Full Name" log property in Event Viewer to specify complex Event Log channel names properly

You can use the "Full Name" Event Log property in Event Viewer to ensure that you specify the correct Event Log channel in an `inputs.conf` stanza.

For example, to monitor the Task Scheduler application log (Microsoft-Windows-TaskScheduler-Operational):

1. Launch Event Viewer.
2. Expand Applications and Services Logs > Microsoft > Windows > TaskScheduler.
3. Right-click `Operational` and select **Properties**.
4. In the dialog that appears, copy the text in the "Full Name" field.
5. Append this text into the `WinEventLog://` stanza:

```
[WinEventLog://Microsoft-Windows-TaskScheduler/Operational]
disabled = 0
```

Disable an event log stanza

To disable indexing for an event log, add `disabled = 1` below its listing in the stanza in `%SPLUNK_HOME%\etc\system\local\inputs.conf`.

Splunk Enterprise uses the following attributes in `inputs.conf` to monitor Event Log files:

Attribute	Description	Default
<code>start_from</code>	<p>How Splunk Enterprise should read events chronologically. Acceptable values are <code>oldest</code> (meaning read logs from the oldest to the newest) and <code>newest</code> (meaning read logs from the newest to the oldest.)</p> <p>If you set the attribute to <code>newest</code>, Splunk Enterprise reads logs from the most recent to the oldest, then stops.</p>	<code>oldest</code>

	You cannot set this attribute to <code>newest</code> while also setting the <code>current_only</code> attribute to 1. Splunk Enterprise ignores this combination.	
<code>current_only</code>	<p>How Splunk Enterprise should index events after it starts. Acceptable values are 1 (where the input acquires events that arrive after the input starts for the first time, like 'tail -f' on *nix systems) or 0 (where the input gets all existing events in the log and then continues to monitor incoming events in real time.)</p> <p>You cannot set this attribute to 1 and also set the <code>start_from</code> attribute to <code>newest</code>. Splunk Enterprise ignores this combination.</p>	0
<code>checkpointInterval</code>	<p>How frequently, in seconds, that the Windows Event Log input should save a checkpoint.</p> <p>Checkpoints store the eventID of acquired events. This lets Splunk Enterprise continue monitoring at the correct event after a shutdown or outage.</p>	5
<code>evt_resolve_ad_ds</code>	<p>The domain controller Splunk Enterprise should use when it interacts with Active Directory while indexing Windows Event Log channels. Valid only when you set the <code>evt_resolve_ad_obj</code> attribute to 1 and do not set the <code>evt_dc_name</code> attribute.</p> <p>Valid values are <code>auto</code> (meaning choose the nearest domain controller to bind to for AD object resolution) or <code>PDC</code> (meaning bind to the primary domain controller for the AD site that the host is in.) If you also set the <code>evt_dc_name</code> attribute, Splunk Enterprise ignores this attribute.</p>	<code>auto</code>
<code>evt_resolve_ad_obj</code>	How Splunk Enterprise should interact with Active Directory while indexing Windows Event Log channels. Valid values are 1 (meaning resolve Active Directory objects like Globally Unique Identifier (GUID) and Security Identifier (SID) objects to their canonical names for a	0

	<p>specific Windows event log channel) and 0 (meaning not to attempt any resolution.)</p> <p>When you set this value to 1, you can optionally specify the Domain Controller name and/or DNS name of the domain to bind to, which Splunk Enterprise will then use to resolve the AD objects. If you do not set this value, Splunk attempts to resolve the AD objects.</p>	
evt_dc_name	<p>Which Active Directory domain controller Splunk Enterprise should bind to in order to resolve AD objects. This name can be the NetBIOS name of the domain controller, the fully-qualified DNS name of the domain controller, or an environment variable name, specified as <code>\$Environment_variable</code>.</p> <p>If you set this attribute, then Splunk Enterprise ignores the <code>evt_resolve_ad_ds</code> attribute, which controls how the software determines the best domain controller to bind to for AD object resolution.</p> <p>If you specify an environment variable, you must prepend a dollar sign (\$) to the environment variable name. Splunk Enterprise then uses the specified environment variable as the domain controller to connect to for AD object resolution. For example, to use the <code>%LOGONSERVER%</code> variable, specify <code>evt_dc_name = \$logonserver</code>.</p> <p>You can precede either format with two backslash characters. This attribute does not have a default.</p>	N/A
evt_dns_name	The fully-qualified DNS name of the domain that Splunk Enterprise should bind to in order to resolve AD objects.	N/A
suppress_text	Whether or not Splunk Enterprise should include the message text that comes with a	0

	security event. A value of 1 suppresses the message text, and a value of 0 preserves the text.	
<code>whitelist</code>	<p>Whether or not to index events that match the specified text string. This attribute is optional.</p> <p>You can specify one of two formats:</p> <ul style="list-style-type: none"> • One or more Event Log event codes or event IDs (Event Code/ID format.) • One or more sets of keys and regular expressions (Advanced filtering format.) <p>You cannot mix formats in a single entry. You also cannot mix formats in the same stanza.</p> <p>Splunk Enterprise processes whitelists first, then blacklists. If no whitelist is present, Splunk Enterprise indexes all events.</p> <p>When you use the Event Code/ID format:</p> <ul style="list-style-type: none"> • For multiple codes/IDs, separate the list with commas. • For ranges, use hyphens (for example "0-1000,5000-1000"). <p>When using the advanced filtering format:</p> <ul style="list-style-type: none"> • Use '=' between the key and the regular expression that represents your filter (for example "whitelist = EventCode=%^1([8-9])\$%") • You can have multiple key/regular expression sets in a single advanced filtering entry. Splunk Enterprise conjuncts the sets logically. This means that the entry is valid only if all of the sets in the entry are true. • You can specify up to 10 whitelists per stanza by adding a number to the end of the <code>whitelist</code> attribute, for example 	N/A

	<code>whitelist1...whitelist9.</code>	
<code>blacklist</code>	<p>Do not index events that match the text string specified. This attribute is optional.</p> <p>You can specify one of two formats:</p> <ul style="list-style-type: none"> • One or more Event Log event codes or event IDs (Event Log code/ID format.) • One or more sets of keys and regular expressions. (Advanced filtering format.) <p>You cannot mix formats in a single entry. You also cannot mix formats in the same stanza.</p> <p>Splunk Enterprise processes whitelists first, then processes any blacklists. If no blacklist is present, Splunk Enterprise indexes all events.</p> <p>When using the Event Log code/ID format:</p> <ul style="list-style-type: none"> • For multiple codes/IDs, separate the list with commas. • For ranges, use hyphens (for example "0-1000,5000-1000"). <p>When using the advanced filtering format:</p> <ul style="list-style-type: none"> • Use '=' between the key and the regular expression that represents your filter (for example "blacklist = EventCode=%^1([8-9])\$%") • You can have multiple key/regular expression sets in a single advanced filtering entry. Splunk Enterprise conjuncts the sets logically. This means that the entry is valid only if all of the sets in the entry are true. • You can specify up to 10 blacklists per stanza by adding a number to the end of the <code>blacklist</code> attribute, for example <code>blacklist1...blacklist9.</code> 	
<code>renderXml</code>		0 (false)

	<p>Render event data as XML supplied by the Windows Event Log subsystem. This attribute is optional.</p> <p>A value of '1' or 'true' means to render the events as XML. A value of '0' or 'false' means to render the events as plain text.</p>	
index	The index that this input should send the data to.	the default index
disabled	<p>Whether or not the input should run.</p> <ul style="list-style-type: none"> Valid values are 0 (meaning that the input should run) and 1 (meaning that the input should not run). 	0

Use the Security event log to monitor changes to files

You can monitor changes to files on your system by enabling security auditing on a set of files and/or directories and then monitoring the Security event log channel for change events. The event log monitoring input includes three attributes which you can use in `inputs.conf`. For example:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
# only index events with these event IDs.
whitelist = 0-2000,3001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

To enable security auditing for a set of files or directories, read "Auditing Security Events How To"

(<http://technet.microsoft.com/en-us/library/cc727935%28v=ws.10%29.aspx>) on MS Technet.

You can also use the `suppress_text` attribute to include or exclude the message text that comes with a security event.

Note: When you set `suppress_text` to 1 in a Windows Event Log Security stanza, the entire message text does not get indexed. This includes any contextual information about the security event. If you need this contextual

information, do not set `suppress_text` in the stanza.

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
# suppress message text, we only want the event number.
suppress_text = 1
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

To use a specific domain controller, set the `evt_dc_name` attribute:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
evt_dc_name = boston-dc1.contoso.com
checkpointInterval = 5
# suppress message text, we only want the event number.
suppress_text = 1
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

To use the primary domain controller to resolve AD objects, set the `evt_resolve_ad_ds` attribute to `PDC`. Otherwise, it locates the nearest domain controller:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
evt_resolve_ad_ds = PDC
checkpointInterval = 5
# suppress message text, we only want the event number.
suppress_text = 1
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

Create advanced filters with 'whitelist' and 'blacklist'

You can perform advanced filtering of incoming events with the `whitelist` and `blacklist` attributes in addition to filtering based solely on event codes. To do this, specify the key/regular expression format in the attribute:

```
whitelist = key=<regular expression> [key=<regular expression>] ...
```

In this format, `key` is a valid entry from the following list:

Key	Description
\$TimeGenerated	The time that the computer generated the event. Only generates the time string event.
\$Timestamp	The time that the event was received and recorded by the Event Log service. S Enterprise only generates the time string as the event.
Category	The category number for a specific event source.
CategoryString	A string translation of the category. The translation depends on the event source.
ComputerName	The name of the computer that generated the event.
EventCode	The event ID number for an event. Corresponds to "Event ID" in Event Viewer.
EventType	A numeric value that represents one of the five types of events that can be generated (Error, Warning, Information, Success Audit, and Failure Audit.) Available only on Windows Server 2003 and earlier or clients running Windows XP and earlier. See "Win32_NTLogEvent class (Windows)" (http://msdn.microsoft.com/en-us/library/aa394226(v=vs.85).aspx) on MSDN.
Keywords	An element used to classify different types of events within an event log channel. The Security Event Log channel has this element, for example.
LogName	The name of the Event Log channel that received the event. Corresponds to "Log Name" in Event Viewer.
Message	The text of the message in the event.
OpCode	The severity level of the event ("OpCode" in Event Viewer.)
RecordNumber	The Windows Event Log record number. Each event on a Windows host gets a unique record number. This number starts at 0 with the first event generated on the system, and increments with each new event generated, until it reached a maximum of 4294967295. It then wraps back over to 0.
Sid	The Security Identifier (SID) of the principal (such as a user, group, computer, or process) that was associated with or generated the event. See "Win32_UserAccount" (http://msdn.microsoft.com/en-us/library/windows/desktop/aa394507%28v=vs.85%29.aspx) on MSDN.

SidType	A numeric value that represents the type of SID that was associated with the event. See "Win32_UserAccount class" (http://msdn.microsoft.com/en-us/library/windows/desktop/aa394507%28v=vs.85%29.aspx) on MSDN.
SourceName	The source of the entity that generated the event ("Source" in Event Viewer)
TaskCategory	The task category of the event. Event sources let you define categories so that you can filter them with Event Viewer (using the "Task Category" field. See Event Categories (http://msdn.microsoft.com/en-us/library/aa363649%28VS.85%29.aspx) on MSDN.
Type	A numeric value that represents one of the five types of events that can be generated by Windows ("Error", "Warning", "Information", "Success Audit", and "Failure Audit".) Only available on hosts that run Windows Server 2008 or later, or Windows Vista or later. See "Win32_NTLogEvent class (Windows)" (http://msdn.microsoft.com/en-us/library/aa394226(v=vs.85).aspx) on MSDN.
User	The user associated with the event. Correlates to "User" in Event Viewer.

and `<regular expression>` is any valid regular expression that represents the filters that you want to include (when used with the `whitelist` attribute) or exclude (when used with the `blacklist` attribute).

You can specify more than one key/regular expression set on a single entry line. When you do this, Splunk Enterprise logically conjuncts the sets. This means that only events that satisfy all of the sets on the line are valid for inclusion or exclusion. For example, this entry:

```
whitelist = EventCode="^1([0-5])$" Message="^Error"
```

means to include events that have an `EventCode` ranging from 10 to 15 and contain a `Message` that begins with the word `Error`.

You can specify up to 10 separate `whitelist` or `blacklist` entries in each stanza. To do so, add a number at the end of the `whitelist` or `blacklist` entry on a separate line:

```
whitelist = key=<regular expression>
whitelist1 = key=<regular expression> key2=<regular expression 2>
whitelist2 = key=<regular expression>
```

Note: You cannot specify an entry that has more than one key/regular expression set that references the same key. If, for example, you specify:

```
whitelist = EventCode="^1([0-5])$" EventCode="^2([0-5])$"
```

Splunk Enterprise ignores the first set and only attempts to include events that match the second set. In this case, only events that contain an `EventCode` between 20 and 25 match. Events that contain an `EventCode` between 10 and 15 do not match. Only the last set in the entry ever matches. To resolve this problem, specify two separate entries in the stanza:

```
whitelist = EventCode="^1([0-5])$"
whitelist1 = EventCode="^2([0-5])$"
```

Resolve Active Directory objects in event log files

To specify whether Active Directory objects like globally unique identifiers (GUIDs) and security identifiers (SIDs) are resolved for a given Windows event log channel, use the `evt_resolve_ad_obj` attribute (1=enabled, 0=disabled) for that channel's stanza in your local copy of `inputs.conf`. The `evt_resolve_ad_obj` attribute is on by default for the Security channel.

For example:

```
[WinEventLog://Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
```

To specify a domain controller for the domain that Splunk should bind to in order to resolve AD objects, use the `evt_dc_name` attribute.

The string specified in the `evt_dc_name` attribute can represent either the domain controller's NetBIOS name, or its fully-qualified domain name (FQDN). Either name type can, optionally, be preceded by two backslash characters.

The following examples are correctly formatted domain controller names:

- FTW-DC-01
- \\FTW-DC-01
- FTW-DC-01.splunk.com
- \\FTW-DC-01.splunk.com

To specify the FQDN of the domain to bind to, use the `evt_dns_name` attribute.

For example:

```
[WinEventLog://Security]
```



```
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
evt_dc_name = ftw-dc-01.splunk.com
evt_dns_name = splunk.com
checkpointInterval = 5
```

Constraints for using the `evt_dc_name` and `evt_resolve_ad_obj` attributes

When you use the `evt_resolve_ad_obj` and `evt_dc_name` attributes:

- Splunk Enterprise first attempts to resolve SIDs and GUIDs using the domain controller (DC) specified in the `evt_dc_name` attribute first. If it cannot resolve SIDs using this DC, it attempts to bind to the default DC to perform the translation.
- If Splunk cannot contact a DC to translate SIDs, it attempts to use the local machine for translation.
- If none of these methods works, then Splunk prints the SID as it was captured in the event.
- Splunk cannot translate SIDs that are not in the format

```
S-1-N-NN-NNNNNNNNNN-NNNNNNNNNN-NNNNNNNNNN-NNNN.
```

If you discover that Splunk Enterprise does not translate SIDs properly, review `splunkd.log` for clues on what the problem might be.

Specify whether to index starting at earliest or most recent event

Use the `start_from` attribute to specify whether Splunk Enterprise indexes events starting at the earliest event or the most recent. By default, Splunk starts with the oldest data and indexes forward. Do not change this setting, because Splunk Enterprise stops indexing after it has indexed the backlog using this method.

Use the `current_only` attribute to specify whether to index all preexisting events in a given log channel. When set to 1, Splunk indexes only new events that appear from the moment Splunk was started. When set to 0, Splunk indexes all events.

For example:

```
[WinEventLog://Application]
disabled = 0
start_from = oldest
current_only = 1
```

Display events in XML

To have Splunk Enterprise generate events in XML, use the `renderXml` attribute:

```
[WinEventLog://System]
disabled = 0
renderXml = 1
evt_resolve_ad_obj = 1
evt_dns_name = "\"SV5DC02\""
```

This input stanza generates events like the following:

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'>
  <System>
    <Provider Name='Service Control Manager'
      Guid='{555908d1-a6d7-4695-8e1e-26931d2012f4}' EventSourceName='Service
      Control Manager' />
    <EventID Qualifiers='16384'>7036</EventID>
    <Version>0</Version>
    <Level>4</Level>
    <Task>0</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8080000000000000</Keywords>
    <TimeCreated SystemTime='2014-04-24T18:38:37.868683300Z' />
    <EventRecordID>412598</EventRecordID>
    <Correlation />
    <Execution ProcessID='192' ThreadID='210980' />
    <Channel>System</Channel>
    <Computer>SplunkDoc.splunk-docs.local</Computer>
    <Security />
  </System>
  <EventData>
    <Data Name='param1'>Application Experience</Data>
    <Data Name='param2'>>stopped</Data>
    <Binary>410065004C006F006F006B00750070005300760063002F0031000000</Binary>
  </EventData>
</Event>
```

When you instruct Splunk Enterprise to render events in XML, event keys within the XML event render in English regardless of the host system locale. Compare the following events generated on a French version of Windows Server:

Standard event:

```
04/29/2014 02:50:23 PM
LogName=Security
SourceName=Microsoft Windows security auditing.
EventCode=4672
```

EventType=0
 Type=Information
 ComputerName=sacreblue
 TaskCategory=Ouverture de session spéciale
 OpCode=Informations
 RecordNumber=2746
 Keywords=Succès de l'audit
 Message=Privilèges spéciaux attribués à la nouvelle ouverture de session.

Sujet :

ID de sécurité :	AUTORITE NT\Système
Nom du compte :	Système
Domaine du compte :	AUTORITE NT
ID d'ouverture de session :	0x3e7

Privilèges :

SeAssignPrimaryTokenPrivilege
 SeTcbPrivilege
 SeSecurityPrivilege
 SeTakeOwnershipPrivilege
 SeLoadDriverPrivilege
 SeBackupPrivilege
 SeRestorePrivilege
 SeDebugPrivilege
 SeAuditPrivilege
 SeSystemEnvironmentPrivilege
 SeImpersonatePrivilege

XML event:

```

<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'>
  <System><Provider
    Name='Microsoft-Windows-Security-Auditing'
    Guid='{54849625-5478-4994-A5BA-3E3B0328C30D}' />
    <EventID>4672</EventID>
    <Version>0</Version>
    <Level>0</Level>
    <Task>12548</Task>
    <OpCode>0</OpCode>
    <Keywords>0x8020000000000000</Keywords>
    <TimeCreated
      SystemTime='2014-04-29T22:15:03.280843700Z' />
    <EventRecordID>2756</EventRecordID>
    <Correlation/><Execution ProcessID='540'
      ThreadID='372' />
    <Channel>Security</Channel>
    <Computer>sacreblue</Computer>
    <Security/>
  </System>
  <EventData>

```

```

                                <Data Name='SubjectUserSid'>AUTHORITY
NT\System</Data>
                                <Data
Name='SubjectUserName'>System</Data>
                                <Data Name='SubjectDomainName'>AUTHORITY
NT</Data>
                                <Data Name='SubjectLogonId'>0x3e7</Data>
                                <Data
Name='PrivilegeList'>SeAssignPrimaryTokenPrivilege
                                SeTcbPrivilege
                                SeSecurityPrivilege
                                SeTakeOwnershipPrivilege
                                SeLoadDriverPrivilege
                                SeBackupPrivilege
                                SeRestorePrivilege
                                SeDebugPrivilege
                                SeAuditPrivilege
                                SeSystemEnvironmentPrivilege
                                SeImpersonatePrivilege</Data>
                                </EventData>
</Event>

```

The `Data Name` keys in the XML event render in English despite rendering in the system's native language in the standard event.

Use the CLI to configure event log monitoring

You can use the CLI to configure local event log monitoring. Before you use the CLI, create stanza entries in `inputs.conf` first. See ["Use inputs.conf to configure event log monitoring"](#) in this topic.

Note: The CLI is not available for remote Event Log collections.

To list all configured Event Log channels on the local machine:

```
> splunk list eventlog
```

You can also list a specific channel by specifying its name:

```
> splunk list eventlog <ChannelName>
```

To enable an Event Log channel:

```
> splunk enable eventlog <ChannelName>
```

To disable a channel:

```
> splunk disable eventlog <ChannelName>
```

Index exported event log (.evt or .evtx) files

To index exported Windows event log files, use the [instructions for monitoring files and directories](#) to monitor the directory that contains the exported files.

Caution: Do not attempt to monitor an .evt or .evtx file that is open for writing. Windows does not allow read access to these files. Use the event log monitoring feature instead.

Constraints

- As a result of API and log channel processing constraints on Windows XP and Server 2003 systems, imported .evt files from those systems do not contain the "Message" field. This means that the contents of the "Message" field do not appear in your Splunk index.
- Splunk Enterprise on Windows XP and Windows Server 2003/2003 R2 cannot index .evtx files exported from systems running Windows Vista and later or Windows Server 2008/2008 R2 and later.
- Splunk Enterprise on Windows Vista and later and Server 2008/2008 R2 and later can index both .evt and .evtx files.
- If your .evt or .evtx file is not from a standard event log channel, you must make sure that any dynamic link library (DLL) files required by that channel are present on the computer on which you are indexing.
- Splunk Enterprise indexes an .evt or .evtx file in the primary locale/language of the computer that collects the file.
- Files that have been exported from another host do not work with the Splunk Web Upload feature. This is because those files contain information that is specific to the host that generated them. Other hosts won't be able to process the files in their unaltered form.

Note: When producing .evt or .evtx files on one system, and monitoring them on another, it's possible that not all of the fields in each event expand as they would on the system producing the events. This is caused by variations in DLL versions, availability and APIs. Differences in OS version, language, Service Pack level and installed third party DLLs, etc. can also have this effect.

Answers

Have questions? Visit Splunk Answers and see what questions and answers the Splunk community has around Windows event logs.

Monitor file system changes

Splunk Enterprise supports the monitoring of Windows file system changes through the Security Event Log channel. To enable monitoring of changes to files and directories, you first enable security auditing for the files and folders you want to monitor for changes, then use the event log monitor to monitor the Security event log channel.

This procedure of monitoring file system changes replaces the deprecated file system change monitor input.

What do you need to monitor file system changes?

Activity:	Required permissions:
Monitor file system changes	<ul style="list-style-type: none">• Splunk Enterprise must run on Windows AND• Splunk Enterprise must run as the Local System user OR as a domain user with specific security policy rights to read the Security event log AND• You must enable security auditing for the file(s) or director(ies) you want Splunk Enterprise to monitor changes to

Use the Security event log to monitor changes to files

You can monitor changes to files on your system by enabling security auditing on a set of files and/or directories and then monitoring the Security event log channel for change events. The event log monitoring input includes three attributes which you can use in `inputs.conf`.

You can use these attributes outside of the context of the Security event log and file system changes. Also, this list of attributes is only a subset of the available attributes for `inputs.conf`. For additional attributes, read [Monitor Windows event log data](#) in this manual.

Attribute	Description	Default
<code>whitelist</code>	Index events that match the text string specified. This attribute is optional. You can specify one of two formats:	N/A

	<ul style="list-style-type: none"> • One or more Event Log event codes or event IDs (Event Log code/ID format.) • One or more sets of keys and regular expressions (Advanced filtering format.) <p>You cannot mix formats in a single entry. You also cannot mix formats in the same stanza.</p> <p>Splunk Enterprise processes whitelists first, then blacklists. If no whitelist is present, Splunk Enterprise indexes all events.</p> <p>When using the Event Code/ID format:</p> <ul style="list-style-type: none"> • For multiple codes/IDs, separate the list with commas. • For ranges, use hyphens (for example "0-1000,5000-1000"). <p>When using the advanced filtering format:</p> <ul style="list-style-type: none"> • Use '=' between the key and the regular expression that represents your filter (for example "whitelist = EventCode=%^1([8-9])\$%") • You can have multiple key/regular expression sets in a single advanced filtering entry. Splunk Enterprise joins the sets logically. This means that the entry is valid only if all of the sets in the entry are true. • You can specify up to 10 whitelists per stanza by adding a number to the end of the <code>whitelist</code> attribute, for example <code>whitelist1...whitelist9</code>. 	
<code>blacklist</code>	<p>Do not index events that match the text string specified. This attribute is optional.</p> <p>You can specify one of two formats:</p> <ul style="list-style-type: none"> • One or more Event Log event codes or event IDs (Event Log code/ID format.) 	N/A

	<ul style="list-style-type: none"> • One or more sets of keys and regular expressions (Advanced filtering format.) <p>You cannot mix formats in a single entry. You also cannot mix formats in the same stanza.</p> <p>Splunk Enterprise processes whitelists first, then blacklists. If no whitelist is present, Splunk Enterprise indexes all events.</p> <p>When using the Event Code/ID format:</p> <ul style="list-style-type: none"> • For multiple codes/IDs, separate the list with commas. • For ranges, use hyphens (for example "0-1000,5000-1000"). <p>When using the advanced filtering format:</p> <ul style="list-style-type: none"> • Use '=' between the key and the regular expression that represents your filter (for example "whitelist = EventCode=%^1([8-9])\$%") • You can have multiple key/regular expression sets in a single advanced filtering entry. Splunk Enterprise joins the sets logically. This means that the entry is valid only if all of the sets in the entry are true. • You can specify up to 10 blacklists per stanza by adding a number to the end of the <code>blacklist</code> attribute, for example <code>blacklist1...blacklist9</code>. 	
<code>suppress_text</code>	<p>Whether or not to include the message text that comes with a security event.</p> <p>A value of 1 suppresses the message text. A value of 0 preserves the text.</p>	0

Create advanced filters with `whitelist` and `blacklist`

You can perform advanced filtering of incoming events with the `whitelist` and `blacklist` attributes in addition to filtering based solely on event codes. To do this, specify the key/regular expression format in the attribute:

whitelist = key=<regular expression> [key=<regular expression>] ...

In this format, *key* is a valid entry from the following list:

Key	Description
\$TimeGenerated	The time that the computer generated the event. Only generates the time string for the event.
\$Timestamp	The time that the event was received and recorded by the Event Log service. Server Enterprise only generates the time string as the event.
Category	The category number for a specific event source.
CategoryString	A string translation of the category. The translation depends on the event source.
ComputerName	The name of the computer that generated the event.
EventCode	The event ID number for an event. Corresponds to "Event ID" in Event Viewer.
EventType	A numeric value that represents one of the five types of events that can be generated ("Error", "Warning", "Information", "Success Audit", and "Failure Audit".) Available on server machines running Windows Server 2003 and earlier or clients running Windows XP and earlier. See Win32_NTLogEvent class (Windows) (http://msdn.microsoft.com/en-us/library/aa394226(v=vs.85).aspx) on MSDN.
Keywords	An element used to classify different types of events within an event log channel. The Security Event Log channel has this element, for example.
LogName	The name of the Event Log channel that received the event. Corresponds to "Log" in Event Viewer.
Message	The text of the message in the event.
OpCode	The severity level of the event ("OpCode" in Event Viewer.)
RecordNumber	The Windows Event Log record number. Each event on a Windows server gets a unique record number. This number starts at 0 with the first event generated on the system, and increments with each new event generated, until it reached a maximum of 4294967295. It then wraps back over to 0.
Sid	The Security Identifier (SID) of the principal (such as a user, group, computer, or process) entity that was associated with or generated the event. See Win32_UserAccount class (http://msdn.microsoft.com/en-us/library/windows/desktop/aa394507%28v=vs.85%29.aspx) on MSDN.
SidType	A numeric value that represents the type of SID that was associated with the event. See Win32_UserAccount class (http://msdn.microsoft.com/en-us/library/windows/desktop/aa394507%28v=vs.85%29.aspx) on MSDN.
SourceName	The source of the entity that generated the event ("Source" in Event Viewer)

TaskCategory	The task category of the event. Event sources allow you to define categories so you can filter them with Event Viewer (using the "Task Category" field. See Event Categories (Windows) (http://msdn.microsoft.com/en-us/library/aa363649%28VS.85%29.aspx) on MSDN.
Type	A numeric value that represents one of the five types of events that can be generated by Windows ("Error", "Warning", "Information", "Success Audit", and "Failure Audit"). Only available on server machines that run Windows Server 2008 or later, or clients that run Windows Vista or later. See Win32_NTLogEvent class (Windows) (http://msdn.microsoft.com/en-us/library/aa394226(v=vs.85).aspx) on MSDN.
User	The user associated with the event. Correlates to "User" in Event Viewer.

`<regular expression>` is any valid regular expression that represents the filters that you want to include (when used with the `whitelist` attribute) or exclude (when used with the `blacklist` attribute).

To learn more about regular expressions and how to use them, visit the [Regularexpressions.info](http://www.regular-expressions.info) (<http://www.regular-expressions.info>) website.

You can specify more than one regular expression on a single entry line. When you do this, Splunk Enterprise joins the expressions logically. This means that only events that satisfy all of the entries on the line are valid for inclusion or exclusion. For example, this entry:

```
whitelist = EventCode="^1([0-5])$" Message="^Error"
```

means to include events that have an `EventCode` ranging from 10 to 15 and contain a `Message` that begins with the word `Error`.

You can specify up to 10 separate whitelist or blacklist entries in each stanza. To do so, add a number at the end of the `whitelist` or `blacklist` entry on a separate line:

```
whitelist = key=<regular expression>
whitelist1 = key=<regular expression> key2=<regular expression 2>
whitelist2 = key=<regular expression>
```

Note: You cannot specify an entry that has more than one expression that references the same key. If, for example, you specify:

```
whitelist = EventCode="^1([0-5])$" EventCode="^2([0-5])$"
```

Splunk Enterprise ignores the first expression and only attempts to include events that match the second expression. In this case, only events that contain an `EventCode` between 20 and 25 match. Events that contain an `EventCode`

between 10 and 15 do not match. Only the last expression in the entry ever matches.

To resolve this problem, specify two separate entries in the stanza:

```
whitelist = EventCode="^1([0-5])$"
whitelist1 = EventCode="^2([0-5])$"
```

Monitor file system changes

You can monitor file system changes for a set of files or directories.

1. Confirm that you have administrator privileges.
2. Follow the instructions at Auditing Security Events How To (<http://technet.microsoft.com/en-us/library/cc727935%28v=ws.10%29.aspx>) on MS Technet to enable security auditing.
3. Configure the Splunk Enterprise event log monitor input to monitor the Security event log channel.

Note: For instructions on how to configure the Event Log monitor input, read [Monitor Windows event log data](#) in this manual.

Examples of file system change monitoring

Following are `inputs.conf` stanzas that show examples of how to monitor file system changes.

This stanza collects security events with event ID codes 0 to 2000 and 3001-10000.

```
[WinEventLog:Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

This stanza collects security events with event ID codes 0 to 2000 and 3001-10000. It also suppresses the message text that comes in the event ID.

```
[WinEventLog:Security]
disabled = 0
start_from = oldest
current_only = 0
evt_resolve_ad_obj = 1
checkpointInterval = 5
# suppress message text, we only want the event number.
suppress_text = 1
# only index events with these event IDs.
whitelist = 0-2000,2001-10000
# exclude these event IDs from being indexed.
blacklist = 2001-3000
```

Monitor data through Windows Management Instrumentation (WMI)

Splunk Enterprise supports the use of Windows Management Instrumentation (WMI) providers for agentless access to Windows performance and event log data on remote machines. You can pull event logs from all the Windows machines in your environment without installing anything on those machines.

If possible, use a **universal forwarder** rather than WMI to collect data from remote machines. The resource load of WMI can exceed that of a Splunk universal forwarder in many cases. Use a forwarder if you collect multiple event logs or performance counters from each host, or from very busy hosts like domain controllers. See [Considerations for deciding how to monitor remote Windows data](#) in this manual.

WMI-based data inputs can connect to multiple WMI providers. The input runs as a separate process called `splunk-wmi.exe`. It is a **scripted input**.

What do you need to monitor WMI-based data?

Here are the basic minimum requirements to monitor WMI-based data. You might need additional permissions based on the logs or performance counters you want to monitor.

For additional details on what's required to monitor WMI-based data, see "Security and remote access considerations" later in this topic.

Activity	Required permissions

Monitor remote event logs over WMI	<ul style="list-style-type: none"> * Splunk Enterprise must run on Windows * Splunk Enterprise must run as a domain user with at least read access to WMI * Splunk Enterprise must run as a domain user with appropriate access to the desired event logs
Monitor remote performance monitor counters over WMI	<ul style="list-style-type: none"> * Splunk Enterprise must run on Windows * Splunk Enterprise must run as a domain user with at least read access to WMI * Splunk Enterprise must run as a domain user with appropriate access to the Performance Data Helper libraries

Security and remote access considerations

Splunk Enterprise and your Windows network must be correctly configured for WMI data access. Review the following prerequisites before attempting to use Splunk Enterprise to get WMI data.

Before Splunk Enterprise can get WMI-based data:

- It must be installed with a user that has permissions to perform remote network connections.
- The user Splunk Enterprise runs as must be a member of an Active Directory (AD) domain or forest and must have appropriate privileges to query WMI providers.
- The Splunk user must also be a member of the local Administrators group on the computer that runs Splunk Enterprise.
- The computer that runs Splunk Enterprise must be able to connect to the remote machine and must have permissions to get the desired data from the remote machine once it has connected.
- Both the Splunk Enterprise instance and the target machines must be part of the same AD domain or forest.

The user that Splunk Enterprise runs as does not need to be a member of the Domain Admins group (and for security reasons, should not be). However, you must have domain administrator privileges to configure access for the user. If you don't have domain administrator access, find someone who can either configure Splunk user access or give domain administrator rights to you.

If you install Splunk Enterprise as the Local System user, remote authentication over WMI does not work. The Local System user has no access to other machines on the network. It is not possible to grant privileges to a Local System account for access to another host.

You can give the Splunk user access to WMI providers by doing one of the following:

- Adding it to the local Administrators group on each member host you want to poll (not recommended for security reasons).
- Adding it to the Domain Admins global group (not recommended for security reasons).
- Assigning least-permissive rights as detailed below (recommended).

Group memberships and resource access control lists (ACLs)

To maintain security integrity, place Splunk users into a domain global group and assign permissions on Windows machines and resource ACLs to that group, instead of assigning permissions directly to the user. Assignment of permissions directly to users is a security risk, and can cause problems during security audits or future changes.

Configure WMI for least permissive access

If the user you configured Splunk Enterprise to run as is not a domain administrator, you must configure WMI to provide access to this user. Grant only least-permissive access to all Windows resources, including WMI. In order to grant this type of access, follow this checklist. For additional information and step-by-step instructions, see *Prepare your Windows network for a Splunk Enterprise installation* in the *Installation* manual.

You must grant several levels of access to the user Splunk Enterprise runs as for Splunk Enterprise to collect data over WMI using the least-permissive method:

To deploy these user rights assignments domain-wide, use the **Domain Security Policy** (`dcompol.msc`) Microsoft Management Console (MMC) snap-in. After deployment, member hosts inherit those rights assignments on the network during the next AD replication cycle. Restart Splunk Enterprise instances on those machines for the changes to take effect.

To extend this access to domain controllers specifically, assign the rights using the **Domain Controller Security Policy** (`dcpol.msc`) snap-in.

- **Local Security Policy Permissions.** The Splunk user needs the following Local Security Policy user rights assignments defined on each machine you poll for WMI-based data:
 - Access this Computer from the Network
 - Act as part of the operating system

- Log on as a batch job
 - Log on as a service
 - Profile System Performance
 - Replace a process level token
- **Distributed Component Object Model (DCOM) configuration and permissions.** DCOM must be enabled on every machine you want to monitor. In addition, the Splunk Enterprise user must be assigned permissions to access DCOM. There are many methods available to do this, but the best is to nest the "Distributed COM Users" domain global group into the "Distributed COM Users" local group on each machine you want to monitor, then add the Splunk Enterprise user to the "Distributed COM Users" domain global group. See "Securing a Remote WMI Connection" ([http://msdn.microsoft.com/en-us/library/aa393266\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa393266(VS.85).aspx)) on MSDN for advanced options to give the Splunk Enterprise user access to DCOM.
 - **Performance Monitor configuration and permissions.** The Splunk Enterprise user must be a member of the "Performance Log Users" local group in order for Splunk Enterprise to access remote performance objects over WMI. The best way to do this is to nest the "Performance Log Users" domain global group into the "Performance Log Users" local group on each member host and then assign the user to the global group.
 - **WMI namespace security.** The WMI namespace that Splunk Enterprise accesses (most commonly `Root\CIMV2`) must have proper permissions. These permissions must be set manually on each host in your enterprise, as there is no global WMI security. Use the WMI Security MMC snap-in (`wmicmgmt.msc`) to enable the following permissions on the WMI tree for each host at the Root namespace for the Splunk user:
 - Execute Methods
 - Enable Account
 - Remote Enable
 - Read Security

These rights must be assigned to the Root namespace and all subnamespaces below it. See "Managing WMI security" (<https://technet.microsoft.com/en-us/library/cc731011.aspx>) on Microsoft TechNet.

Note: There is no standard facility for deploying WMI security settings remotely to multiple machines at once using Group Policy. However, Set WMI namespace

security via GPO

(<http://blogs.msdn.com/spatdsg/archive/2007/11/21/set-wmi-namespace-security-via-gpo-script.aspx>) on MSDN Blogs offers instructions on how to create a startup script that you can place inside a Group Policy Object (GPO), which sets the namespace security once the GPO applies to the desired hosts. You can then deploy this GPO domain-wide or to one or more Organizational Units (OUs).

- **Firewall configuration.** If you have a firewall enabled, you must configure it to allow access for WMI. If you use the Windows Firewall included with recent versions of Windows, the exceptions list explicitly includes WMI. You must set this exception for both the originating and the target machines. See *Connecting to WMI Remotely Starting with Vista* ([http://msdn.microsoft.com/en-us/library/aa822854\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa822854(VS.85).aspx)) on MSDN for more details.
- **User access control (UAC) configuration.** If you run Windows Vista, Windows 7, Windows 8.1, or the Windows Server 2008 or 2012 families, UAC affects how Windows assigns permissions. See "User Account Control and WMI" ([http://msdn.microsoft.com/en-us/library/aa826699\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa826699(v=vs.85).aspx)) on MSDN.

Test access to WMI providers

After you configure WMI and set up the Splunk user for access to your domain, test access to the remote machine.

This procedure includes steps to temporarily change the Splunk Enterprise data store directory (the location `SPLUNK_DB` points to). You must do this before testing access to WMI. Failure to do so can result in missing WMI events. This is because the `splunk-wmi.exe` process updates the WMI checkpoint file every time it runs.

If you attempt to log into a domain controller, you might have to change your domain controller security policy to assign the "Allow log on locally" policy for the designated user.

1. Log into the machine Splunk Enterprise runs on as the Splunk user.
2. Open a command prompt (click **Start -> Run** and type `cmd`).
3. Go to the `bin` subdirectory under your Splunk Enterprise installation (for example, `cd c:\Program Files\Splunk\bin`).

4. Determine where Splunk Enterprise currently stores its data by running:

```
> splunk show datastore-dir
```

Note: Remember where Splunk Enterprise stores its data. You will recall it later.

5. Run the following command to change where Splunk Enterprise stores its data temporarily:

```
> splunk set datastore-dir %TEMP%
```

Note: This example sets the data store directory to the current directory specified in the TEMP environment variable. If you want to set it to a different directory, you can do so, but the directory must already exist.

6. Restart Splunk Enterprise:

```
> splunk restart
```

Note: It might take a while for Splunk Enterprise to restart.

7. Once Splunk Enterprise has restarted, test access to WMI providers, replacing <host> with the name of the remote host:

```
> splunk cmd splunk-wmi -wql "select * from win32_service" -namespace \\<host>\root\cimv2
```

- If you see data streaming back and no error messages, then Splunk Enterprise was able to connect to the WMI provider and query successfully.
- If there is an error, a message with a reason on what caused the error appears. Look for the `error="<msg>"` string in the output for clues on how to correct the problem.

After testing WMI access, point Splunk Enterprise back to the correct database directory by running the following command, and then restarting Splunk Enterprise:

```
> splunk set datastore-dir <directory shown from Step 4>
```

Configure WMI-based inputs

All remote data collection in Splunk Enterprise on Windows happens through either WMI providers or a forwarder. See [Considerations for deciding how to monitor remote Windows data](#) in this manual.

You can configure WMI-based inputs either in Splunk Web or by editing configuration files. More options are available when using configuration files.

Configure WMI-based inputs with Splunk Web

To add WMI-based inputs, use the "Remote event log monitoring" and "Remote Performance monitoring" data inputs. See [Configure remote Windows performance monitoring with Splunk Web](#). See also [Configure remote Windows event log monitoring](#)

Configure WMI-based inputs with configuration files

wmi.conf handles remote data collection configurations. Review this file to see the default values for WMI-based inputs. If you want to make changes to the default values, edit a copy of `wmi.conf` in `%SPLUNK_HOME%\etc\system\local\`. Set values only for the attributes you want to change for a given type of data input. See [About configuration files in the Admin manual](#).

`wmi.conf` contains several stanzas:

- The `[settings]` stanza, which specifies global WMI parameters.
- One or more input-specific stanzas, which define how to connect to WMI providers to get data from the remote machine.

Global settings

The `[settings]` stanza specifies global WMI parameters. The entire stanza and every parameter within it are optional. If the stanza is not present, Splunk Enterprise assumes system defaults.

When Splunk Enterprise cannot connect to a defined WMI provider, it generates an error in `splunkd.log`:

```
05-12-2011 02:39:40.632 -0700 ERROR ExecProcessor - message from
"C:\Program Files\Splunk\bin\splunk-wmi.exe" WMI - Unable to connect to WMI
namespace "\\w2k3m1\root\cimv2" (attempt to connect took 42.06 seconds)
(error="The RPC server is unavailable." HRESULT=800706BA)
```

The following attributes control how Splunk Enterprise reconnects to a given WMI provider when an error occurs.

Attribute	Description	Default value
<code>initial_backoff</code>	How long, in seconds, to wait the first time after an error occurs before trying	5

	to reconnect to the WMI provider. If connection errors continue to occur, Splunk Enterprise doubles the wait time until it reaches the value specified in <code>max_backoff</code> .	
<code>max_backoff</code>	How long, in seconds, to wait between connection attempts, before invoking <code>max_retries_at_max_backoff</code> .	20
<code>max_retries_at_max_backoff</code>	If the wait time between connection attempts reaches <code>max_backoff</code> , how many times to try to reconnect to the provider, every <code>max_backoff</code> seconds. If Splunk Enterprise continues to encounter errors, it gives up, and won't attempt to connect to the problem provider again until you restart. It will continue to log errors such as the example shown above.	2
<code>checkpoint_sync_interval</code>	How long, in seconds, to wait for state data (event log checkpoint) to be written to disk.	2

Input-specific settings

Input-specific stanzas tell Splunk Enterprise how to connect to WMI providers. They are defined by one of two attributes that specify the type of data Splunk Enterprise should gather. The stanza name can be anything, but usually begins with `WMI:`, for example:

```
[WMI:AppAndSys]
```

When you configure WMI-based inputs in Splunk Web, Splunk Enterprise uses this naming convention for input-specific stanza headers.

You can specify one of two types of data inputs in an input-specific stanza:

- **Event log.** The `event_log_file` attribute tells Splunk Enterprise to expect event log data from the sources defined in the stanza.
- **Windows Query Language (WQL).** The `wql` attribute tells Splunk Enterprise to expect data from a WMI provider. You must also specify a valid WQL statement. You must use this attribute when you collect performance data.

Do not define both of these attributes in one stanza. Use only one or the other. Otherwise, the input defined by the stanza will not run.

The common attributes for both types of inputs are:

Attribute	Description	Default value
<code>server</code>	A comma-separated list of hosts from which to get data. If this attribute is missing, Splunk Enterprise assumes that you want to connect to the local machine.	The local host
<code>interval</code>	Tells Splunk Enterprise how often, in seconds, to poll for new data. If this attribute is not present and defined, the input that the stanza defines will not run.	N / A
<code>disabled</code>	Tells Splunk Enterprise whether this input is enabled or disabled. Set this attribute to 1 to disable the input, and 0 to enable it.	0 (enabled)

The event log-specific parameters are:

Attribute	Description	Default value
<code>event_log_file</code>	A comma-separated list of event log channels to monitor.	N / A
<code>current_only</code>	Whether or not to collect events that occur only when it is running. If events are generated when Splunk Enterprise is stopped, it will not attempt to index those events when it is started again. Set to 1 to collect events that occur only when it is running, and 0 to collect all events.	0 (gather all events)
<code>disable_hostname_normalization</code>	Do not normalize the host name that is retrieved from a WMI event. By default, Splunk Enterprise normalizes host names by producing a single name for the host by identifying various equivalent	0 (normalize host names for WMI events)

	host names for the local system. Set this parameter to 1 to disable host name normalization in events, and 0 to normalize host names in events.	
--	---	--

The WQL-specific parameters are:

Attribute	Description	Default value
wql	A valid WQL statement.	N / A
namespace	(Optional) Specifies the path to the WMI provider. The local machine must be able to connect to the remote machine using delegated authentication. If you do not specify a path to a remote machine, Splunk Enterprise connects to the default local namespace (\Root\CIMV2). This default namespace is where most of the providers you are likely to query reside. Microsoft provides a list of namespaces for Windows XP and later versions of Windows (http://msdn.microsoft.com/en-us/library/aa394084(VS.85).aspx).	\\<local server>\Root\CIMV2
current_only	Whether or not an event notification query is expected. See "WQL query types: event notification versus standard" in this topic for additional information. Set this attribute to 1 to tell Splunk Enterprise to expect an event notification query, and 0 to expect a standard query.	0 (expect a standard query)

WQL query types: event notification versus standard

The `current_only` attribute in WQL stanzas determines the type of query the stanza expects to use to collect WMI-based data. When you set the attribute to 1, the stanza expects event notification data. Event notification data is data that alerts you of an incoming event. To get event notification data, you must use an event notification query.

For example, to find out when a remote host spawns processes, you must use an event notification query. Standard queries have no facilities for notifying you when an event has occurred, and can only return results on information that already exists.

Conversely, if you want to know what already-running processes on your system begin with the word "splunk", you must use a standard query. Event notification queries cannot tell you about static, preexisting information.

Event notification queries require that the WQL statement defined for the stanza be structurally and syntactically correct. Improperly formatted WQL will cause the input defined by the stanza to not run. Review the `wmi.conf` configuration file reference for specific details and examples.

WQL query stanzas do not update the WMI checkpoint file

When you use a WQL query stanza to gather data through WMI, Splunk Enterprise does not update the WMI checkpoint file - the file that determines if WMI data has been indexed. This is by design - a WQL query of any type returns dynamic data and a context for saving a checkpoint for the data produced cannot be built. This means that Splunk Enterprise indexes WMI data that it collects through WQL query stanzas as fresh data each time the stanza runs. This can result in the indexing of duplicate events and possibly impact license volume.

If you need to index data regularly, such as event logs, use the appropriate monitor on a universal forwarder. If you must use WMI, use a standard WMI query type.

Examples of wmi.conf

The following is an example of a `wmi.conf` file:

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

[WMI:AppAndSys]
server = foo, bar
interval = 10
event_log_file = Application, System, Directory Service
disabled = 0

[WMI:LocalSplunkWmiProcess]
interval = 5
wql = select * from Win32_PerfFormattedData_PerfProc_Process where Name
= "splunk-wmi"
disabled = 0

# Listen from three event log channels, capturing log events that occur
only
# while Splunk Enterprise runs. Gather data from three machines.
[WMI:TailApplicationLogs]
interval = 10
```

```

event_log_file = Application, Security, System
server = srv1, srv2, srv3
disabled = 0
current_only = 1

# Listen for process-creation events on a remote machine
[WMI:ProcessCreation]
interval = 1
server = remote-machine
wql = select * from __InstanceCreationEvent within 1 where
TargetInstance isa 'Win32_Process'
disabled = 0
current_only = 1

# Receive events whenever someone plugs/unplugs a USB device to/from the
computer
[WMI:USBChanges]
interval = 1
wql = select * from __InstanceOperationEvent within 1 where
TargetInstance ISA 'Win32_PnPEntity' and
TargetInstance.Description='USB Mass Storage Device'
disabled = 0
current_only = 1

```

Fields for WMI data

When Splunk Enterprise indexes data from WMI-based inputs, it sets the originating host from the data received. It sets the **source** for received events to `wmi`. It sets the **source type** of the incoming events based on the following conditions:

- For event log data, Splunk Enterprise sets the source type to `WinEventLog:<name of log file>`. For example, `WinEventLog:Application`.
- For WQL data, Splunk Enterprise sets the source type to the name of the stanza that defines the input. For example, for a stanza named `[WMI:LocalSplunkdProcess]`, Splunk sets the source type to `WMI:LocalSplunkdProcess`.

WMI and event transformations

WMI events are not available for transformation at index time. You cannot modify or extract WMI events as Splunk Enterprise indexes them. This is because WMI events arrive as a single source (a scripted input), which means they can be matched only as a single source.

You can modify and extract WMI events at search time. You can also address WMI-based inputs at parse time by specifying the sourcetype `[wmi]`.

For information on how to transform events as they arrive in Splunk Enterprise, see [About indexed field extraction](#) in this manual.

Troubleshooting WMI inputs

If you encounter problems receiving events through WMI providers or are not getting the results you expect, see Common Issues with Splunk and WMI in the *Troubleshooting Manual*.

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`.

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.

Activity	Required permissions
Monitor the Registry	* Splunk Enterprise must run on Windows AND * Splunk Enterprise must run as either the local system user OR * Splunk Enterprise must run as a domain user with read access to the Registry hives or keys that you want to monitor

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.

Note: 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:

Event Type	Description
Set	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.
Create	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.
Delete	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.
Rename	Splunk Enterprise generates a Rename event when you rename a Registry key or subkey in RegEdit.
Open	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.
Close	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.
Query	Splunk Enterprise generates a Query event when a program executes the GetValue method on a Registry subkey.

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 `c:\.*` 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**.

Note: The baseline snapshot is an index of your entire Registry, at the time the snapshot is taken. Scanning the Registry to set a baseline index is a CPU-intensive process and might take some time.

11. 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.

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](#).

Note: **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.

Attribute	Description
-----------	-------------

event_status	The result of the registry change attempt. This should always be "(0) The operation completed successfully.". If it is not, there might be problems with the Registry that might eventually require a restore from a backup.
pid	The process ID of the process that attempted to make the Registry change.
process_image	The name of the process that attempted to make the Registry change.
registry_type	The type of Registry operation that the process_image attempted to invoke.
key_path	The Registry key path that the process_image attempted to make a change to.
data_type	The type of Registry data that the process_image making the Registry change tried to get or set.
data	The data that the process_image making the Registry change tried to read or write.

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.
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:

	<ul style="list-style-type: none"> • <code>\\REGISTRY\\USER\\</code> maps to <code>HKEY_USERS</code> or <code>HKU</code> • <code>\\REGISTRY\\USER_Classes</code> maps to <code>HKEY_CLASSES_ROOT</code> or <code>HKCR</code> • <code>\\REGISTRY\\MACHINE</code> maps to <code>HKEY_LOCAL_MACHINE</code> or <code>HKLM</code> • <code>\\REGISTRY\\MACHINE\\SOFTWARE\\Classes</code> maps to <code>HKEY_CLASSES_ROOT</code> or <code>HKCR</code> • <code>\\REGISTRY\\MACHINE\\SYSTEM\\CurrentControlSet\\Hardware Profiles\\Current</code> maps to <code>HKEY_CURRENT_CONFIG</code> or <code>HKCC</code> • There is no direct mapping for <code>HKEY_CURRENT_USER</code> or <code>HKCU</code>, as the Registry monitor runs in kernel mode. Use <code>\\REGISTRY\\USER\\. *</code> (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 <code>\\REGISTRY\\USER\\<SID></code>, where <code>SID</code> is the SID of the desired user.
<code>type</code>	The subset of event types to monitor. Can be one or more of <code>delete</code> , <code>set</code> , <code>create</code> , <code>rename</code> , <code>open</code> , <code>close</code> or <code>query</code> . The values here must be a subset of the values for <code>event_types</code> that you set in <code>sysmon.conf</code> .
<code>baseline</code>	Whether or not to capture a baseline snapshot for that particular hive path. Set to 1 for yes, and 0 for no.
<code>baseline_interval</code>	How long Splunk Enterprise has to have been down before re-taking the snapshot, in seconds. The default value is 86,400 seconds (1 day).
<code>disabled</code>	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 Splunk Enterprise has been down for more than 24 hours since the last checkpoint, 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.

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"

(<http://msdn.microsoft.com/en-us/library/aa373083%28v=VS.85%29.aspx>) 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.

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 needed 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 what's required to monitor performance metrics, read "Security and remote access considerations" later in this topic.

Activity	Required permissions
Monitor local performance metrics	<ul style="list-style-type: none">* Splunk Enterprise must run on Windows.* Splunk Enterprise must run as the Local System user.
Monitor remote performance metrics on another computer over WMI	<ul style="list-style-type: none">* Splunk Enterprise must run on Windows.* Splunk Enterprise must run as a domain or remote user with at least read access to WMI on the target computer.* Splunk Enterprise must run as a domain or remote user with appropriate access to the Performance Data Helper libraries on the target computer.

Security and remote access considerations

Splunk Enterprise gets data from remote machines using either a forwarder or WMI. Splunk recommends using a universal forwarder to send performance data from remote machines to an indexer. See "The universal forwarder" in the *Universal Forwarder* manual.

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 network. You cannot install Splunk Enterprise as the Local System user, and the user you choose determines what Splunk Enterprise sees. See [Security and remote access considerations](#) in the "Monitor WMI Data" topic in this manual.

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

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

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. This is because you can make typos when using configuration files, and it's 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. See [Forward data](#) in this manual.
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.

Note: 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.

Caution: 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.

Note: 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.

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](#).

Note: **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 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, 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 monitor.

In each stanza, you can specify the following attributes.

Attribute	Required?	Description
<code>interval</code>	Yes	How often, in seconds, to poll for new data. If this attribute is not present and defined, the input will not run, as there is no default.
<code>object</code>	Yes	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.
<code>counters</code>	Yes	One or more valid performance counters that are associated with the object specified in <code>object</code> . Separate multiple counters with semicolons. You can also use an asterisk (*) to specify all available counters under a given <code>object</code> . If this attribute is not present and defined, the input will not run, there is no default.
<code>instances</code>	No	One or more valid instances associated with the performance counter specified in <code>counters</code> . 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.
<code>index</code>	No	The index to route performance counter data to. If not present, the default index is used.

<code>disabled</code>	No	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).
<code>showZeroValue</code>	No	<p>Advanced option. Whether or not Splunk Enterprise should collect events that have values of zero.</p> <p>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.)</p>
<code>samplingInterval</code>	No	<p>Advanced option. How often, in milliseconds, that Splunk should collect performance data.</p> <p>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 <code>interval</code> attribute.</p>
<code>stats</code>	No	<p>Advanced option. A semicolon-separated list of statistic values that Splunk Enterprise reports for high-frequency performance sampling.</p> <p>Allowed values are: <code>average</code>, <code>min</code>, <code>max</code>, <code>dev</code>, and <code>count</code>.</p> <p>The default is no setting (disabled).</p>
<code>mode</code>	No	<p>Advanced option. When you enable high-performance sampling, this attribute controls how Splunk Enterprise outputs events.</p> <p>Allowed values are: <code>single</code>, <code>multikv</code>, <code>multiMS</code>, and <code>multikvMS</code></p> <p>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>)</p>

		<p>If you do not enable high-performance sampling, the <code>multikvMS</code> output mode is the same as the <code>multikv</code> output mode.</p> <p>The default is <code>single</code>.</p>
<code>useEnglishOnly</code>	No	<p>Advanced option. 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.</p> <p>If set to true, Splunk Enterprise collects the performance metrics in English regardless of the system locale. It uses the <code>PdhAddEnglishCounter()</code> API to add the counter string. It also disables regular expression and wildcard matching for the <code>object</code> and <code>counter</code> attributes.</p> <p>If set to false, Splunk Enterprise collects the performance metrics in the system language and expects you to configure the <code>object</code> and <code>counter</code> attributes in that language. It uses the <code>PdhAddCounter()</code> API to add the counter string. You can use wildcards and regular expressions, but you must specify valid <code>object</code>, <code>counters</code>, and <code>instances</code> values that are specific to the locale of the operating system.</p> <p>The default is false.</p>
<code>formatString</code>	No	<p>Advanced option. Controls how Splunk Enterprise formats the output of floating-point values for performance counter events.</p> <p>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.</p> <p>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 "printf - C++ reference"(http://www.cplusplus.com/reference/cstdio/printf/) on cplusplus.com.</p>

		<p>When specifying the format, do not use quotes ("). Specify only the valid characters needed to format the string the way you want.</p> <p>The default is <code>%.20g</code>.</p>
--	--	---

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

Here are some example stanzas which 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:

Correct	Incorrect
---------	-----------

[perfmon://CPUTime]	[Perfmon://CPUTime] [PERFMON://CPUTime]
---------------------	--

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 collecting 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 and the machine(s) Splunk collects performance data from must reside in the same AD domain or forest.

Note: 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 manual.

WMI-based performance values versus Performance Monitor values

When gathering remote performance metrics through WMI, some metrics return zero values or values that are not in line with values returned by Performance Monitor. This is because of a limitation in the implementation of WMI for performance monitor counters and 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" (<http://msdn.microsoft.com/en-us/library/aa394084%28v=vs.85%29.aspx>) 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 are running. Performance Monitor objects, meanwhile, are defined 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 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, it's better to 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.

Note: Forwarding performance data requires additional setup.

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

Select the input source

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 the "Query" button to get a list of the performance objects available on the Windows machine you specified in the "Select Target Host" field.

Note: `Win32_PerfFormattedData_*` classes do not show up as available objects in Splunk Web. If you wish to monitor `Win32_PerfFormattedData_*` classes, you must [add them directly](#) in `wmi.conf`.

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.

Note: 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.

Note: 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.

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](#).

Note: **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 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

Remote performance monitoring configurations are controlled by `wmi.conf`. 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. This is because 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

Attribute	Required?	Description	Default
<code>initial_backoff</code>	No	How long, in seconds, to wait before retrying a connection to a WMI provider when an error 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 <code>max_backoff</code> attribute.	5
<code>max_backoff</code>	No	The maximum amount of time, in seconds to attempt to reconnect to a WMI provider.	20
<code>max_retries_at_max_backoff</code>	No	How many times, after <code>max_backoff</code> seconds has been reached between reconnection attempts	2

		with a WMI provider, to continue to attempt to reconnect to that provider.	
<code>checkpoint_sync_interval</code>	No	How long, in seconds, to wait for state data to be flushed to disk.	2

Input-specific settings

Attribute	Required?	Description	Default
<code>interval</code>	Yes	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.	N/A
<code>server</code>	No	A comma-separated list of one or more valid hosts on which you want to monitor performance.	The local machine
<code>event_log_file</code>	No	<p>The names of one or more Windows event log channels to poll. This attribute tells Splunk Enterprise that the incoming data is in event log format.</p> <p>Do not use the <code>event_log_file</code> attribute in a stanza that already contains the <code>wql</code> attribute.</p>	N/A
<code>wql</code>	No	<p>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.</p> <p>Do not use the <code>wql</code> attribute in a stanza that already contains the <code>event_log_file</code> attribute.</p>	N/A
<code>namespace</code>	No		Root\CIMV2

		<p>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.</p> <p>Only use the <code>namespace</code> attribute in a stanza that contains the <code>wql</code> attribute.</p>	
<code>index</code>	No	The desired index to route performance counter data to.	default
<code>current_only</code>	No	<p>The characteristics and interaction of WMI-based event collections.</p> <ul style="list-style-type: none"> • if <code>wql</code> is defined, this attribute tells Splunk Enterprise whether or not it should expect an event 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. • if <code>event_log_file</code> 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. 	N/A

disabled	No	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.	0
----------	----	--	---

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

```
[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
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:

```
[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:

```
[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.

Monitor Windows data with PowerShell scripts

PowerShell is a scripting language that comes with many versions of Windows. It lets you handle Windows operations from a command-line interface. You can create scripts with the language and output the results of those scripts as objects to other scripts.

Splunk Enterprise supports the monitoring of events received through PowerShell scripts. You can use the PowerShell input to run a single PowerShell command or reference a PowerShell script. Splunk Enterprise then indexes the output of these commands or scripts as events.

What do you need to monitor data with PowerShell scripts?

Activity	Required permissions
----------	----------------------

Monitor data with PowerShell scripts	<p>Splunk Enterprise must run on Windows.</p> <p>Splunk Enterprise must run as the Local System user to run all PowerShell scripts.</p> <p>PowerShell v3.0 or later must be installed on the host.</p> <p>Microsoft .NET version 4.5 or later must be installed on the host.</p>
--------------------------------------	--

Configure inputs with configuration files

To configure a PowerShell input in Splunk Enterprise with the `inputs.conf` configuration file:

1. Write a PowerShell command or script to capture the information you want.
2. Copy `inputs.conf` from `%SPLUNK_HOME%\etc\system\default` to `etc\system\local`.
3. Open the file and edit it to enable Windows PowerShell inputs.
4. Restart Splunk Enterprise.

PowerShell input configuration values

Splunk uses the following stanzas in `inputs.conf` to monitor data gathered by PowerShell.

Attribute	Description	Default
<code>script</code>	<p>The PowerShell command or script file to execute.</p> <p>When you specify a script file (.ps1), prepend the script name with a period and a space (". ").</p>	n/a
<code>schedule</code>	<p>How often the command or script should execute.</p> <p>You can specify a number to indicate the interval in seconds, or you can use a valid <code>cron</code> schedule format.</p>	Script runs once
<code>disabled</code>	<p>Whether or not to enable the input.</p> <p>Set to 1 to disable and 0 to enable</p>	0 (enabled)

Following are some examples of how to configure the input:

Single command example:

```
[powershell://Processes-EX1]
script = Get-Process | Select-Object Handles, NPM, PM, WS, VM, Id,
ProcessName, @{n="SplunkHost";e={$Env:SPLUNK_SERVER_NAME}}
schedule = 0 */5 * ? * *
sourcetype = Windows:Process
```

Script example:

```
[powershell://Processes-EX2]
script = . "$SplunkHome\etc\apps\My-App\bin\getprocesses.ps1"
schedule = 0 */5 * ? * *
sourcetype = Windows:Process
```

For more guidance on writing scripts, see [Write scripts for the PowerShell input](#).

Configure inputs with Splunk Web

To configure PowerShell inputs with Splunk Web:

1. Select **Settings > Data inputs** from the system bar.
2. Select "PowerShell v3 modular input."
3. Click **New**.
4. Enter an input name in the "Name" field.
5. Enter a command or path to a script in the "Command or Script Path" field.
6. Enter an interval or cron schedule in the "Cron Schedule" field.

You can also select the source type, host, and default index by clicking the "More settings" checkbox.

7. Click **Next**. The success page loads.

Write scripts for the PowerShell input

Architecture

Splunk Enterprise provides one modular PowerShell input handler. The **PowerShell** handler supports Microsoft PowerShell version 3 and later.

The PowerShell modular input provides a single-instance, multi-threaded script

host that provides a supporting schema, XML configuration through the `stdin` stream, and XML streaming output.

You can define many PowerShell stanzas and run them simultaneously. You can schedule each stanza through the cron syntax. Because all scripts run within the same process, scripts share environment variables such as the current working directory.

Note: The input does not set a host variable in your PowerShell environment. When you write a script for the input, do not refer to `$host` or use the `Write-Host` or `Out-Host` PowerShell cmdlets. Instead, use either the `Write-Output` or `Write-Error` cmdlets.

The input automatically converts all output to key/value pairs based on public properties that are defined in the schema.

Splunk Enterprise also includes a PowerShell module called `LocalStorage`, which exposes three cmdlets:

- `Get-LocalStoragePath`
- `Export-LocalStorage`
- `Import-LocalStorage`

These cmdlets use the Splunk Enterprise checkpoint directory and let you persist key/value pairs of data between scheduled runs of your script. Normally, data does not persist from one invocation to the next.

Specify paths

The input sets the `SplunkHome` variable so you can easily address scripts in add-ons by writing paths like this:

```
[powershell://MSExchange_Health]
script=.
$SplunkHome/etc/apps/TA-Exchange-2010/powershell/health.ps1
```

Besides `$SplunkHome`, there are several other read-only constant variables:

- `SplunkHome` - the directory where you installed Splunk Enterprise (useful for appending `/etc/apps/` paths to)
- `SplunkServerName` - the name configured for this machine to use in events
- `SplunkServerUri` - the Splunk Enterprise REST API address
- `SplunkSessionKey` - the session key (authentication token) needed for accessing the Splunk Enterprise REST API

- `SplunkCheckpointPath` - the path for storing persistent state
- `SplunkServerHost` - the name of the Splunk Enterprise instance that you want to communicate with
- `SplunkStanzaName` - the name of the `inputs.conf` stanza that defined this script

Handle output

The input currently requires that any PowerShell scripts it executes produce output objects that do not have any script properties. Pipe output through the `Select-Object` cmdlet to ensure proper formatting.

The input currently does not process the output of your scripts until your pipeline and runspace are finished. This means the input does not process `ScriptProperty` values, and it also means that you should avoid long-running scripts. You should not write scripts which wait for things to happen unless you exit every time there is output. It also means that all of your output essentially has the same timestamp, unless you override it using the `SplunkTime` variable, as recommended below.

Splunk Enterprise takes each object that your script produces as an output and turns it into an event, wrapped in `<event>` and tags. Splunk Enterprise converts the properties of each object into key/value pairs. However, the value can only be a quoted string, converted by calling the `.ToString()` method. Thus, the output must be simple, and you should flatten any complex nested objects in your script before they are output.

There are a few special property names which have significance for Splunk Enterprise modular inputs and let you override the defaults in the `inputs.conf` stanza. They are:

- `SplunkIndex` - Overrides the index that the output will be stored in
- `SplunkSource` - Overrides the "source" for the output
- `SplunkHost` - Overrides the "host" name for the output
- `SplunkSourceType` - Overrides the "sourcetype" for the output
- `SplunkTime` - Overrides the "time". If you don't specify this, all objects that your script generates in a single execution will get roughly the same timestamp. This is because the script holds the objects for output until it has finished executing, and then marks the objects with the output time. You must specify this value in epoch or POSIX time, which is a positive integer that represents the number of seconds that has elapsed since 0:00 UTC on Thursday, January 1, 1970.

These properties never appear as objects in the key/value output.

If you want to set these properties and override the defaults, use a calculated expression with the `Select-Object` cmdlet or use the `Add-Member` cmdlet to add a `NoteProperty` property.

Monitor Windows host information

Splunk Enterprise supports the monitoring of detailed statistics about the local Windows machine. It can collect the following information about the Windows host:

- **General computer.** The make and model of the computer, its host name and the Active Directory domain it is in.
- **Operating system.** The version and build number of the operating system installed on the computer, as well as any service packs; the computer name; the last time it was started, the amount of installed and free memory, and the system drive.
- **Processor.** The make and model of the CPU(s) installed in the system, their speed and version, the number of processor(s) and core(s), and the processor ID.
- **Disk.** A listing of all drives available to the system and, if available, their file system type and total and available space.
- **Network Adapter.** Information about the installed network adapters in the system, including manufacturer, product name and MAC address.
- **Service.** Information about the installed services on the system, including name, display name, description, path, service type, start mode, state, and status.
- **Process.** Information on the running processes on the system, including the name, the command line (with arguments), when they were started, and the executable's path.

Both full instances of Splunk Enterprise and universal forwarders support local collection of host information.

The host monitor input runs as a process called `splunk-winhostmon.exe`. This process runs once for every input defined, at the interval specified in the input. You can configure host monitoring using Splunk Web or `inputs.conf`.

Why monitor host information?

Windows host monitoring gives you detailed information about your Windows hosts. You can monitor changes to the system, such as installation and removal of software, the starting and stopping of services, and uptime. When a system failure occurs, you can use Windows host monitoring information as a first step into the forensic process. With the Splunk Enterprise search language, you can give your team at-a-glance statistics on all machines in your Windows network.

What's required to monitor host information?

Activity	Required permissions
Monitor host information	<ul style="list-style-type: none">* Splunk Enterprise must run on Windows.* Splunk Enterprise must run as the Local System user or a local administrator account to read all local host information.

Security and remote access considerations

Splunk Enterprise must run as the Local System user to collect Windows host information by default.

Splunk recommends using a universal forwarder to send host information from remote machines to an indexer. Review *The universal forwarder* in the *Universal Forwarder* manual for information about how to install, configure and use the forwarder to collect Windows host data.

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

If you run Splunk Enterprise as a user other than the "Local System" user, then that user must have local Administrator rights on the machine that you want to collect host data. It must also have other permissions, as detailed in *Choose the Windows user Splunk Enterprise should run as* in the *Installation* manual.

Use Splunk Web to configure host monitoring

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 **Files & Directories**.
4. Click **New** to add an input.

By Splunk Home:

1. Click the **Add Data** link in Splunk Home.
2. Click **Monitor** to monitor host information from the local Windows machine.

Select the input source

1. In the left pane, locate and select **Local Windows host monitoring**.
2. In the **Collection Name** field, enter a unique name for this input that you will remember.
3. In the **Event Types** list box, locate the host monitoring event types you want this input to monitor.
4. Click once on each type you want to monitor. Splunk Enterprise moves the type from the "Available type(s)" window to the "Selected type(s)" window.
5. To unselect a type, click on its name in the "Selected type(s)" window. Splunk Enterprise moves the counter from the "Selected type(s)" window to the "Available type(s)" window.
6. (Optional) To select or unselect all of the types, click on the "add all" or "remove all" links. **Note:** Selecting all of the types can result in the indexing of a lot of data, possibly more than your license allows.
7. In the **Interval** field, enter the time, in seconds, between polling attempts for the input.

8. 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](#).

Note: **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 host information.

Use inputs.conf to configure host monitoring

You can edit `inputs.conf` to configure host monitoring. For more information on how to edit configuration files, see *About configuration files* in the *Admin* manual.

1. Create an `inputs.conf` in `%SPLUNK_HOME%\etc\system\local` and open it for editing.

2. Open `%SPLUNK_HOME%\etc\system\default\inputs.conf` and review it for the Windows event log inputs you want to enable.

3. Copy the Windows event log input stanzas you want to enable from `%SPLUNK_HOME%\etc\system\default\inputs.conf`.

4. Paste the stanzas you copied into `%SPLUNK_HOME%\etc\system\local\inputs.conf`.

5. Make edits to the stanzas to collect the Windows event log data you desire.

6. Save `%SPLUNK_HOME%\etc\system\local\inputs.conf` and close it.

7. Restart Splunk Enterprise.

Windows host monitor configuration values

Splunk Enterprise uses the following attributes in `inputs.conf` to monitor Windows host information.

Attribute	Required?	Description
<code>interval</code>	Yes	How often, in seconds, to poll for new data. If you set the interval to a negative number, Splunk Enterprise runs the input one time. If you do not define this attribute, the input does not run, as there is no default.
<code>type</code>	Yes	The type of host information to monitor. Can be one of <code>Computer</code> , <code>operatingSystem</code> , <code>processor</code> , <code>disk</code> , <code>networkAdapter</code> , <code>service</code> , <code>process</code> , or <code>driver</code> . The input does not run if this attribute is not present.
<code>disabled</code>	No	Whether or not to run the input. If you set this attribute to <code>1</code> , then Splunk Enterprise does not run the input.

Examples of Windows host monitoring configurations

Following are some examples of how to use the Windows host monitoring configuration attributes in `inputs.conf`.

```
# Queries computer information.
[WinHostMon://computer]
type = Computer
```

```

interval = 300

# Queries OS information.
# 'interval' set to a negative number tells Splunk Enterprise to
# run the input once only.
[WinHostMon://os]
type = operatingSystem
interval = -1

# Queries processor information.
[WinHostMon://processor]
type = processor
interval = -1

# Queries hard disk information.
[WinHostMon://disk]
type = disk
interval = -1

# Queries network adapter information.
[WinHostMon://network]
type = networkAdapter
interval = -1

# Queries service information.
# This example runs the input ever 5 minutes.
[WinHostMon://service]
type = service
interval = 300

# Queries information on running processes.
# This example runs the input every 5 minutes.
[WinHostMon://process]
type = process
interval = 300

```

Fields for Windows host monitoring data

When Splunk Enterprise indexes data from Windows host monitoring inputs, it sets the **source** for received events to `windows`. It sets the **source type** of the incoming events to `WinHostMon`.

Answers

Have questions? Visit [Splunk Answers](#) and see what questions and answers the Splunk community has around Windows host information.

Monitor Windows printer information

Splunk Enterprise supports the monitoring of statistics about all of the printers and drivers, print jobs, and printer ports on the local Windows host. It can collect the following print system information:

- **Printer.** Information on the print subsystem, such as the status of installed printers, and when printers get added or deleted.
- **Job.** Information on print jobs, including who has printed what, details on the jobs, and the status of existing jobs.
- **Driver.** Information on the print driver subsystem, including information on existing print drivers, and when a print driver gets added or removed.
- **Port.** Information on printer ports installed on the system, and when they get added or removed.

Both full instances of Splunk Enterprise and universal forwarders support local collection of printer subsystem information.

The printer monitor input runs as a process called `splunk-winprintmon.exe`. This process runs once for every input you define, at the interval specified in the input. You can configure printer subsystem monitoring using Splunk Web or `inputs.conf`.

Why monitor printer information?

Windows printer monitoring gives you detailed information about your Windows printer subsystem. You can monitor any changes to the system, such as installation and removal of printers, print drivers, and ports, the starting and completion of print jobs, and learn who printed what when. When a printer failure occurs, you can use print monitoring information as a first step into the forensic process. With the Splunk Enterprise search language, you can give your team at-a-glance statistics on all printers in your Windows network.

What's required to monitor printer information?

Activity	Required permissions
Monitor host information	<ul style="list-style-type: none">* Splunk Enterprise must run on Windows.* Splunk Enterprise must run as the Local System user to read all local host information.

Security and remote access considerations

Splunk Enterprise must run as the Local System user to collect Windows print subsystem information by default.

Use a universal forwarder to send printer information from remote machines to an indexer. If you choose to install forwarders on your remote machines to collect printer subsystem data, then you can install the forwarder as the Local System user on these machines. The Local System user has access to all data on the local machine, but not on remote machines.

If you run Splunk Enterprise as a user other than the "Local System" user, then that user must have local Administrator rights to the machine, and other permissions as detailed in Choose the Windows user Splunk Enterprise should run as in the Installation manual.

Use Splunk Web to configure printer information

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 Windows print 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 print information from the local Windows machine.
3. In the left pane, locate and select **Local Windows print monitoring**.

Select the input source

1. In the **Collection Name** field, enter a unique name for this input that you will remember.
2. In the **Event Types** list box, locate the print monitoring event types you want this input to monitor.
3. Click once on each type you want to monitor. Splunk Enterprise moves the type from the "Available type(s)" window to the "Selected type(s)" window.
4. To unselect a type, click on its name in the "Selected type(s)" window. Splunk Enterprise moves the counter from the "Selected type(s)" window to the "Available type(s)" window.
5. (Optional) To select or unselect all of the types, click on the "add all" or "remove all" links. **Important:** Selecting all of the types can result in the indexing of a lot of data, possibly more than your license allows.
6. In the **Baseline** control, click the **Yes** radio button to run the input as soon as it starts, and no further. Click **No** to run the input at the interval specified in the **Interval (in minutes)** field.
7. 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.

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](#).

Note: **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 print information.

Use inputs.conf to configure host monitoring

You can edit `inputs.conf` to configure host monitoring. For information on how to edit configuration files, see About configuration files in the *Admin* manual.

1. Copy `inputs.conf` from `%SPLUNK_HOME%\etc\system\default` to `etc\system\local`.
2. Use Explorer or the `ATTRIB` command to remove the file's "Read Only" flag.
3. Open the file and edit it to enable Windows print monitoring inputs.
4. Restart Splunk.

Print monitoring configuration values

Splunk Enterprise uses the following attributes in `inputs.conf` to monitor Windows printer subsystem information:

Attribute	Required?	Description
<code>type</code>	Yes	The type of host information to monitor. Can be one of <code>printer</code> , <code>job</code> , <code>driver</code> , or <code>port</code> . The input will not run if this variable is not present.
<code>baseline</code>	No	Whether or not to generate a baseline of the existing state of the printer, job, driver, or port. If you set this attribute to <code>1</code> , then Splunk Enterprise writes a baseline.

		This might take additional time and CPU resources when Splunk Enterprise starts.
disabled	No	Whether or not to run the input. If you set this attribute to 1, then Splunk Enterprise does not run the input.

Examples of Windows host monitoring configurations

Following are some examples of how to use the Windows host monitoring configuration attributes in `inputs.conf`.

```
# Monitor printers on system.
[WinPrintMon://printer]
type = printer
baseline = 0

# Monitor print jobs.
[WinPrintMon://job]
type = job
baseline = 1

# Monitor printer driver installation and removal.
[WinPrintMon://driver]
type = driver
baseline = 1

# Monitor printer ports.
[WinPrintMon://port]
type = port
baseline = 1
```

Fields for Windows print monitoring data

When Splunk Enterprise indexes data from Windows print monitoring inputs, it sets the **source** for received events to `windows`. It sets the **source type** of the incoming events to `WinPrintMon`.

Answers

Have questions? Visit [Splunk Answers](#) and see what questions and answers the Splunk community has around Windows print monitoring.

Monitor Windows network information

Splunk Enterprise supports the monitoring of detailed statistics about network activity into or out of a Windows host. It can collect the following network

information:

- **Network activity.** When a Windows machine performs any kind of network action, Splunk Enterprise can monitor it.
- **Address family.** Whether or not the network transaction was made over the IPv4 or IPv6 protocols.
- **Packet type.** The type of packet sent in the transaction (for example, a 'connect' or 'transport' packet).
- **Protocol.** Whether or not the network transaction was made over the TCP or UDP protocols.
- **Hosts.** Information about the hosts involved in the network transaction, including the local and remote hosts, the ports which the hosts used to communicate, and any available DNS information.
- **Application.** Which application initiated the network transaction.
- **User.** The user that initiated the network transaction, including his or her ID and SID.
- **Miscellany.** Miscellaneous information about the network transaction, including the transport header size and whether or not the transaction was protected by IPsec.

Both full instances of Splunk Enterprise and universal forwarders support local collection of network information.

The network monitor input runs as a process called `splunk-netmon.exe`. This process runs once for every input defined, at the interval specified in the input. You can configure network monitoring using Splunk Web or `inputs.conf`.

Windows network monitoring in Splunk Enterprise is only available on 64-bit Windows systems. It does not function on 32-bit Windows systems.

Why monitor network information?

Windows network monitoring gives you detailed information about your Windows network activity. You can monitor all transactions on the network, such as the initiation of a network connection by a user or process or whether or not the transaction uses the IPv4 or IPv6 address families. The network monitoring facilities in Splunk Enterprise can help you detect and interrupt an incoming (or outgoing) denial of service attack by telling you the involved machines. With Splunk Enterprise search language, you can give your team at-a-glance statistics on all Windows network operations.

What's required to monitor network information?

Activity	Requirements
Monitor network information	<ul style="list-style-type: none">• Splunk must run on Windows.• The Windows version on the machine must be one of:<ul style="list-style-type: none">◆ Windows Vista.◆ Windows 7.◆ Windows 8.◆ Windows 8.1.◆ Windows Server 2008.◆ Windows Server 2008 R2, or◆ Windows Server 2012 R2.• The Windows system must have all available updates and service packs applied, including the Kernel-Mode Driver Framework version 1.11 update on machines that run Windows Vista, Windows 7, Windows Server 2008, and Windows Server 2008 R2.• Splunk must run as the Local System user or a local administrator account to read all local host information.

Security and remote access considerations

Splunk Enterprise must run as the Local System user to collect Windows network information by default.

Use a universal forwarder to send host information from remote machines to an indexer when possible. If you choose to install forwarders on your remote machines to collect Windows network information, then install the forwarder as the Local System user on these machines. The Local System user has access to all data on the local machine, but not on remote machines.

If you run Splunk Enterprise as a user other than the "Local System" user, then that user must have local Administrator rights to the machine and other explicit permissions, as detailed in Choose the Windows user Splunk Enterprise should run as in the *Installation* manual.

Use Splunk Web to configure host monitoring

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 Windows network 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 network information from the local Windows machine, or **Forward** to forward network information from another Windows machine. Splunk Enterprise loads the "Add Data - Select Source" page.

Note: Forwarding network information requires additional setup.

3. In the left pane, locate and select **Local Windows network monitoring**.

Select the input source

1. In the **Network Monitor Name** field, enter a unique name for this input that you will remember.
2. Under **Address family**, check the IP address family types that you want Splunk Enterprise to monitor (either IPv4 or IPv6.)
3. Under **Packet Type**, check the packet types you want the input to monitor (any of **connect**, **accept**, or **transport**.)
4. Under **Direction**, check the network directions that you want the input to monitor (any of **inbound** (toward the monitoring host) or **outbound** (away from the monitoring host)).
5. Under **Protocol**, check the network protocol types that you want the input to monitor (any of **tcp** (Transmission Control Protocol) or **udp** (User Datagram Protocol)).

6. In the **Remote address** text field, enter the host name or IP address of a remote host whose network communications with the monitoring host that you want the input to monitor.

Note: If you want to monitor multiple hosts, enter a regular expression in this field.

7. In the **Process** text field, enter the partial or full name of a process whose network communications you want the input to monitor.

Note: As with the remote address, you can monitor multiple processes by entering a regular expression.

8. In the **User** text field, enter the partial or full name of a user whose network communications you want the input to monitor.

Note: As with the remote address and process entries, you can monitor multiple users by entering a regular expression in this field.

9. 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](#).

Note: **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 the green **Submit** button.

Splunk Enterprise then loads the "Success" page and begins indexing the specified print information.

Use inputs.conf to configure network monitoring

You can edit `inputs.conf` to configure network monitoring. For information on how to edit configuration files, see [About configuration files in the Admin manual](#).

1. Copy `inputs.conf` from `%SPLUNK_HOME%\etc\system\default` to `etc\system\local`.
2. Use Explorer or the `ATTRIB` command to remove the file's "Read Only" flag.
3. Open the file and edit it to enable Windows network monitoring inputs.
4. Restart Splunk.

The next section describes the specific configuration values for host monitoring.

Windows host monitor configuration values

To define a Windows network monitoring input, use the `[WinNetMon://<name>]` stanza in `inputs.conf`. Splunk Enterprise uses the following attributes to configure the Windows network monitor input.

Attribute	Description	Default
<code>disabled = [0 1]</code>	Whether or not the input should run. Set to 1 to disable the input, and 0 to enable it.	0 (enabled)
<code>index = <string></code>	The index that this input should send the data to. This attribute	The default index

	is optional.	
remoteAddress = <regular expression>	<p>Matches against the remote IP address involved in the network transaction. Accepts regular expressions that represent IP addresses only, not host names. Filters out events with remote addresses that do not match the regular expression. Passes through events with remote addresses that match the regular expression.</p> <p>For example: 192\163\..* matches all IP addresses in the 192.163.x.x range.</p>	(empty string - matches everything)
process = <regular expression>	<p>Matches against the process or application name which performed the network access. Filters out events generated by processes that do not match the regular expression. Passes through events generated by processes that match the regular expression.</p>	(empty string - matches all processes or applications)
user = <regular expression>	<p>Matches against the user name which performed network access. Filters out events generated by users that do not match the regular expression. Passes through events generated by users that match the regular expression.</p>	(empty string - includes access by all users)
addressFamily = [ipv4;ipv6]	<p>If set, matches against the address family used in the network access. Accepts semicolon-separated values, for example "ipv4;ipv6".</p>	(empty string - includes all IP traffic.)
packetType = [connect;accept;transport]	<p>Matches against the packet type used in the transaction. Accepts semicolon-separated</p>	(empty string - includes all packet types.)

	values, for example "connect;transport".	
direction = [inbound;outbound]	<ul style="list-style-type: none"> • If set, matches against the general direction of the network traffic. • "Inbound" means traffic coming into the monitoring machine, "outbound" means traffic leaving the monitoring machine. • Accepts semicolon-separated values, for example "inbound;outbound". 	(empty string - includes both directions.)
protocol = [tcp;udp]	<p>Matches against the specified network protocol.</p> <p>"tcp" means Transmission Control Protocol, where networks use handshakes to and state to set up transactions. "udp" means User Datagram Protocol, a stateless, "fire and forget" protocol.</p> <p>Accepts semicolon-separated values, for example "tcp;udp".</p>	(empty string - includes both protocol types.)
readInterval = <integer>	<p>Advanced option. Use the default value unless there is a problem with input performance.</p> <p>How often, in milliseconds, to read the network monitor filter driver. Allows for the adjustment of call frequency into the kernel driver. Higher frequencies might affect network performance, while</p>	100

	lower frequencies can cause event loss. The minimum legal value is 10 and the maximum legal value is 1000.	
driverBufferSize = <integer>	<p>Advanced option. Use the default value unless there is a problem with input performance.</p> <p>The number of network packets it should keep in the network monitor filter driver buffer. Controls the amount of packets that the driver caches. Lower values might result in event loss, while higher values might increase the size of non-paged memory. The minimum legal value is 128 and the maximum legal value is 8192.</p>	1024
mode = <string>	How to output each event. Splunk Enterprise can output each event in either <code>single</code> or <code>multikv</code> (key-value pair) mode.	single
multikvMaxEventCount = <integer>	<p>Advanced option. Use the default value unless there is a problem with input performance.</p> <p>The maximum amount of events to output when you set <code>mode</code> to <code>multikv</code>. The minimum legal value is 10 and the maximum legal value is 500.</p>	100
multikvMaxTimeMs = <integer>	<p>Advanced option. Use the default value unless there is a problem with input performance.</p> <p>The maximum amount of time, in milliseconds, to output</p>	1000

	multikv events when you set mode to multikv. The minimum legal value is 100 and the maximum legal value is 5000.	
--	--	--

Fields for Windows network monitoring data

When Splunk Enterprise indexes data from Windows network monitoring inputs, it sets the **source** for received events to `windows`. It sets the **source type** of the incoming events to `WinNetMon`.

Confirm that your Windows machine is fully patched

If you encounter issues while running the network monitoring input on a Windows Vista, Windows 7, Windows Server 2008, or Windows Server 2008 R2 machine, confirm that you have updated the machine with all available patches, including the Kernel-Mode Driver Framework version 1.11 Update (<http://support.microsoft.com/kb/2685811>) that is part of Knowledge Base article 2685811. Network monitoring input might not function if this update is not present on your system.

Answers

Have questions? Visit Splunk Answers and see what questions and answers the Splunk community has around Windows network monitoring.

Get other kinds of data in

Monitor First In, First Out (FIFO) queues

This topic describes how to configure a First In, First Out (FIFO) input in Splunk Enterprise using `inputs.conf`. Splunk Web does not currently support the definition FIFO inputs.

Note: Data that you send over FIFO queues does not remain in computer memory and can be an unreliable method for data sources. To ensure data integrity, use the [monitor](#) input instead.

Add a FIFO input to `inputs.conf`

To add a FIFO input, add a stanza for it to `inputs.conf` in `$SPLUNK_HOME/etc/system/local/` or your own custom application directory in `$SPLUNK_HOME/etc/apps/`.

If you have not worked with configuration files before, read [About Configuration Files](#) in the *Admin* manual before you begin.

This input stanza directs Splunk Enterprise to read from a FIFO at the specified path.

```
[fifo://<path>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

You can use the following attributes with FIFO stanzas:

Attribute	Description	Default
<code>host = <string></code>	<p>The host key/field to a static value for this stanza. The <code><string></code> is prepended with 'host::'.</p> <p>Sets the host key's initial value. Splunk Enterprise uses this key during parsing/indexing, in</p>	The IP address or fully qualified domain name of the host where the data originated

	particular to set the host field. It also uses the host field at search time.	
<code>index = <string></code>	The index where events from this input will be stored. The <code><string></code> is prepended with 'index::'.	<code>main</code> , or whatever you have set as your default index.
<code>sourcetype = <string></code>	<p>The sourcetype key/field for events from this input. Explicitly declares the source type for this data, as opposed to letting it be determined automatically. This is important both for searchability and for applying the relevant formatting for this type of data during parsing and indexing.</p> <p>Sets the sourcetype key's initial value. Splunk Enterprise uses the key during parsing/indexing, in particular to set the source type field during indexing. It is also the source type field used at search time.</p> <ul style="list-style-type: none"> • The <code><string></code> is prepended with 'sourcetype::'. • For more information about source types, see Why source types matter in this manual. 	Splunk Enterprise picks a source type based on various aspects of the data. There is no hard-coded default.
<code>source = <string></code>	Sets the source key/field for events from this input. The <code><string></code> is prepended with 'source::'.	The input file path.

	Do not override the source field unless absolutely necessary. The input layer provides a more accurate string to aid in problem analysis and investigation, accurately recording the file from which the data was retrieved. Consider use of source types, tagging, and search wildcards before overriding this value.	
<pre>queue = [parsingQueue indexQueue]</pre>	<p>Where the input processor should deposit the events that it reads.</p> <p>Set to "parsingQueue" to apply <code>props.conf</code> and other parsing rules to your data. Set to "indexQueue" to send your data directly into the index.</p>	Defaults to <code>parsingQueue</code> .

Monitor changes to your file system

This feature has been deprecated.

This feature has been deprecated as of Splunk Enterprise version 5.0. This means that although it continues to function in version 6.x of Splunk Enterprise, it might be removed in a future version. As an alternative, you can:

- Learn how to [monitor file system changes on Windows systems](#).
- Use the auditd daemon on *nix systems and monitor output from the daemon.

For a list of all deprecated features, see the topic *Deprecated features* in the *Release Notes*.

The Splunk Enterprise **file system change monitor** tracks changes in your file system. The monitor watches a directory you specify and generates an event when that directory undergoes a change. It is completely configurable and can detect when any file on the system is edited, deleted, or added (not just Splunk-specific files).

For example, you can tell the file system change monitor to watch `/etc/sysconfig/` and alert you any time the system configurations change.

To monitor file system changes on Windows, see [Monitor file system changes](#) in this manual to learn how with Microsoft native auditing tools.

How the file system change monitor works

The file system change monitor detects changes using:

- modification date/time
- group ID
- user ID
- file mode (read/write attributes, etc.)
- optional SHA256 hash of file contents

You can configure the following features of the file system change monitor:

- whitelist using regular expressions
 - ◆ specify files that will be checked, no matter what
- blacklist using regular expressions
 - ◆ specify files to skip
- directory recursion
 - ◆ including symbolic link traversal
 - ◆ scanning multiple directories, each with their own polling frequency
- cryptographic signing
 - ◆ creates a distributed audit trail of file system changes
- indexing entire file as an event on add/change
 - ◆ size cutoffs for sending entire file and/or hashing
- all change events indexed by, and searchable through, Splunk Enterprise

By default, the file system change monitor generates **audit events** whenever the contents of `$SPLUNK_HOME/etc/` are changed, deleted, or added to. When you start Splunk Enterprise for the first time, it generates an audit event for each file in the `$SPLUNK_HOME/etc/` directory and all subdirectories. Afterward, any change in configuration (regardless of origin) generates an audit event for the affected file. If you have configured `signedaudit=true`, Splunk Enterprise indexes the file system change into the **audit index** (`index=_audit`). If `signedaudit` is not turned on, by default, Splunk Enterprise writes the events to the **main** index unless you specify another index.

The file system change monitor does not track the user name of the account executing the change, only that a change has occurred. For user-level

monitoring, consider using native operating system audit tools, which have access to this information.

Caution: Do not configure the file system change monitor to monitor your root file system. This can be dangerous and time-consuming if directory recursion is enabled.

Configure the file system change monitor

Configure the file system change monitor in `inputs.conf`. There is no support for configuring the file system change monitor in Splunk Web. You must restart Splunk Enterprise any time you make changes to the `[fschange]` stanza.

1. Open `inputs.conf`.
2. Add `[fschange:<directory>]` stanzas to specify files or directories that Splunk Enterprise should monitor for changes.
3. Save the `inputs.conf` file and close it.
4. Restart Splunk Enterprise. File system change monitoring begins immediately.

If you want to use this feature with **forwarding**, follow these guidelines:

- To send the events to a remote indexer, use a **heavy forwarder**.
- If you cannot use a heavy forwarder, then follow the configuration instructions at [Use with a universal forwarder](#).

To use the file system change monitor to watch any directory, add or edit an `[fschange]` stanza to `inputs.conf` in `$SPLUNK_HOME/etc/system/local/` or your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files in general, see [About configuration files in the Admin manual](#).

Syntax

Here is the syntax for the `[fschange]` stanza:

```
[fschange:<directory or file to monitor>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

Note the following:

- Splunk Enterprise monitors all adds/updates/deletes to the directory and its subdirectories.
- Any change generates an event that Splunk indexes.
- <directory or file to monitor> defaults to \$SPLUNK_HOME/etc/.

Attributes

All attributes are optional. Here is the list of available attributes:

Attribute	Description	Default
<code>index=<indexname></code>	The index where all generated events should be stored.	<code>main</code> (unless you have turned on audit event signing).
<code>recurse=<true false></code>	Whether or not to recurse all directories within the directory specified in <code><code[fschange]</code></code> . Set to true to recurse all subdirectories and false to specify only the current directory.	<code>true</code>
<code>followLinks=<true false></code>	Whether or not the file system change monitor follows symbolic links. Set to true to follow symbolic links and false not to follow symbolic links.	<code>false</code> Caution: If you are not careful when setting <code>followLinks</code> , file system loops can occur.
<code>pollPeriod=N</code>	Check this directory for changes every N seconds.	3600 seconds If you make a change, the file system audit events could take anywhere between 1 and 3600 seconds to be generated and become available in audit search.
<code>hashMaxSize=N</code>	Calculate a SHA1 hash for every file that is less	-1 (no hashing used for change)

	<p>than or equal to N size in bytes.</p> <p>This hash can be used as an additional method for detecting change in the file/directory.</p>	detection).
<code>signedaudit=<true false></code>	<p>Send cryptographically signed add/update/delete events.</p> <p>Set to true to generate events in the <code>_audit</code> index. Set to false if you're setting the <code>index</code> attribute. Note: When setting <code>signedaudit</code> to true, make sure auditing is enabled in <code>audit.conf</code>.</p>	false
<code>fullEvent=<true false></code>	<p>* Send the full event if an add or update change is detected.</p> <ul style="list-style-type: none"> • Further qualified by the <code>sendEventMaxSize</code> attribute. 	false
<code>sendEventMaxSize=N</code>	<p>* Only send the full event if the size of the event is less than or equal to N bytes.</p> <p>This limits the size of indexed file data.</p>	-1 (unlimited.)
<code>sourcetype = <string></code>	<p>Set the source type for events from this input.</p> <p>"sourcetype::" is prepended to <code><string></code>.</p>	<p><code>audittrail</code> (if <code>signedaudit=true</code>) or <code>fs_notification</code> (if <code>signedaudit=false</code>)</p>
<code>filesPerDelay = <integer></code>	<p>Injects a delay specified by <code>delayInMills</code> after</p>	n/a

	processing <code><integer></code> files. This throttles file system monitoring so it does not consume as much CPU.	
<code>delayInMills = <integer></code>	The delay in milliseconds to use after processing every <code><integer></code> files as specified in <code>filesPerDelay</code> . This is used to throttle file system monitoring so it does not consume as much CPU.	
<code>filters=<filter1>,<filter2>,...<filterN></code>	Each of these filters will apply from left to right for each file or directory that is found during the monitors poll cycle. See the next section for information on defining filters.	n/a

Define a filter

To define a filter to use with the `filters` attribute, add a `[filter...]` stanza as follows:

```
[filter:blacklist:backups]
regex1 = .*bak
regex2 = .*bk
[filter:whitelist:code]
regex1 = .*\.c
regex2 = .*\.h
```

```
[fschange:/etc]
filters = backups,code
```

The following list describes how Splunk Enterprise handles `fschange` whitelist and blacklist logic:

- The events run down through the list of filters until they reach their first match.
- If the first filter to match an event is a whitelist, then Splunk Enterprise indexes the event.

- If the first filter to match an event is a blacklist, the filter prevents the event from getting indexed.
- If an event reaches the end of the chain with no matches, then Splunk Enterprise indexes the event. This means that there is an implicit "all pass" filter built in.

To default to a situation where Splunk Enterprise does not index events if they don't match a whitelist explicitly, end the chain with a blacklist that matches all remaining events.

For example:

```
...
filters = <filter1>, <filter2>, ... terminal-blacklist
```

```
[filter:blacklist:terminal-blacklist]
regex1 = .?
```

If you blacklist a directory including a terminal blacklist at the end of a series of whitelists, then Splunk Enterprise blacklists all its subfolders and files, as they do not pass any whitelist. To accommodate this, whitelist all desired folders and subfolders explicitly ahead of the blacklist items in your filters.

Example of explicit whitelisting and terminal blacklisting

This configuration monitors files in the specified directory with the extensions `.config`, `.xml`, `.properties`, and `.log` and ignores all others.

In this example, a directory could be blacklisted. If this is the case, Splunk Enterprise blacklists **all** of its subfolders and files as well. Only files in the specified directory would be monitored.

```
[filter:whitelist:configs]
regex1 = .*\.config
regex2 = .*\.xml
regex3 = .*\.properties
regex4 = .*\.log

[filter:blacklist:terminal-blacklist]
regex1 = .?

[fschange:/var/apache]
index = sample
recurse = true
followLinks = false
```

```
signedaudit = false
fullEvent = true
sendEventMaxSize = 1048576
delayInMills = 1000
filters = configs,terminal-blacklist
```

Use with a universal forwarder

To forward file system change monitor events from a universal forwarder, you must set `signedaudit = false` and `index=_audit`.

```
[fschange:<directory or file to monitor>]
signedaudit = false
index=_audit
```

With this workaround, Splunk Enterprise indexes file system change monitor events into the `_audit` index with `sourcetype` set to `fs_notification` and `source` set to `fschangemonitor`, instead of the default value of `audittrail` for both `sourcetype` and `source`.

Get data from APIs and other remote data interfaces through scripted inputs

Splunk Enterprise can accept events from scripts that you provide. Scripted input is useful in conjunction with some Windows and *nix command-line tools, such as `vmstat`, `iostat`, `netstat`, `top`, and so on. You can use scripted input to get data from application program interfaces (APIs) and other remote data interfaces and message queues. You can then use commands like `vmstat` and `iostat` on that data to generate metrics and status data. On Windows platforms, you can enable text-based scripts, such those in `perl` and `python`, with an intermediary Windows batch (`.bat`) or PowerShell (`.ps1`) file.

This topic describes how to add scripted inputs that you have already written to your set of inputs. To learn how to develop scripted inputs, see *Build scripted inputs* in the *Developing Views and Apps for Splunk Web* manual.

You can configure scripted inputs from the Settings menu or by editing `inputs.conf`.

Scripts that scripted inputs launch inherit the Splunk Enterprise environment. Be sure to clear environment variables that can affect the operation of a script. The only environment variable that could cause problems is the library path (most commonly known as `LD_LIBRARY_PATH` on Linux, Solaris, and FreeBSD).

Splunk Enterprise logs any messages that scripted inputs send to the `stderr` I/O channel to `splunkd.log`.

Add a scripted input 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 **Scripts**.
4. Click **New** to add an input.

By Splunk Home:

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

Note: Forwarding data from scripted inputs requires additional setup.

3. In the left pane, locate and select **Scripts**.

Select the input source

1. In the **Script Path** drop down, select the path where the script resides. Splunk Enterprise updates the page to include a new drop down list, "Script Name."
2. In the **Script Name** drop-down, select the script that you want to run. Splunk Enterprise updates the page to populate the "Command" field with the script name.

3. In the **Command** field, add any arguments needed to invoke the script.
4. In the **Interval** field, enter the amount of time (in seconds) that Splunk Enterprise should wait before invoking the script.
5. Optionally, In the **Source Name Override** field, enter a new source name to override the default source value, if necessary.
6. 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 source type for the script. You can choose **Select** to pick from the list of available source types on the local machine, or "Manual" to enter the name of a source type.
2. Select the appropriate **Application context** for this input.
3. Set the **Host** name value. You have several choices for this setting. Learn more about setting the host value in ["About hosts"](#).

Note: **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.

4. 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 drop down box.
5. 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 Active Directory node.

Add a scripted input with inputs.conf

You add a scripted input in `inputs.conf` by adding a `[script]` stanza.

Syntax

Here is the syntax for the `[script]` stanza:

```
[script://$SCRIPT]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

Note the following:

- `$SCRIPT` is the full path to the location of the script.
- As a best practice, put your script in the `bin/` directory that is nearest the `inputs.conf` that calls your script. For example, if you are configuring `$SPLUNK_HOME/etc/system/local/inputs.conf`, place your script in `$SPLUNK_HOME/etc/system/bin/`. If you work on an application in `$SPLUNK_HOME/etc/apps/$APPLICATION/`, put your script in `$SPLUNK_HOME/etc/apps/$APPLICATION/bin/`.

Attributes

All attributes are optional. Here is the list of available attributes:

Attribute	Description	Default
<code>interval =</code> <code><number> <cron</code> <code>schedule></code>	<p>How often to execute the specified command. Specify either an integer value representing seconds or a valid cron schedule.</p> <p>When you specify a <code>cron schedule</code>, the script does not execute on start up, but rather at the times that the cron schedule defines.</p>	60 seconds

	<p>Splunk Enterprise keeps one invocation of a script per instance. Intervals are based on when the script completes. If you configure a script to run every 10 minutes and the script takes 20 minutes to complete, the next run will occur 30 minutes after the first run.</p> <p>For constant data streams, enter 1 (or a value smaller than the script's interval). For one-shot data streams, enter -1. Setting <code>interval</code> to -1 causes the script to run each time the splunk daemon restarts.</p>	
<code>index = <string></code>	<p>The index where events from this input should be stored. Splunk Enterprise prepends the <code><string></code> with <code>index::</code>.</p> <p>For more information about the index field, see "How indexing works" in the Managing Indexers and Clusters manual.</p>	<p><code>main</code>, or whatever you have set as your default index.</p>
<code>sourcetype = <string></code>	<p>Sets the sourcetype key/field for events from this input. * The <code><string></code> is prepended with 'sourcetype::'.</p> <p>Explicitly declares the source type for this data, as opposed to letting it be determined automatically. This is important both for searchability and for applying the relevant formatting for this type of data during parsing and indexing.</p> <p>Sets the sourcetype key initial value. Splunk Enterprise uses this key is during parsing/indexing, in particular to set the source type field during indexing. It also uses the source type field at search time.</p>	<p>Splunk Enterprise picks a source type based on various aspects of the data. There is no hard-coded default.</p>

<pre>source = <string></pre>	<p>* Sets the source key/field for events from this input.</p> <ul style="list-style-type: none"> • Note: Do not override the source key unless absolutely necessary. Typically, the input layer will provide a more accurate string to aid in problem analysis and investigation, accurately recording the file from which the data was retrieved. Consider use of source types, tagging, and search wildcards before overriding this value. • Splunk Enterprise prepends <code><string> with source::</code>. 	<p>The input file path</p>
<pre>disabled = <true false></pre>	<p>Whether or not the input should run. Set to true if you want to disable the input.</p>	<p>false</p>

Run scripts continuously

If you want the script to run continuously, write the script to never exit and set it on a short interval. This helps to ensure that if there is a problem the script gets restarted. Splunk Enterprise keeps track of scripts it has spawned and shuts them down on exit.

Use a wrapper script

It is best practice to write a wrapper script for scripted inputs that use commands with arguments. In some cases, the command can contain special characters that Splunk Enterprise escapes when validating text that you have entered in Splunk Web. This causes updates to a previously configured input to fail to save.

Note: Characters that Splunk Enterprise escapes when validating text are those that should not be in paths, such as equals (=) and semi-colon (;).

For example, the following scripted input is not correctly saved when edited in Splunk Web because Splunk Enterprise escapes the equals (=) sign in the parameter to the `myUtil.py` utility:

```
[script://$SPLUNK_HOME/etc/apps/myApp/bin/myUtil.py file=my_data.csv]
disabled = false
```

To avoid this problem, write a wrapper script that contains the scripted input. (Inputs updated by editing the conf file directly are not subject to this input validation.) For information on writing wrapper scripts, see Scripted inputs overview in the *Developing Views and Apps for Splunk Web* manual.

Example using inputs.conf

This example shows the use of the UNIX `top` command as a data input source:

1. Create a new application directory. This example uses `scripts/`:

```
$ mkdir $SPLUNK_HOME/etc/apps/scripts
```

2. All scripts should be run out of a `bin/` directory inside your application directory:

```
$ mkdir $SPLUNK_HOME/etc/apps/scripts/bin
```

3. This example uses a small shell script `top.sh`:

```
$ #!/bin/sh
top -bn 1 # linux only - different OSes have different parameters
```

4. Make sure the script is executable:

```
chmod +x $SPLUNK_HOME/etc/apps/scripts/bin/top.sh
```

5. Test that the script works by running it via the shell:

```
$SPLUNK_HOME/etc/apps/scripts/bin/top.sh
```

The script should send one `top` output.

6. Add the script entry to `inputs.conf` in

```
$SPLUNK_HOME/etc/apps/scripts/local/:
```

```
[script:///opt/splunk/etc/apps/scripts/bin/top.sh]
interval = 5                                # run every 5 seconds
sourcetype = top                            # set sourcetype to top
source = script:///./bin/top.sh             # set source to name of script
```

Note: You might need to modify `props.conf`:

- By default Splunk Enterprise breaks the single `top` entry into multiple events.

- The easiest way to fix this problem is to tell the server to break only before something that does not exist in the output.

For example, adding the following to

`$SPLUNK_HOME/etc/apps/scripts/default/props.conf` forces all lines into a single event:

```
[top]
BREAK_ONLY_BEFORE = <stuff>
```

Since there is no timestamp in the top output we need to tell Splunk Enterprise to use the current time. This is done in `props.conf` by setting:

```
DATETIME_CONFIG = CURRENT
```

Set interval attribute to cron schedule

In the above example, you can also set the `interval` attribute to a "cron" schedule by specifying strings like the following:

`0 * * * *:` Means run once an hour, at the top of the hour.

`*/15 9-17 * * 1-5:` Means run every 15 minutes from 9 am until 5 pm, on Monday to Friday.

`15,35,55 0-6,20-23 1 */2 *:` Means run at 15, 35, and 55 minutes after the hour, between midnight and 7 am and again between 8pm and midnight, on the first of every even month (February, April, June and so on).

For more information about setting cron schedules, read `CRONTAB(5)` on the Crontab website.

Find more data sources to monitor with crawl

This feature has been deprecated.

This feature has been deprecated as of Splunk Enterprise version 6.0. This means that although it continues to function, it might be removed in a future version. As an alternative, you can search for files and directories to monitor manually.

For a list of all deprecated features, see the topic [Deprecated features](#) in the [Release Notes](#).

Use the `crawl` search command to search your file system or network for new data sources to add to your Splunk Enterprise index.

Change default crawler settings by editing `crawl.conf`. You can override the crawler defaults at the time that you run `crawl`.

`crawl` produces a log of crawl activity which it stores in `$SPLUNK_HOME/var/log/splunk/crawl.log`.

Change crawler defaults

Edit `$SPLUNK_HOME/etc/system/local/crawl.conf` to change the default crawler configuration settings. You define the files and network crawlers separately, in their own stanzas.

Syntax

`crawl.conf` contains two stanzas: `[files]` and `[network]`, which define defaults for the files and network crawlers, respectively.

For information on the definable attributes for those stanzas and their default values, read the `crawl.conf` spec file.

Example

Here is an example `crawl.conf` file with settings defined for both the files and network crawlers:

```
[files]
bad_directories_list= bin, sbin, boot, mnt, proc, tmp, temp, home,
mail, .thumbnails, cache, old
bad_extensions_list= mp3, mpg, jpeg, jpg, m4, mcp, mid
bad_file_matches_list= *example*, *makefile, core.*
packed_extensions_list= gz, tgz, tar, zip
collapse_threshold= 10
days_sizek_pairs_list= 3-0,7-1000, 30-10000
big_dir_filecount= 100
index=main
max_badfiles_per_dir=100

[network]
host = myserver
subnet = 24
```

Configure event processing

Overview of event processing

Events are records of activity in log files and machine data. They are primarily what Splunk Enterprise indexes. Events provide information about the systems that produce the machine data. The term **event data** refers to the contents of a Splunk index.

Here is a sample event:

```
172.26.34.223 - - [01/Jul/2005:12:05:27 -0700] "GET /trade/app?action=logout HTTP/1.1" 200 2953
```

When Splunk Enterprise indexes events, it:

- [Configures character set encoding](#).
- [Configures linebreaking for multi-line events](#).
- [Identifies event timestamps](#) (and applies timestamps to events if they do not exist).
- [Extracts a set of useful standard fields](#) such as `host`, `source`, and `sourcetype`.
- [Segments events](#).
- [Dynamically assigns metadata to events](#), if specified.
- [Anonymizes data](#), if specified.

For an overview of the Splunk Enterprise indexing process, see the Indexing overview chapter of the Managing Indexers and Clusters manual.

Configure character set encoding

You can configure **character set encoding** for your data sources. Splunk Enterprise has built-in character set specifications to support internationalization of your **deployment**. Splunk Enterprise supports many languages (including some that do not use Universal Character Set Transformation Format - 8-bit (UTF-8) encoding).

Splunk Enterprise attempts to apply UTF-8 encoding to your sources by default. If a source does not use UTF-8 encoding or is a non-ASCII file, Splunk Enterprise tries to convert data from the source to UTF-8 encoding unless you specify a character set to use by setting the `CHARSET` key in `props.conf`.

You can retrieve a list of the valid character encoding specifications by using the `iconv -l` command on most *nix systems. A port for `iconv` on Windows is available.

Supported character sets

Splunk Enterprise supports an extremely wide range of character sets, including such key ones as:

- UTF-8
- UTF-16LE
- Latin-1
- BIG5
- SHIFT-JIS

See "Comprehensive list of supported character sets" at the end of this topic for the exhaustive list.

Here is a short list of some of the main character sets that Splunk Enterprise supports, along with the languages they correspond to.

Language	Code
Arabic	CP1256
Arabic	ISO-8859-6
Armenian	ARMSCII-8
Belarus	CP1251
Bulgarian	ISO-8859-5
Czech	ISO-8859-2
Georgian	Georgian-Academy
Greek	ISO-8859-7
Hebrew	ISO-8859-8
Japanese	EUC-JP
Japanese	SHIFT-JIS
Korean	EUC-KR
Russian	CP1251
Russian	ISO-8859-5

Russian	KOI8-R
Slovak	CP1250
Slovenian	ISO-8859-2
Thai	TIS-620
Ukrainian	KOI8-U
Vietnamese	VISCII

Manually specify a character set

To manually specify a character set to apply to an input, set the `CHARSET` key in `props.conf`:

```
[spec]
CHARSET=<string>
```

For example, if you have a host that generates data in Greek (called "GreekSource" in this example) and that uses ISO-8859-7 encoding, set `CHARSET=ISO-8859-7` for that host in `props.conf`:

```
[host::GreekSource]
CHARSET=ISO-8859-7
```

Note: Splunk Enterprise only parses character encodings that have UTF-8 mappings. Some EUC-JP characters do not have a mapped UTF-8 encoding.

Automatically specify a character set

Splunk Enterprise can automatically detect languages and proper character sets using its sophisticated character set encoding algorithm.

Configure Splunk Enterprise to automatically detect the proper language and character set encoding for a particular input by setting `CHARSET=AUTO` for the input in `props.conf`. For example, if you want Splunk Enterprise to automatically detect character set encoding for the host "my-foreign-docs", set `CHARSET=AUTO` for that host in `props.conf`:

```
[host::my-foreign-docs]
CHARSET=AUTO
```

Train Splunk Enterprise to recognize a character set

If you want to use a character set encoding that Splunk Enterprise does not recognize, train it to recognize the character set by adding a sample file to the following path:

```
$SPLUNK_HOME/etc/ngram-models/_<language>-<encoding>.txt
```

Once you add the character set specification file, you must restart Splunk Enterprise. After you restart, it recognizes sources that use the new character set, and automatically converts them to UTF-8 format at index time.

For example, if you want to use the "vulcan-ISO-12345" character set, copy the specification file to the following path:

```
/SPLUNK_HOME/etc/ngram-models/_vulcan-ISO-12345.txt
```

Comprehensive list of supported character sets

The common character sets described earlier are a small subset of what the CHARSET attribute can support. Splunk Enterprise also supports a long list of character sets and aliases, identical to the list supported by the *nix `iconv` utility.

Note: Splunk Enterprise ignores punctuation and case when matching CHARSET, so, for example, "utf-8", "UTF-8", and "utf8" are all considered identical.

Here is the full list, with aliases indicated in parentheses:

- utf-8 (aka, CESU-8, ANSI_X3.4-1968, ANSI_X3.4-1986, ASCII, CP367, IBM367, ISO-IR-6, ISO646-US ISO_646.IRV:1991, US, US-ASCII, CSASCII)
- utf-16le (aka, UCS-2LE, UNICODELITTLE)
- utf-16be (aka, ISO-10646-UCS-2, UCS-2, CSUNICODE, UCS-2BE, UNICODE-1-1, UNICODEBIG, CSUNICODE11, UTF-16)
- utf-32le (aka, UCS-4LE)
- utf-32be (aka, ISO-10646-UCS-4, UCS-4, CSUCS4, UCS-4BE, UTF-32)
- utf-7 (aka, UNICODE-1-1-UTF-7, CSUNICODE11UTF7)
- c99 (aka, java)
- utf-ebcdic
- latin-1 (aka, CP819, IBM819, ISO-8859-1, ISO-IR-100, ISO_8859-1:1987, L1, CSISOLATIN1)

- latin-2 (aka, ISO-8859-2, ISO-IR-101, ISO_8859-2:1987, L2, CSISOLATIN2)
- latin-3 (aka, ISO-8859-3, ISO-IR-109, ISO_8859-3:1988, L3, CSISOLATIN3)
- latin-4 (aka, ISO-8859-4, ISO-IR-110, ISO_8859-4:1988, L4, CSISOLATIN4)
- latin-5 (aka, ISO-8859-9, ISO-IR-148, ISO_8859-9:1989, L5, CSISOLATIN5)
- latin-6 (aka, ISO-8859-10, ISO-IR-157, ISO_8859-10:1992, L6, CSISOLATIN6)
- latin-7 (aka, ISO-8859-13, ISO-IR-179, L7)
- latin-8 (aka, ISO-8859-14, ISO-CELTIC, ISO-IR-199, ISO_8859-14:1998, L8)
- latin-9 (aka, ISO-8859-15, ISO-IR-203, ISO_8859-15:1998)
- latin-10 (aka, ISO-8859-16, ISO-IR-226, ISO_8859-16:2001, L10, LATIN10)
- ISO-8859-5 (aka, CYRILLIC, ISO-IR-144, ISO_8859-5:1988, CSISOLATINCYRILLIC)
- ISO-8859-6 (aka, ARABIC, ASMO-708, ECMA-114, ISO-IR-127, ISO_8859-6:1987, CSISOLATINARABIC, MACARABIC)
- ISO-8859-7 (aka, ECMA-118, ELOT_928, GREEK, GREEK8, ISO-IR-126, ISO_8859-7:1987, ISO_8859-7:2003, CSISOLATINGREEK)
- ISO-8859-8 (aka, HEBREW, ISO-8859-8, ISO-IR-138, ISO8859-8, ISO_8859-8:1988, CSISOLATINHEBREW)
- ISO-8859-11
- roman-8 (aka, HP-ROMAN8, R8, CSHPROMAN8)
- KOI8-R (aka, CSKOI8R)
- KOI8-U
- KOI8-T
- GEORGIAN-ACADEMY
- GEORGIAN-PS
- ARMSCII-8
- MACINTOSH (aka, MAC, MACROMAN, CSMACINTOSH) [Note: these MAC* charsets are for MacOS 9; OS/X uses unicode]
- MACGREEK
- MACCYRILLIC
- MACUKRAINE
- MACCENTRALEUROPE
- MACTURKISH
- MACCROATIAN
- MACICELAND
- MACROMANIA
- MACHEBREW

- MACTHAI
- NEXTSTEP
- CP850 (aka, 850, IBM850, CSPC850MULTILINGUAL)
- CP862 (aka, 862, IBM862, CSPC862LATINHEBREW)
- CP866 (aka, 866, IBM866, CSIBM866)
- CP874 (aka, WINDOWS-874)
- CP932
- CP936 (aka, MS936, WINDOWS-936)
- CP949 (aka, UHC)
- CP950
- CP1250 (aka, MS-EE, WINDOWS-1250)
- CP1251 (aka, MS-CYRL, WINDOWS-1251)
- CP1252 (aka, MS-ANSI, WINDOWS-1252)
- CP1253 (aka, MS-GREEK, WINDOWS-1253)
- CP1254 (aka, MS-TURK, WINDOWS-1254)
- CP1255 (aka, MS-HEBR, WINDOWS-1255)
- CP1256 (aka, MS-ARAB, WINDOWS-1256)
- CP1257 (aka, WINBALTRIM, WINDOWS-1257)
- CP1258 (aka, WINDOWS-1258)
- CP1361 (aka, JOHAB)
- BIG-5 (aka, BIG-FIVE, CN-BIG5, CSBIG5)
- BIG5-HKSCS(aka, BIG5-HKSCS:2001)
- CN-GB (aka, EUC-CN, EUCCN, GB2312, CSGB2312)
- EUC-JP (aka,
EXTENDED_UNIX_CODE_PACKED_FORMAT_FOR_JAPANESE,
CSEUCPKDFMTJAPANESE)
- EUC-KR (aka, CSEUCKR)
- EUC-TW (aka, CSEUCTW)
- GB18030
- GBK
- GB_1988-80 (aka, ISO-IR-57, ISO646-CN, CSISO57GB1988, CN)
- HZ (aka, HZ-GB-2312)
- GB_2312-80 (aka, CHINESE, ISO-IR-58, CSISO58GB231280)
- SHIFT-JIS (aka, MS_KANJI, SJIS, CSSHIFTJIS)
- ISO-IR-87 (aka, JIS0208 JIS_C6226-1983, JIS_X0208 JIS_X0208-1983,
JIS_X0208-1990, X0208, CSISO87JISX0208, ISO-IR-159, JIS_X0212,
JIS_X0212-1990, JIS_X0212.1990-0, X0212, CSISO159JISX02121990)
- ISO-IR-14 (aka, ISO646-JP, JIS_C6220-1969-RO, JP,
CSISO14JISC6220RO)
- JISX0201-1976 (aka, JIS_X0201, X0201, CSHALFWIDTHKATAKANA)
- ISO-IR-149 (aka, KOREAN, KSC_5601, KS_C_5601-1987,
KS_C_5601-1989, CSKSC56011987)
- VISCII (aka, VISCII1.1-1, CSVISCII)

- ISO-IR-166 (aka, TIS-620, TIS620-0, TIS620.2529-1, TIS620.2533-0, TIS620.2533-1)

Configure event line breaking

Some events consist of more than one line. Splunk Enterprise handles most multiline events correctly by default. If you have multiline events that Splunk Enterprise doesn't handle properly, configure the software to change its line breaking behavior.

How Splunk Enterprise determines event boundaries

Splunk Enterprise determines event boundaries in two steps:

1. Line breaking, which uses the `LINE_BREAKER` attribute regular expression value to split the incoming stream of bytes into separate lines. By default, the `LINE_BREAKER` is any sequence of newlines and carriage returns (that is, `([\r\n]+)`).
2. Line merging, which only occurs when you set the `SHOULD_LINEMERGE` attribute to "true" (the default). This step uses all the other line merging settings (for example, `BREAK_ONLY_BEFORE`, `BREAK_ONLY_BEFORE_DATE`, `MUST_BREAK_AFTER`, etc.) to merge the previously separated lines into events.

If the second step does not run (because you set the `SHOULD_LINEMERGE` attribute to "false"), then the events are simply the individual lines that the `LINE_BREAKER` attribute determines. The first step is relatively efficient, while the second is relatively slow. Appropriate use of the `LINE_BREAKER` regular expression can produce the results you want in the first step. This is valuable if a significant amount of your data consists of multiline events.

How to configure event boundaries

Many event logs have a strict one-line-per-event format, but some do not. Usually, Splunk Enterprise can recognize the event boundaries. However, if event boundary recognition does not work properly, you can set custom rules in `props.conf`.

To configure multiline events,

1. First, examine the event format. Determine a pattern in the events to set as the start or end of an event.
2. Then, edit `$SPLUNK_HOME/etc/system/local/props.conf`, and set the necessary attributes to configure your data.

There are two ways to handle multiline events:

Break and reassemble the data stream into events

This method usually simplifies the configuration process, as it gives you access to several attributes that you can use to define line-merging rules.

1. Specify a stanza in `props.conf` that represents the stream of data you want to break and reassemble into events.
2. In that stanza, use the `LINE_BREAKER` attribute to break the data stream into multiple lines.
3. Set the `SHOULD_LINEMERGE` attribute to true.
4. Set your line-merging attributes (`BREAK_ONLY_BEFORE`, etc.) to tell Splunk how to reassemble the lines into events.

If your data conforms well to the default `LINE_BREAKER` setting (any number of newlines and carriage returns), you don't need to alter `LINE_BREAKER`. Instead, just set `SHOULD_LINEMERGE=true` and use the line-merging attributes to reassemble it.

Break the data stream directly into real events using the `LINE_BREAKER` feature

This might increase your indexing speed, but is somewhat more difficult to work with. If you find that indexing is slow and a significant amount of your data consists of multiline events, this method can provide significant improvement.

1. Specify a stanza in `props.conf` that represents the stream of data you want to break directly into events.
2. Use the `LINE_BREAKER` attribute with `SHOULD_LINEMERGE=false`.

Line breaking general attributes

These are the `props.conf` attributes that affect line breaking:

Attribute	Description	Default
<code>TRUNCATE =</code> <code><non-negative integer></code>	<p>Change the default maximum line length (in bytes). Although this attribute is a byte measurement, Splunk rounds down line length when this attribute would otherwise land mid-character for multibyte characters.</p> <p>Set to 0 if you never want truncation (very long lines are, however, often a sign of garbage data).</p>	10000 bytes
<code>LINE_BREAKER =</code> <code><regular expression></code>	<p>A regular expression that determines how the raw text stream gets broken into initial events, before any line merging takes place (if specified by the <code>SHOULD_LINEMERGE</code> attribute, described below).</p> <p>The expression must contain a capturing group (a pair of parentheses that defines an identified subcomponent of the match.)</p> <p>Wherever the expression matches, Splunk Enterprise considers the start of the first capturing group to be the end of the previous event, and considers the end of the first capturing group to be the start of the next event.</p>	<code>([\r\n]+)</code> (Splunk Enterprise breaks data into an event for each line, delimited by any number of carriage return (<code>\r</code>) or newline (<code>\n</code>) characters.

	<p>Splunk Enterprise discards the contents of the first capturing group. This content will not be present in any event, as Splunk Enterprise considers this text to come between lines.</p> <p>You can realize a significant boost to processing speed when you use <code>LINE_BREAKER</code> to delimit multiline events (as opposed to using <code>SHOULD_LINEMERGE</code> to reassemble individual lines into multiline events). Consider using this method if a significant portion of your data consists of multiline events.</p> <p>See the <code>props.conf</code> specification file for information on how to use <code>LINE_BREAKER</code> with branched expressions and additional information.</p>	
<code>LINE_BREAKER_LOOKBEHIND = <integer></code>	<p>When there is leftover data from a previous raw chunk, <code>LINE_BREAKER_LOOKBEHIND</code> indicates the number of characters before the end of the raw chunk (with the next chunk concatenated) that Splunk applies the <code>LINE_BREAKER</code> regular expression. You might want to increase this value from its default if you are dealing with especially large or multiline events.</p>	100 characters
<code>SHOULD_LINEMERGE = [true false]</code>	<p>When set to true, Splunk Enterprise combines several input lines into a single event, with configuration based on the attributes described in the next section.</p>	true

Attributes that apply only when *SHOULD_LINEMERGE* is set to true

When you set `SHOULD_LINEMERGE=true` (the default), use these attributes to define line breaking behavior:

Attribute	Description	Default
<code>BREAK_ONLY_BEFORE_DATE = [true false]</code>	When set to true, Splunk Enterprise creates a new event if it encounters a new line with a date.	true Note: If <code>DATETIME_CONFIG</code> is set to <code>CURRENT</code> or <code>NONE</code> , this attribute is not meaningful, because in those cases, Splunk Enterprise does not identify timestamps.
<code>BREAK_ONLY_BEFORE = <regular expression></code>	When set, Splunk Enterprise creates a new event if it encounters a new line that matches the regular expression.	empty string
<code>MUST_BREAK_AFTER = <regular expression></code>	When set and the regular expression matches the current line, Splunk Enterprise always creates a new event for the next input line. Splunk Enterprise might still break before the current line if another rule matches.	empty string
<code>MUST_NOT_BREAK_AFTER = <regular expression></code>	When set and the current line matches the regular expression, Splunk Enterprise does not break on any subsequent lines until the <code>MUST_BREAK_AFTER</code> expression matches.	empty string
<code>MUST_NOT_BREAK_BEFORE = <regular expression></code>	When set and the current line matches the regular expression, Splunk Enterprise does not break the last event	empty string

	before the current line.	
MAX_EVENTS = <integer>	<p>Specifies the maximum number of input lines that will be added to any event.</p> <ul style="list-style-type: none"> • Splunk Enterprise breaks after reading the specified number of lines. 	256 lines

Examples of configuring event line breaking

Specify event breaks

```
[my_custom_sourcetype]
```

```
BREAK_ONLY_BEFORE = ^\d+\s*$
```

This example instructs Splunk Enterprise to divide events by assuming that any line that consists of only digits is the start of a new event. It does this for any data whose source type is set to `my_custom_sourcetype`.

Merge multiple lines into a single event

The following log event contains several lines that are part of the same request. The differentiator between requests is "Path". For this example, assume that all these lines need to be shown as a single event entry.

```
{{"2006-09-21, 02:57:11.58", 122, 11, "Path=/LoginUser
Query=CrmId=ClientABC&ContentItemId=TotalAccess&SessionId=3A1785URH117BEA&Ticket=6
Method=GET, IP=209.51.249.195, Content=", ""}}
```

```
    {"2006-09-21, 02:57:11.60", 122, 15, "UserData:<User
CrmId="clientabc"
UserId="p12345678"><EntitlementList></EntitlementList></User>", ""}}
    {"2006-09-21, 02:57:11.60", 122, 15, "New Cookie:
SessionId=3A1785URH117BEA&Ticket=646A1DA4STF896EE&CrmId=clientabc&UserId=p12345678&Acco
MANUser:
Version=1&Name=&Debit=&Credit=&AccessTime=&BillDay=&Status=&Language=&Country=&Email=&E
""}}
```

To index this multiline event properly, use the `Path` differentiator in your configuration. Add the following to your

```
$SPLUNK_HOME/etc/system/local/props.conf:
```

```
[source::source-to-break]
SHOULD_LINEMERGE = True
BREAK_ONLY_BEFORE = Path=
```

This code tells Splunk Enterprise to merge the lines of the event, and only break before the term `Path=`.

Multiline event line breaking and segmentation limitations

Splunk Enterprise applies line breaking and segmentation limitations to extremely large events:

- **Lines over 10,000 bytes.** Splunk Enterprise breaks lines over 10,000 bytes into multiple lines of 10,000 bytes each when it indexes them. It appends the field `meta::truncated` to the end of each truncated section. It still groups these lines into a single event.
- **Segmentation for events over 100,000 bytes.** In search results, Splunk Enterprise only displays the first 100,000 bytes of an event. Segments after those first 100,000 bytes of a very long line are still searchable, however.
- **Segmentation for events over 1,000 segments.** In search results, Splunk Enterprise displays the first 1,000 segments of an event as segments separated by whitespace and highlighted on mouseover. It displays the rest of the event as raw text without interactive formatting.

Answers

Have questions? Visit [Splunk Answers](#) and see what questions and answers the Splunk community has around line breaking.

Configure event timestamps

This topic discusses what timestamps are in Splunk Enterprise and leads you into the chapter for how to configure them.

Examine this sample event:

```
172.26.34.223 - - [01/Jul/2005:12:05:27 -0700] "GET
/trade/app?action=logout HTTP/1.1" 200 2953
```

The time information in the event:

```
[01/Jul/2005:12:05:27 -0700]
```

is a **timestamp**.

Splunk Enterprise uses timestamps to correlate events by time, create the histogram in Splunk Web, and set time ranges for searches. Most events contain timestamps, and in those cases where an event does not contain timestamp information, Splunk Enterprise attempts to assign a timestamp value to the event at index time.

In most cases, Splunk Enterprise extracts timestamps correctly, but there are situations where you might need to configure timestamp handling. For example, when dealing with some sources or with distributed deployments, you might need to reconfigure timestamp recognition and formatting.

See the ["Configure timestamps"](#) chapter of this manual for specific instructions on how to configure timestamps.

Configure indexed field extraction

This topic discusses what fields Splunk Enterprise extracts at index time and leads to the chapter on how to configure extraction.

Splunk Enterprise can extract the following at index time:

- Default fields
- Custom fields
- File header fields

Splunk Enterprise always extracts a set of default fields for each event. You can configure it to extract custom fields and, for some data, file header fields.

For more information on indexed field extraction, see the chapter [Configure indexed field extraction](#) in this manual.

Anonymize data

This topic discusses how to anonymize data that comes into Splunk Enterprise, such as credit card and Social Security numbers.

You might want to mask sensitive personal data when indexing log events. Credit card numbers and social security numbers are two examples of data that you

might not want to appear in an index. This topic describes how to mask part of confidential fields to protect privacy while providing enough remaining data for use in tracking events.

Splunk Enterprise lets you anonymize data in two ways:

- Through a regular expression (regex) transform.
- Through a `sed` script.

Anonymize data with a regular expression transform

You can configure `transforms.conf` to mask data by means of regular expressions.

This example masks all but the last four characters of fields `SessionId` and `Ticket number` in an application server log.

Here is an example of the desired output:

```
SessionId=#####7BEA&Ticket=#####96EE
```

A sample input:

```
"2006-09-21, 02:57:11.58", 122, 11, "Path=/LoginUser
Query=CrmId=ClientABC&
ContentItemId=TotalAccess&SessionId=3A1785URH117BEA&Ticket=646A1DA4STF896EE&
SessionTime=25368&ReturnUrl=http://www.clientabc.com,
Method=GET, IP=209.51.249.195,
Content=", ""
"2006-09-21, 02:57:11.60", 122, 15, "UserData:<User CrmId="clientabc"
UserId="p12345678"><EntitlementList></EntitlementList></User>", ""
"2006-09-21, 02:57:11.60", 122, 15, "New Cookie:
SessionId=3A1785URH117BEA&
Ticket=646A1DA4STF896EE&CrmId=clientabcUserId=p12345678&AccountId=&AgentHost=man&
AgentId=man, MANUser:
Version=1&Name=&Debit=&Credit=&AccessTime=&BillDay=&Status=
&Language=&Country=&Email=&EmailNotify=&Pin=&PinPayment=&PinAmount=&PinPG=
&PinPGRate=&PinMenu=&", ""
```

To mask the data, modify the `props.conf` and `transforms.conf` files in your `$SPLUNK_HOME/etc/system/local/` directory.

Configure props.conf

1. Edit `$SPLUNK_HOME/etc/system/local/props.conf` and add the following stanza:

```
[<spec>]
```

```
TRANSFORMS-anonymize = session-anonymizer, ticket-anonymizer
```

In this stanza, <spec> must be one of the following:

- <sourcetype>, the source type of an event.
- host::<host>, where <host> is the host of an event.
- source::<source>, where <source> is the source of an event.

In this example, `session-anonymizer` and `ticket-anonymizer` are arbitrary TRANSFORMS class names whose actions you define in stanzas in a corresponding `transforms.conf` file. Use the class names you create in `transforms.conf`.

Configure transforms.conf

2. In `$SPLUNK_HOME/etc/system/local/transforms.conf`, add your TRANSFORMS:

```
[session-anonymizer]
REGEX = (?m)^(.*)SessionId=\w+(\w{4}[&"].*)$
FORMAT = $1SessionId=#####$2
DEST_KEY = _raw
[ticket-anonymizer]
REGEX = (?m)^(.*)Ticket=\w+(\w{4}&.*)$
FORMAT = $1Ticket=#####$2
DEST_KEY = _raw
```

In this transform:

- **REGEX** should specify the regular expression that points to the string in the event you want to anonymize. **FORMAT** specifies the masked values.
- **\$1** is all the text leading up to the regex and **\$2** is all the text of the event after the regular expression.
- **DEST_KEY = _raw** specifies to write the value from **FORMAT** to the raw value in the log - thus modifying the event.

Note: The regular expression processor does not handle multiline events. As a workaround, specify that the event is multiline by placing `(?m)` before the regular expression in `transforms.conf`.

Anonymize data through a `sed` script

You can also anonymize data by using a `sed` script to replace or substitute strings in events.

Most UNIX users are familiar with `sed`, a Unix utility which reads a file and modifies the input as specified by a list of commands. Splunk Enterprise lets you use `sed`-like syntax in `props.conf` to anonymize your data.

Define the `sed` script in `props.conf`

1. Edit or create a copy of `props.conf` in `$SPLUNK_HOME/etc/system/local`.

Create a `props.conf` stanza that uses `SEDCMD` to indicate a `sed` script:

```
[<spec>]
SEDCMD-<class> = <sed script>
```

In this stanza, `<spec>` must be one of the following:

- `<sourcetype>`, the source type of an event.
- `host::<host>`, where `<host>` is the host of an event.
- `source::<source>`, where `<source>` is the source of an event.

The `sed script` applies only to the `_raw` field at index time. Splunk Enterprise currently supports the following subset of `sed` commands:

- ♦ `replace (s)`
 - ♦ character substitution (`y`).

2. After making changes to `props.conf`, restart Splunk Enterprise to enable the configuration.

Replace strings with regular expression match

The syntax for a `sed` replace is:

```
SEDCMD-<class> = s/<regex>/<replacement>/flags
```

In this stanza:

- `regex` is a PERL regular expression.
- `replacement` is a string to replace the regular expression match. It uses `"\n"` for back-references, where `n` is a single digit.
- `flags` can be either `"g"` to replace all matches or a number to replace a specified match.

Example

In the following example, you want to index data containing Social Security and credit card numbers. At index time, you want to mask these values so that only the last four digits are present in your events. Your `props.conf` stanza might look like this:

```
[source::.../accounts.log]
SEDCMD-accounts = s/ssn=\d{5}(\d{4})/ssn=xxxxx\1/g
s/cc=(\d{4}-){3}(\d{4})/cc=xxxx-xxxx-xxxx-\2/g
```

In your accounts events, Social Security numbers appear as `ssn=xxxxx6789` and credit card numbers appear as `cc=xxxx-xxxx-xxxx-1234`.

Substitute characters

The syntax for a `sed` character substitution is:

```
SEDCMD-<class> = y/<string1>/<string2>/
```

This substitutes each occurrence of the characters in `string1` with the characters in `string2`.

Example

You have a file you want to index, `abc.log`, and you want to substitute the capital letters "A", "B", and "C" for every lowercase "a", "b", or "c" in your events. Add the following to your `props.conf`:

```
[source::.../abc.log]
SEDCMD-abc = y/abc/ABC/
```

When you search for `source="*/abc.log"`, you should not find the lowercase letters "a", "b", and "c" in your data. Splunk Enterprise substituted "A" for each "a", "B" for each "b", and "C" for each "c".

Caveats for anonymizing data

Splunk Enterprise does not parse structured data that has been forwarded to an indexer

When you forward structured data to an indexer, Splunk Enterprise does not parse this data once it arrives at the indexer, even if you have configured `props.conf` on that indexer with `INDEXED_EXTRactions`. Forwarded data skips the following queues on the indexer, which precludes any parsing of that data on the

indexer:

- parsing
- aggregation
- typing

The forwarded data must arrive at the indexer already parsed. To achieve this, you must also set up `props.conf` on the forwarder that sends the data. This includes configuration of `INDEXED_EXTRactions` and any other parsing, filtering, anonymizing, and routing rules.

Universal forwarders are capable of performing these tasks solely for structured data. See [Forward data extracted from structured data files](#).

Configure timestamps

How timestamp assignment works

Timestamps are very important to Splunk Enterprise. The software uses timestamps to:

- Correlate **events** by time.
- Create the timeline histogram in Splunk Web.
- Set time ranges for searches.

Splunk Enterprise assigns timestamps to events at **index time**. It usually assigns timestamp values automatically, using information in the raw event data. If an event doesn't contain an explicit timestamp, Splunk Enterprise attempts to assign a timestamp value through other means. For some data, it might need your help to tell it how to recognize the timestamps.

Splunk Enterprise stores timestamp values in the `_time` field (in UTC time format).

Timestamp processing is one of the key steps in **event processing**. For more information on event processing, see the chapter in this manual called "[Configure event processing](#)".

How Splunk Enterprise assigns timestamps

Splunk Enterprise uses the following precedence rules to assign timestamps to events:

1. It looks for a time or date in the event itself using an explicit `TIME_FORMAT`, if provided. You configure the `TIME_FORMAT` attribute in `props.conf`.
2. If no `TIME_FORMAT` was configured for the data, Splunk Enterprise attempts to automatically identify a time or date in the event itself. It uses the source type of the event (which includes `TIME_FORMAT` information) to try to find the timestamp.
3. If an event does not have a time or date, Splunk Enterprise uses the timestamp from the most recent previous event of the same source.
4. If no events in a source have a date, Splunk Enterprise tries to find a date in the source name or file name. Time of day is not identified in filenames. (This

requires that the events have a time, even though they don't have a date.)

5. For file sources, if no date can be identified in the file name, Splunk Enterprise uses the file's modification time.

6. As a last resort, Splunk Enterprise sets the timestamp to the current system time when indexing each event.

Note: Splunk Enterprise can only extract dates from a source, not times. If you need to extract a time from a source, [use a transform](#).

Configure timestamps

Most events don't require any special timestamp handling. Splunk Enterprise automatically recognizes and extracts their timestamps. However, for some sources and distributed deployments, you might need to configure how Splunk Enterprise extracts timestamps, so that they format properly.

There are two ways to configure timestamp extraction:

- Use the "Set Sourcetype" page in Splunk Web to interactively adjust timestamps on sample data. Once you are happy with the results, save the changes to a new source type and then apply that source type to your data inputs. See [The "Set Sourcetypes" page](#).
- Edit props.conf directly. See [Configure timestamp recognition](#).

You can also configure Splunk's timestamp extraction processor to:

- [Apply time zone offsets](#).
- [Pull the correct timestamp from events with more than one timestamp](#).
- [Improve indexing performance](#).

Considerations when adding data from new inputs

If you index some data from a new input and then discover that you need to adjust the timestamp extraction process, you must reindex that data once you've made the configuration changes. Consider previewing your data to prevent the need to reindex.

Alternatively, you can test new data inputs in a test instance of Splunk Enterprise (or in a separate index on the production Splunk instance) before adding data to your production instance. That way, you can delete and reindex until you get the

results you want.

Configure timestamp recognition

Most events do not require special timestamp handling. Splunk Enterprise recognizes and extracts their timestamps. With some sources and distributed deployments, you might need to configure how Splunk Enterprise extracts timestamps, so that they format properly.

There are two ways to configure timestamp extraction:

- Use the "Set Sourcetype" page in Splunk Web to interactively adjust timestamps on sample data. Once you are happy with the results, you can save the changes to a new source type and then apply that source type to your data inputs. See [The "Set Sourcetype" page](#).
- Edit `props.conf` directly, as explained in this topic.

The timestamp processor

The Splunk Enterprise timestamp processor resides at `$SPLUNK_HOME/etc/datetime.xml` by default. You do not need to touch this file normally, unless you work with unusual, custom timestamps. If you need to configure timestamp recognition in some way, you can make the necessary changes by setting `props.conf` timestamp attributes, as described below.

If you have a custom timestamp that cannot be handled by configuring `props.conf`, substitute your own timestamp processor with the `DATETIME_CONFIG` attribute. This attribute specifies the file Splunk Enterprise should use for timestamp processing.

Edit timestamp properties in props.conf

To configure how Splunk Enterprise recognizes timestamps, edit `props.conf`. There are a number of attributes that pertain to timestamps. In particular, you can determine how Splunk Enterprise recognizes a timestamp by using the `TIME_FORMAT` attribute to specify a `strptime()` format for the timestamp.

You can also set other attributes pertaining to timestamps. For example, to specify where a timestamp is located in an event, what time zone to use, or how to deal with timestamps of varying currency.

Edit the `props.conf` file in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files in general, see [About configuration files in the Admin manual](#).

To set timestamp recognition, configure one or more of the timestamp attributes in `props.conf`. Refer to the `props.conf` specification file for detailed information regarding these and other attributes.

Syntax overview

An overview of the syntax for the timestamp attributes follows:

```
[<spec>]
DATETIME_CONFIG = <filename relative to $SPLUNK_HOME>
TIME_PREFIX = <regular expression>
MAX_TIMESTAMP_LOOKAHEAD = <integer>
TIME_FORMAT = <strptime-style format>
TZ = <posix time zone string>
MAX_DAYS_AGO = <integer>
MAX_DAYS_HENCE = <integer>
MAX_DIFF_SECS_AGO = <integer>
MAX_DIFF_SECS_HENCE = <integer>
```

In this syntax, `<spec>` can be:

- `<sourcetype>`, the source type of an event.
- `host::<host>`, where `<host>` is the host value for an event.
- `source::<source>`, where `<source>` is the source value for an event.

If an event contains data that matches the value of `<spec>`, then the timestamp rules specified in the stanza apply to that event. You can have multiple stanzas, to handle different `<spec>` values.

Timestamp attributes

These are the timestamp attributes settable through `props.conf`:

Attribute	Description	Default
DATETIME_CONFIG = <filename relative to \$SPLUNK_HOME>	<p>Specify a file to use to configure the Splunk Enterprise timestamp processor.</p> <p>Under normal circumstances, you do not need to create your own</p>	\$SPLUNK_HOME/etc/datetime

	<p>timestamp processor file or modify the default <code>datetime.xml</code> file. The other <code>props.conf</code> attributes, described in this topic, can usually tweak the Splunk Enterprise timestamp recognition capability to meet your needs. However, if your data has a custom timestamp format, you might need to substitute your own version of this file.</p> <p>Set <code>DATETIME_CONFIG = NONE</code> to prevent the timestamp processor from running. When timestamp processing is off, Splunk Enterprise does not look at the text of the event for the timestamp--it instead uses the event's "time of receipt"; in other words, the time the event is received via its input. For file-based inputs, this means that Splunk Enterprise derives the event timestamp from the modification time of the input file.</p> <p>Set <code>DATETIME_CONFIG = CURRENT</code> to assign the current system time to each event as it's indexed.</p> <p>Note: Both <code>CURRENT</code> and <code>NONE</code> explicitly disable timestamp identification, so the default event boundary detection (<code>BREAK_ONLY_BEFORE_DATE = true</code>) is likely not to work as desired. When using these settings, use <code>SHOULD_LINEMERGE</code> and/or the <code>BREAK_ONLY_*</code> , <code>MUST_BREAK_*</code> settings to control event merging.</p>	
<code>TIME_PREFIX = <regular expression></code>	<p>When set, Splunk Enterprise looks for a match for this regex in the</p>	<p>empty string</p>

	<p>event text before attempting to extract a timestamp. The timestamp algorithm only looks for a timestamp in the event text that follows the end of the first regex match.</p> <p>You should use a regular expression that points exactly before your event's timestamp. For example, if the timestamp follows the phrase <code>abc123</code> in your events, you should set <code>TIME_PREFIX</code> to <code>abc123</code>.</p> <p>If the <code>TIME_PREFIX</code> cannot be found in the event text, timestamp extraction does not take place.</p>	
<code>MAX_TIMESTAMP_LOOKAHEAD = <integer></code>	<p>Specify how far (how many characters) into an event Splunk Enterprise should look for a timestamp.</p> <p>This constraint is applied starting from the location positioned by <code>TIME_PREFIX</code>.</p> <p>For example, if <code>TIME_PREFIX</code> positions a location 11 characters into the event, and <code>MAX_TIMESTAMP_LOOKAHEAD</code> is set to 10, timestamp extraction will be constrained to characters 11 through 20.</p> <p>If set to 0 or -1, the length constraint for timestamp recognition is effectively disabled. This can have negative performance implications which scale with the length of input lines (or with event size when</p>	150 characters

	LINE_BREAKER is redefined for event splitting).	
TIME_FORMAT = <strptime-style format>	<p>Specifies a <code>strptime()</code> format string to extract the timestamp.</p> <p><code>strptime()</code> is a Unix standard for designating time formats. For more information, see the section Enhanced strptime() support, below.</p> <p>TIME_FORMAT starts reading after the TIME_PREFIX (or directly at the start of the event, if there is no TIME_PREFIX attribute). If you use a TIME_PREFIX, it <i>must</i> match up to and including the character before the timestamp begins. If you don't set TIME_PREFIX but you do set TIME_FORMAT, the timestamp must appear at the very start of each event; otherwise, Splunk Enterprise will not be able to process the formatting instructions, and every event will contain a warning about the inability to use strptime. (It's possible that you will still end up with a valid timestamp, based on how Splunk attempts to recover from the problem.)</p> <p>For best results, the <code><strptime-style format></code> should describe the day of the year and the time of day.</p> <p>If <code><strptime-style format></code> contains an hour component, but no minute component, TIME_FORMAT ignores the hour component. It treats the format as an anomaly and considers the</p>	empty string

	precision to be date-only.	
TZ = <timezone_identifier>	<p>Splunk Enterprise determines a particular event's time zone as follows:</p> <ul style="list-style-type: none"> • If the event has a time zone in its raw text (such as UTC or -08:00), use that. • Otherwise, if TZ is set to a valid time zone string, use that. Specify a time zone setting using a value from the zoneinfo TZ database. • If an event that arrives at an indexer has originated from a forwarder, and both indexer and forwarder run Splunk Enterprise version 6.0 or later, then use the time zone that the forwarder provides. • Otherwise, use the time zone of the system that runs splunkd. <p>For more details and examples, see Specify time zones of timestamps in this manual.</p>	empty string
TZ_ALIAS = <key=value>[, <key=value>]...	<p>Provides admin-level control over how timezone strings extracted from events are interpreted. For example, EST can mean Eastern (US) Standard Time or Eastern (Australian) Standard Time. There are many other three letter timezone acronyms with multiple expansions.</p> <p>There is no requirement to use TZ_ALIAS if the traditional Splunk default mappings for these values work as expected. For example,</p>	not set

	<p>EST maps to the Eastern US by default.</p> <p>Has no effect on the <code>TZ</code> value. It affects only timezone strings from event text, either from any configured <code>TIME_FORMAT</code> or from pattern-based guess fallback.</p> <p>The setting is a list of <code>key=value</code> pairs, separated by commas.</p> <p>The key is matched against the text of the timezone specifier of the event, and the value is the timezone specifier to use when mapping the timestamp to UTC/GMT.</p> <p>The value is another <code>TZ</code> specifier that expresses the desired offset.</p> <p>Example: <code>TZ_ALIAS = EST=GMT+10:00</code> (See the props.conf example file in the Configuration File Reference for more examples).</p>	
<code>MAX_DAYS_AGO = <integer></code>	<p>Specifies the maximum number of days in the past, from the current date, that an extracted date can be valid.</p> <p>For example, if <code>MAX_DAYS_AGO = 10</code>, Splunk Enterprise ignores dates older than 10 days from the current date and instead either uses the timestamp of the previous event, or uses the current index time of the event if it cannot determine a timestamp in the previous event.</p> <p>The maximum settable number of days in the past is 10951.</p>	<p>2000 days</p> <p>Note: If you have data that more than 2000 days old, increase this setting.</p>

<p>MAX_DAYS_HENCE = <integer></p>	<p>Specifies the maximum number of days in the future from the current date that an extracted date can be valid.</p> <p>For example, if MAX_DAYS_HENCE = 3, the software ignores dates that are more than 3 days in the future and instead either uses the timestamp of the previous event, or uses the current index time of the event if it cannot determine a timestamp from the previous event.</p> <p>Note: False positives are less likely with a tighter window. Change this attribute with caution.</p> <p>If your servers have the wrong date set or are in a time zone that is one day ahead, set this value to at least 3.</p> <p>This allows timestamp extractions that are up to a day in the future.</p> <p>The maximum settable number of days is 10950.</p>	<p>2 days</p>
<p>MAX_DIFF_SECS_AGO = <integer></p>	<p>If the event timestamp is more than <integer> seconds before the previous timestamp, Splunk only accepts it if it has the same time format as the majority of timestamps from the source.</p> <p>If your timestamps are wildly out of order, consider increasing this value.</p> <p>The maximum settable number of seconds is 2147483646.</p>	<p>3600 seconds (1 hour)</p>
		<p>604800 seconds (1 week)</p>

<pre>MAX_DIFF_SECS_HENCE = <integer></pre>	<p>If the event's timestamp is more than <code><integer></code> seconds after the previous timestamp, Splunk only accepts it if it has the same time format as the majority of timestamps from the source.</p> <p>If your timestamps are wildly out of order, or if you have logs that are written less than once a week, consider increasing this value.</p> <p>The maximum settable number of seconds is 2147483646.</p>
--	---

Enhanced `strptime()` support

Use the `TIME_FORMAT` attribute in `props.conf` to configure timestamp parsing. This attribute takes a `strptime()` format string, which it uses to extract the timestamp.

Splunk Enterprise implements an enhanced version of Unix `strptime()` that supports additional formats, allowing for microsecond, millisecond, any time width format, and some additional time formats for compatibility. The additional formats include:

<code>%N</code>	For GNU date-time nanoseconds. Specify any sub-second parsing by providing the width: <code>%3N</code> = milliseconds, <code>%6N</code> = microseconds, <code>%9N</code> = nanoseconds.
<code>%Q,%q</code>	For milliseconds, microseconds for Apache Tomcat. <code>%Q</code> and <code>%q</code> can format any time resolution if the width is specified.
<code>%l</code>	For hours on a 12-hour clock format. If <code>%l</code> appears after <code>%S</code> or <code>%s</code> (like <code>"%H:%M:%S.%l"</code>), it takes on the <code>log4cpp</code> meaning of milliseconds.
<code>%+</code>	For standard Unix date format timestamps.
<code>%v</code>	For BSD and OSX standard date format.
<code>%Z, %z, %::z, %:::z</code>	The time zone abbreviation (<code>%Z</code>) or ISO-8601-style numeric timezone (<code>%z</code> -for example, <code>-0800</code> for PST or <code>+0000</code> for GMT, or nothing if the time zone cannot be determined.) GNU libc support.

%o	For AIX timestamp support (%o used as an alias for %Y).
%p	The locale's equivalent of AM or PM. (Note: there may be none.)
%s	Epoch (10 digits)

Note: A `strptime` expression that ends with a literal dot and subsecond specifier such as %Q, %q, %N will treat the terminal dot and conversion specifier as optional. If the .subseconds portion is absent from the text, it will still extract.

strptime() format expression examples

Here are some sample date formats, with the `strptime()` expressions that handle them:

1998-12-31	%Y-%m-%d
98-12-31	%y-%m-%d
1998 years, 312 days	%Y years, %j days
Jan 24, 2003	%b %d, %Y
January 24, 2003	%B %d, %Y
1397477611.862	%s.%3N

Note: Splunk Enterprise does not currently recognize non-English month names in timestamps. If you have an app that writes non-English month names to log files, reconfigure the app to use numerical months, if possible.

Examples

Your data might contain an easily recognizable timestamp, such as:

```
...FOR: 04/24/07 PAGE 01...
```

To extract that timestamp, add this stanza in `props.conf`:

```
[host::foo]
TIME_PREFIX = FOR:
TIME_FORMAT = %m/%d/%y
```

Another example that includes time zone information:

```
?Valid_Until=Thu Dec 31 17:59:59 GMT-06:00 2020
```

To extract the timestamp, add this to `props.conf`:

```
[host::bar]
TIME_PREFIX = Valid_Until=
TIME_FORMAT = %b %d %H:%M:%S %Z%z %Y
```

Your data might contain other information that Splunk Enterprise parses as timestamps, for example:

```
...1989/12/31 16:00:00 Wed May 23 15:40:21 2007...
```

Splunk Enterprise extracts the date as Dec 31, 1989, which is not useful. In this case, configure `props.conf` to extract the correct timestamp from events from `host::foo`:

```
[host::foo]
TIME_PREFIX = \d{4}/\d{2}/\d{2} \d{2}:\d{2}:\d{2} \w+\s
TIME_FORMAT = %b %d %H:%M:%S %Y
```

This configuration assumes that all timestamps from `host::foo` are in the same format. Configure your `props.conf` stanza to be as granular as possible to avoid potential timestamping errors.

For more information on extracting the correct timestamp from events containing multiple timestamps, see [Configure timestamp assignment for events with multiple timestamps](#).

Configure timestamps for specific needs

You can use the attributes described in this topic to configure the Splunk Enterprise timestamp extraction processor for some specialized purposes, such as:

- [To apply time zone offsets.](#)
- [To pull the correct timestamp from events with more than one timestamp.](#)
- [To improve indexing performance.](#)

Configure how timestamps appear in search results

You can use your browser locale setting to configure how the browser formats Splunk Enterprise timestamps in search results. For information on setting the browser locale, see [User language and locale](#).

Reconfigure how timestamps appear in raw data

Even though Splunk Enterprise uses the browser locale to configure how timestamps appear in search results, the raw data still remains in its original format. You might want to change this so that the data format is standardized in both raw data and search results. Do this with `props.conf` and `transforms.conf`. Here is an example:

Assume the timestamp data in the raw event looks like this:

```
06/07/2011 10:26:11 PM
```

but you want it to look like this (to correspond with how it appears in search results):

```
07/06/2011 10:26:11 PM
```

This example shows briefly how you can use `props.conf` and `transforms.conf` to transform the timestamp in the raw event.

In `transforms.conf`, add this stanza:

```
[resortdate]
REGEX = ^(\d{2})\./(\d{2})\./(\d{4})\s([^\s/]+)
FORMAT = $2/$1/$3 $4
DEST_KEY = _raw
```

In `props.conf`, add this stanza, where `<spec>` qualifies your data:

```
[<spec>]
TRANSFORMS-sortdate = resortdate
```

Answers

Have questions? Visit [Splunk Answers](#) and see what questions and answers the Splunk community has around timestamp recognition and configuration.

Configure timestamp assignment for events with multiple timestamps

If an event contains more than one timestamp, you can specify which timestamp Splunk Enterprise uses. This is especially useful when indexing events that contain syslog host-chaining data.

Configure positional timestamp extraction by editing `props.conf`. For general information on editing `props.conf` for timestamps, see [Configure timestamp recognition](#).

Configure positional timestamp extraction

Configure Splunk Enterprise to recognize a timestamp anywhere in an event by adding `TIME_PREFIX` and `MAX_TIMESTAMP_LOOKAHEAD` attributes to a `props.conf` stanza. By setting a regular expression (regex) value for `TIME_PREFIX`, you tell Splunk Enterprise what pattern of characters indicate the point to start looking for the timestamp. Set a value for `MAX_TIMESTAMP_LOOKAHEAD` to tell Splunk Enterprise how far into an event (past the `TIME_PREFIX` location) to look for the timestamp. By constraining lookahead, you can improve both accuracy and performance.

When `TIME_PREFIX` is set, Splunk Enterprise scans the event text for a match to its regex before it tries to extract a timestamp. The Splunk Enterprise timestamping algorithm only looks for a timestamp in the text following the end of the first regex match. So if `TIME_PREFIX` is set to `abc123`, only the text following the first occurrence of `abc123` is used for timestamp extraction.

`TIME_PREFIX` also sets the start point for `MAX_TIMESTAMP_LOOKAHEAD`; the lookahead starts after the matched portion of text in the `TIME_PREFIX` regex. For example, if `TIME_PREFIX` matches text through the first 11 characters of the event and the timestamp you want to extract is always within the next 30 characters, you can set `MAX_TIMESTAMP_LOOKAHEAD=30`. Timestamp extraction would be limited to text starting with character 12 and ending with character 41.

Example

Say you have an event that looks like this:

```
1989/12/31 16:00:00 Wed May 23 15:40:21 2007 ERROR UserManager -  
Exception thrown  
Ignoring unsupported search for eventtype: /doc  
sourcetype="access_combined"  
NOT eventtypetag=bot
```

To identify the timestamp as the second string of time information, `May 23 15:40:21 2007`, configure `props.conf` like this:

```
[source::/Applications/splunk/var/spool/splunk]  
TIME_PREFIX = \d{4}/\d{2}/\d{2} \d{2}:\d{2}:\d{2} \w+\s
```


`MAX_TIMESTAMP_LOOKAHEAD = 21`

This configuration instructs Splunk Enterprise to locate events that match the first timestamp construction, but *ignore* that timestamp in favor of a timestamp that occurs within the following 21 characters (a number it gets from the `MAX_TIMESTAMP_LOOKAHEAD` attribute). Splunk Enterprise will find the second timestamp because it always occurs within that 21-character limit.

Note: Optimize the speed of timestamp extraction by setting the value of `MAX_TIMESTAMP_LOOKAHEAD` to look only as far into an event as you need for the timestamp you want to extract. In this example, `MAX_TIMESTAMP_LOOKAHEAD` is optimized to look just 21 characters into the event past the regular expression value.

Specify time zones for timestamps

If you index data from different time zones, you can use time zone offsets to ensure that they correlate correctly when you search. You can configure time zones based on the host, source, or source type of an event.

Configure time zones in `props.conf`. For general information on editing `props.conf` for timestamps, see [Configure timestamp recognition](#).

How Splunk applies time zones

By default, Splunk Enterprise applies time zones with these rules, in this order:

- Splunk Enterprise uses the time zone specified in raw event data (for example, PST, -0800).
- Splunk Enterprise uses the value of a `TZ` attribute set in `props.conf`, if the event matches the host, source, or source type that the stanza specifies.
- If an event that arrives at an indexer originated from a forwarder, and both the forwarder and the receiving indexer run Splunk Enterprise version 6.0 or later, then Splunk Enterprise uses the time zone that the forwarder provides.
- Otherwise, Splunk Enterprise uses the time zone of the host that indexes the event.

Note: If you change the time zone setting in the system Splunk Enterprise runs on, you must restart Splunk Enterprise for the software to pick up the change.

Specify time zones in props.conf

To configure time zone settings, edit props.conf in

`$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files in general, see [About configuration files in the Admin manual](#).

Configure time zones by adding a `TZ` attribute to the appropriate stanza in `props.conf`. The `TZ` attribute recognizes zoneinfo TZ IDs. (See all the time zone TZ IDs in the zoneinfo (TZ) database.) Inside the stanza for a host, source, or source type, set the `TZ` attribute to the TZ ID for the desired time zone. This should be the time zone of the events coming from that host, source, or sourcetype.

You do not configure the time zone for the indexer in Splunk Enterprise, but in the underlying operating system. As long as the time is set correctly on the host system of the indexer, the offsets to event time zones will be calculated correctly.

Examples of time zone specification in props.conf

Following are some examples of how to specify time zones in props.conf.

In the first example, events come into the indexer from New York City (in the US/Eastern time zone) and Mountain View, California (US/Pacific). To correctly handle the timestamps for these two sets of events, the `props.conf` for the indexer needs the time zone to be specified as US/Eastern and US/Pacific, respectively.

The first example sets the time zone to US/Eastern for any events coming from hosts whose names match the regular expression `nyc.*`:

```
[host::nyc*]  
TZ = US/Eastern
```

The second example sets the time zone to US/Pacific for any events coming from sources in the path `/mnt/ca/...`:

```
[source::/mnt/ca/...]  
TZ = US/Pacific
```

zoneinfo (TZ) database

The zoneinfo database is a publicly maintained database of time zone values.

- UNIX versions of Splunk rely on a TZ database included with the UNIX distribution you're running on. Most UNIX distributions store the database in the directory: `/usr/share/zoneinfo`.
- Solaris versions of Splunk store TZ information in this directory:
`/usr/share/lib/zoneinfo`.
- Windows versions of Splunk ship with a copy of the TZ database.

Refer to the zoneinfo (TZ) database for all permissible TZ values.

Map timezone strings extracted from event data

Use the `TZ_ALIAS` attribute in `props.conf` to change Splunk's interpretation of a timezone acronym string occurring in event data. For example, "EST" means Eastern (US) Standard Time by default, but your event data might be using that value instead to designate Eastern (Australian) Standard Time. To change the meaning of "EST" to the latter, set the attribute like this:

```
TZ_ALIAS = EST=GMT+10:00
```

Then, when Splunk encounters "EST" in event data, it will interpret it as "GMT+10:00", rather than the default of "GMT- 5:00".

As this example shows, you can map a timezone string to an existing string plus offset value. You can also just map one TZ string directly to another.

When mapping timezone strings, be sure to handle both summer and winter versions of the time zones. If mapping EST, also map EDT, for example - depending on whatever your local pairs are. Test your software to see what timezone strings it produces.

You can specify multiple mappings. The syntax for `TZ_ALIAS` is:

```
TZ_ALIAS = <key=value>[,<key=value>]...
```

For more information, including examples, see the `props.conf` specification and example file in the Configuration File Reference.

Set the time zone for a user's search results

When you add or edit users using Splunk's built-in authentication, you can set a user time zone. Search results for that user will appear in the specified time zone. This setting, however, does not change the actual event data, whose time zone is determined at index time. For information on setting this value, see *Configure users with Splunk Web* in the *Securing Splunk* manual.

Tune timestamp recognition for better indexing performance

To speed up indexing, you can use `props.conf` to adjust how far ahead into events the Splunk Enterprise timestamp processor looks, or even turn off the timestamp processor altogether.

For general information on editing `props.conf` for timestamps, see [Configure timestamp recognition](#).

Adjust timestamp lookahead

Timestamp lookahead determines how far (how many characters) into an event the timestamp processor looks for a timestamp. Adjust how far the timestamp processor looks by setting the `MAX_TIMESTAMP_LOOKAHEAD` attribute.

The default number of characters that the timestamp processor looks into an event is 150. You can set `MAX_TIMESTAMP_LOOKAHEAD` to a lower value to speed up indexing. You should particularly do this if the timestamps always occur in the first part of the event.

Example:

This example tells Splunk Enterprise to look for timestamps in just the first 20 characters of events coming from source `foo`.

```
[source::foo]
MAX_TIMESTAMP_LOOKAHEAD = 20
...
```

Disable timestamp processor

You can turn off the timestamp processor entirely to improve indexing performance. Turn off timestamp processing for events matching a specified host, source, or sourcetype by setting the `DATETIME_CONFIG` attribute to `NONE`. When `DATETIME_CONFIG=NONE`, Splunk Enterprise does not look at the text of the event for the timestamp. Instead, it uses the event "time of receipt"; in other words, the time the event is received via its input. For file-based inputs (such as monitor) this means that Splunk Enterprise derives the timestamp from the modification time of the input file.

You can also increase indexing performance by setting `DATETIME_CONFIG` to `CURRENT`. This causes Splunk Enterprise to assign the current system time to each event at the time of indexing.

Example:

This example turns off timestamp extraction for events that come from the source `foo`.

```
[source::foo]
DATETIME_CONFIG = NONE
...
```

Note: Both `CURRENT` and `NONE` disable timestamp identification, so the default event boundary detection (`BREAK_ONLY_BEFORE_DATE = true`) might not work as you expect. When you use these settings, specify `SHOULD_LINEMERGE` or the `BREAK_ONLY_*` and `MUST_BREAK_*` settings to control event merging.

Configure indexed field extraction

About indexed field extraction

When Splunk Enterprise indexes data, it parses the data stream into a series of **events**. As part of this process, it adds a number of **fields** to the **event data**. These fields include **default fields** that it adds automatically and any **custom fields** that you specify.

The process of adding fields to events is known as **field extraction**. There are two types of field extraction:

- **Indexed field extraction**, which was described briefly at the start of this topic and which forms the basis for this chapter. Splunk Enterprise stores these fields in the index, and the fields become part of the event data.
- **Search-time field extraction**, which takes place when you search through data. Splunk Enterprise creates those fields on the fly and does not store them in the index. See *About fields in the Knowledge Manager Manual* for information about this type of field extraction.

There are two types of indexed fields:

- **Default fields**, which Splunk automatically adds to each event. See [About default fields](#) in this chapter.
- **Custom fields**, which you specify. See [Create custom fields at index time](#) in this manual.

Note: When working with fields, consider that most machine data either does not have structure or has structure that changes constantly. For this type of data, Splunk Enterprise recommends using search-time field extraction for the purposes of flexibility. Search-time field extraction is trivial to modify within Splunk Enterprise once you set it.

Other types of data might exhibit a more fixed structure, or the structure might already be defined within the data or events in the file. Splunk Enterprise provides the option to read the structure of these kinds of files (such as comma-separated value files (CSV), tab-separated value files (TSV), pipe-separated value files, and JavaScript Object Notation (JSON) data sources) and perform field mapping at index time. To learn how this works, see [Extract data from files with headers](#) in this manual.

About default fields (host, source, sourcetype, and more)

When Splunk Enterprise indexes data, it tags each event with a number of fields. These fields become part of the index **event data**. The fields that Splunk Enterprise adds automatically are known as **default fields**.

Default fields serve a number of purposes:

- The default field `index` identifies the index in which the event is located.
- The default field `linecount` describes the number of lines the event contains.
- The default field `timestamp` specifies the time at which the event occurred.

Splunk Enterprise uses the values in some of the fields, particularly `sourcetype`, when indexing the data, in order to create events properly. Once the data has been indexed, you can use the default fields in your searches.

The complete list of default fields follows:

Type of field	List of fields	Description
Internal fields	<code>_raw</code> , <code>_time</code> , <code>_indextime</code> , <code>_cd</code>	These fields contain information that Splunk uses for its internal processes.
Basic default fields	<code>host</code> , <code>index</code> , <code>linecount</code> , <code>punct</code> , <code>source</code> , <code>sourcetype</code> , <code>splunk_server</code> , <code>timestamp</code>	These fields provide basic information about an event, such as where it originated, what kind of data it contains, what index it's located in, how many lines it contains, and when it occurred.
Default datetime fields	<code>date_hour</code> , <code>date_mday</code> , <code>date_minute</code> , <code>date_month</code> , <code>date_second</code> , <code>date_wday</code> , <code>date_year</code> , <code>date_zone</code>	These fields provide additional searchable granularity to event timestamps. Note: Only events that have timestamp information in them as generated by their respective systems will have <code>date_*</code> fields. If an event has a <code>date_*</code> field, it represents the value of time/date directly from the event itself. If you have specified any timezone conversions or changed the value of the

		time/date at indexing or input time (for example, by setting the timestamp to be the time at index or input time), these fields will not represent that.
--	--	--

For information about default fields from the search perspective, see [Use default fields](#) in the Knowledge Manager Manual.

You can also specify additional, custom fields for Splunk to include in the index. See [Create custom fields at index-time](#) in this chapter.

This topic focuses on three key default fields:

- **host**
- **source**
- **sourcetype**

Defining host, source, and sourcetype

The host, source, and sourcetype fields are defined as follows:

- **host** - An event host value is typically the hostname, IP address, or fully qualified domain name of the network host from which the event originated. The host value lets you locate data originating from a specific device. For more information on hosts, see [About hosts](#).
- **source** - The source of an event is the name of the file, stream, or other input from which the event originates. For data monitored from files and directories, the value of source is the full path, such as `/archive/server1/var/log/messages.0` or `/var/log/`. The value of source for network-based data sources is the protocol and port, such as `UDP:514`.
- **sourcetype** - The source type of an event is the format of the data input from which it originates, such as `access_combined` or `cisco_syslog`. The source type determines how Splunk formats your data. For more information on source types, see [Why source types matter](#).

Source vs sourcetype

Source and source type are both default fields, but they are entirely different otherwise, and can be easily confused.

- The **source** is the name of the file, stream, or other input from which a particular event originates.

- The **sourcetype** specifies the format for the event. Splunk uses this field to determine how to format the incoming data stream into individual events.

Events with the same source type can come from different sources, for example, if you monitor `source=/var/log/messages` and receive direct syslog input from `udp:514`. If you search `sourcetype=linux_syslog`, Splunk will return events from both of those sources.

Under what conditions should you override host and sourcetype assignment?

Much of the time, Splunk Enterprise can automatically identify host and sourcetype values that are both correct *and* useful. But situations do come up that require you to intervene in this process and provide override values.

Override host assignment

You might want to change your default `host` assignment when:

- You load archive data in bulk that was originally generated from a different host and you want those events to have that host value.
- You forward data from a different host. (The forwarder assigns its host name unless you specify otherwise.)
- You are working with a centralized log server environment, which means that all of the data received from that server will have the same host, even if it originated elsewhere.

For detailed information about hosts, see the chapter [Configure host values](#).

Override sourcetype assignment

You might want to change your default `sourcetype` assignment when:

- Splunk Enterprise cannot automatically format the data properly, resulting in problems such as wrong timestamping or event linebreaking.
- You want to apply source types to specific events coming through a particular input, such as events that originate from a discrete group of hosts, or even events that are associated with a particular IP address or userid.

There are also steps you can take to expand the range of source types that Splunk Enterprise automatically recognizes, or to simply rename source types.

Assign default fields dynamically

This feature lets you dynamically assign default fields, also known as "metadata", to files as they are being consumed by Splunk. Use this feature to specify source type, host, or source dynamically for incoming data. This feature is useful mainly with scripted data -- either a **scripted input** or an existing file processed by a script.

Do not use dynamic metadata assignment with ongoing file monitoring (tail) inputs. For more information about file inputs, see [Monitor files and directories](#) in this manual.

Note: The modular inputs feature has superseded this `***SPLUNK***` header feature. If you need to present dynamically-generated values of host, source and sourcetype to Splunk Enterprise, consider writing a modular input.

To use this feature, you append a single dynamic input header to your file and specify the metadata fields you want to assign values to. The available metadata fields are sourcetype, host, and source.

You can use this method to assign metadata instead of editing `inputs.conf`, `props.conf`, and `transforms.conf`.

Configure a single input file

To use this feature for an existing input file, edit the file (either manually or with a script) to add a single input header:

```
***SPLUNK*** <metadata field>=<string> <metadata field>=<string>
```

```
...
```

1. Set `<metadata field>=<string>` to a valid metadata/value pair. You can specify multiple pairs. For example, `sourcetype=log4j host=swan`.
2. Add the single header anywhere in your file. Any data following the header will be appended with the attributes and values you assign until the end of the file is reached.
3. Add your file to `$SPLUNK_HOME/var/spool/splunk` or any other directory being monitored by Splunk.

Configure with a script

In the more common scenario, you write a script to dynamically add an input header to your incoming data stream. Your script can also set the header dynamically based on the contents of the input file.

Create custom fields at index time

There are times when you might find reason to add custom indexed fields. For example, you might have a situation where certain search-time field extractions are noticeably impacting search performance. This can happen, for example, if you commonly search a large event set with expressions like `foo!=bar` or `NOT foo=bar`, and the field `foo` nearly *always* takes on the value `bar`.

Conversely, you might want to add an indexed field if the value of a search-time extracted field exists outside of the field more often than not. For example, if you commonly search only for `foo=1`, but `1` occurs in many events that do *not* have `foo=1`, you might want to add `foo` to the list of fields extracted by Splunk at index time.

In general, you should try to extract your fields at search time. For more information see [About fields](#) in the Knowledge Manager manual.

Caution: Do not add custom fields to the set of **default fields** that Splunk automatically extracts and indexes at **index time** unless absolutely necessary. This includes fields such as `timestamp`, `punct`, `host`, `source`, and `sourcetype`. Adding to this list of fields can negatively impact indexing performance and search times, because each indexed field increases the size of the searchable index. Indexed fields are also less flexible--whenever you make changes to your set of fields, you must re-index your entire dataset. For more information, see [Index time versus search time](#) in the [Managing Indexers and Clusters](#) manual.

Define additional indexed fields

Define additional indexed fields by editing `props.conf`, `transforms.conf`, and `fields.conf`.

Edit these files in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For more information on configuration files in general, see [About configuration files](#) in the Admin manual.

Where to put the configuration changes in a distributed environment

If you have a **distributed search** deployment, processing is split between search peers (indexers) and a search head. You must deploy the changes as follows:

- Deploy the `props.conf` and `transforms.conf` changes to each of the search peers.
- Deploy the `fields.conf` changes to the search head.

Note: If you are employing heavy forwarders in front of your search peers, the props and transforms processing takes place on the forwarders, not the search peers. Therefore, you must deploy the props and transforms changes to the forwarders, not the search peers.

For details on Splunk Enterprise distributed components, read Scale your deployment with Splunk Enterprise components in the *Distributed Deployment Manual*.

For details on where you need to put configuration settings, read Configuration parameters and the data pipeline in the *Admin Manual*.

Field name syntax restrictions

Splunk only accepts field names that contain alpha-numeric characters or an underscore:

- Valid characters for field names are **a-z, A-Z, 0-9**, or `_`.
- Field names cannot begin with **0-9** or `_`. Leading underscores are reserved for Splunk's internal variables.
- International characters are not allowed.

Add a regex stanza for the new field to transforms.conf

Follow this format when you define an index-time field transform in `transforms.conf` (Note: Some of these attributes, such as `LOOKAHEAD` and `DEST_KEY`, are only required for certain use cases):

```
[<unique_transform_stanza_name>]
REGEX = <regular_expression>
FORMAT = <your_custom_field_name>::$1
WRITE_META = [true|false]
DEST_KEY = <KEY>
DEFAULT_VALUE = <string>
```

```
SOURCE_KEY = <KEY>
REPEAT_MATCH = [true|false]
LOOKAHEAD = <integer>
```

Note the following:

- The `<unique_stanza_name>` is required for all transforms, as is the `REGEX`.
- `REGEX` is a regular expression that operates on your data to extract fields.
 - ◆ Name-capturing groups in the `REGEX` are extracted directly to fields, which means that you don't have to specify a `FORMAT` for simple field extraction cases.
 - ◆ If the `REGEX` extracts both the field name *and* its corresponding value, you can use the following special capturing groups to skip specifying the mapping in the `FORMAT` attribute:

```
_KEY_<string>, _VAL_<string>
```

- For example, the following are equivalent:

Using `FORMAT`:

```
REGEX = ([a-z]+)=([a-z]+)
FORMAT = $1::$2
```

Not using `FORMAT`:

```
REGEX = (?<_KEY_1>[a-z]+)=(?<_VAL_1>[a-z]+)
```

- `FORMAT` is optional. Use it to specify the format of the field/value pair(s) that you are extracting, including any field names or values that you want to add. You don't need to specify the `FORMAT` if you have a simple `REGEX` with name-capturing groups.
- `FORMAT` behaves differently depending on whether the extraction takes place at search time or index time.
 - ◆ For index-time transforms, you use `$n` to specify the output of each `REGEX` match (for example, `$1`, `$2`, and so on).
 - ◆ If the `REGEX` does not have `n` groups, the matching fails.
 - ◆ `FORMAT` defaults to `<unique_transform_stanza_name>::$1`.
 - ◆ The special identifier `$0` represents what was in the `DEST_KEY` before the `REGEX` was performed (in the case of index-time field extractions the `DEST_KEY` is `_meta`). For more information, see "How Splunk builds indexed fields," below.

- ◆ For index-time field extractions, you can set up `FORMAT` in several ways. It can be a `<field-name>::<field-value>` setup like:

```
FORMAT = field1::$1 field2::$2 (where the REGEX extracts field values for captured groups "field1" and "field2")
```

or:

```
FORMAT = $1::$2 (where the REGEX extracts both the field name and the field value)
```

However you can also set up index-time field extractions that create concatenated fields:

```
FORMAT = ipaddress::$1.$2.$3.$4
```

When you create concatenated fields with `FORMAT`, it's important to understand that `$` is the only special character. It is treated as a prefix for regex capturing groups **only** if it is followed by a number and **only** if that number applies to an existing capturing group.

So if your regex has only one capturing group and its value is `bar`, then:

```
FORMAT = foo$1 would yield foobar
FORMAT = foo$bar would yield foo$bar
FORMAT = foo$1234 would yield foo$1234
FORMAT = foo$1\2 would yield foobar\2
```

- `WRITE_META = true` writes the extracted field name and value to `_meta`, which is where Splunk stores indexed fields. This attribute setting is required for all index-time field extractions, except for those where `DEST_KEY = _meta` (see the discussion of `DEST_KEY`, below).
 - ◆ For more information about `_meta` and its role in indexed field creation, see "How Splunk builds indexed fields," below.
- `DEST_KEY` is required for index-time field extractions where `WRITE_META = false` or is not set. It specifies where Splunk sends the results of the `REGEX`.
 - ◆ For index-time searches, `DEST_KEY = _meta`, which is where Splunk stores indexed fields. For other possible `KEY` values see the `transforms.conf` page in this manual.
 - ◆ For more information about `_meta` and its role in indexed field creation, see [How Splunk builds indexed fields](#), below.

- ◆ When you use `DEST_KEY = _meta` you should also add `$0` to the start of your `FORMAT` attribute. `$0` represents the `DEST_KEY` value before Splunk performs the `REGEX` (in other words, `_meta`).
- ◆ **Note:** The `$0` value is in no way derived *from* the `REGEX`.
- `DEFAULT_VALUE` is optional. The value for this attribute is written to `DEST_KEY` if the `REGEX` fails.
 - ◆ Defaults to empty.
- `SOURCE_KEY` is optional. You use it to identify a `KEY` whose values the `REGEX` should be applied to.
 - ◆ By default, `SOURCE_KEY = _raw`, which means it is applied to the entirety of all events.
 - ◆ Typically used in conjunction with `REPEAT_MATCH`.
 - ◆ For other possible `KEY` values see the `transforms.conf` page in this manual.
- `REPEAT_MATCH` is optional. Set it to `true` to run the `REGEX` multiple times on the `SOURCE_KEY`.
 - ◆ `REPEAT_MATCH` starts wherever the last match stopped and continues until no more matches are found. Useful for situations where an unknown number of field/value matches are expected per event.
 - ◆ Defaults to `false`.
- `LOOKAHEAD` is optional. Use it to specify how many characters to search into an event.
 - ◆ Defaults to 4096. You might want to increase your `LOOKAHEAD` value if you have events with line lengths longer than 4096 characters.
 - ◆ Specifically, if the text you need to match is past this number of characters, you will need to increase this value.
 - ◆ Be aware, however, that complex regexes can have very high costs when scanning larger text segments. The speed may fall off quadratically or worse when using multiple greedy branches or lookaheads / lookbehinds.

Note: For a primer on regular expression syntax and usage, see Regular-Expressions.info. You can test regexes by using them in searches with the `rex` search command. Splunk also maintains a list of useful third-party tools for writing and testing regular expressions.

Note: The capturing groups in your regex must identify field names that follow [field name syntax restrictions](#). They can only contain ASCII characters (a-z, A-Z, 0-9 or `_`). International characters will not work.

Link the new field to props.conf

To `props.conf`, add the following lines:

```
[<spec>]
TRANSFORMS-<class> = <unique_stanza_name>
```

Note the following:

- `<spec>` can be:
 - ♦ `<sourcetype>`, the sourcetype of an event.
 - ♦ `host::<host>`, where `<host>` is the host for an event.
 - ♦ `source::<source>`, where `<source>` is the source for an event.
 - ♦ **Note:** You can use regex-type syntax when setting the `<spec>`. Also, source and source type stanzas match in a case-sensitive manner while host stanzas do not. For more information, see the `props.conf` spec file.
- `<class>` is a unique literal string that identifies the namespace of the field (key) you are extracting. **Note:** `<class>` values *do not* have to follow [field name syntax restrictions](#) (see above). You can use characters other than a-z, A-Z, and 0-9, and spaces are allowed.
- `<unique_stanza_name>` is the name of your stanza from `transforms.conf`.

Note: For index-time field extraction, `props.conf` uses `TRANSFORMS-<class>`, as opposed to `EXTRACT-<class>`, which is used for configuring search-time field extraction.

Add an entry to fields.conf for the new field

Add an entry to `fields.conf` for the new indexed field:

```
[<your_custom_field_name>]
INDEXED=true
```

Note the following:

- `<your_custom_field_name>` is the name of the custom field you set in the unique stanza that you added to `transforms.conf`.
- Set `INDEXED=true` to indicate that the field is indexed.

Note: If a field of the same name is extracted at search time, you **must** set `INDEXED=false` for the field. In addition, you must *also* set `INDEXED_VALUE=false` if

events exist that have values of that field that are not pulled out at index time, but which *are* extracted at search time.

For example, say you're performing a simple `<field>::1234` extraction at index time. This could work, but you would have problems if you also implement a search-time field extraction based on a regex like `A(\d+)B`, where the string `A1234B` yields a value for that field of `1234`. This would turn up events for `1234` at search time that Splunk would be unable to locate at index time with the `<field>::1234` extraction.

Restart Splunk for your changes to take effect

Changes to configuration files such as `props.conf` and `transforms.conf` won't take effect until you shut down and restart Splunk on all affected components.

How Splunk builds indexed fields

Splunk builds indexed fields by writing to `_meta`. Here's how it works:

- `_meta` is modified by all matching transforms in `transforms.conf` that contain either `DEST_KEY = _meta` or `WRITE_META = true`.
- Each matching transform can overwrite `_meta`, so use `WRITE_META = true` to append `_meta`.
 - ◆ If you don't use `WRITE_META`, then start your `FORMAT` with `$0`.
- After `_meta` is fully built during parsing, Splunk interprets the text in the following way:
 - ◆ The text is broken into units; each unit is separated by whitespace.
 - ◆ Quotation marks (" ") group characters into larger units, regardless of whitespace.
 - ◆ Backslashes (\) immediately preceding quotation marks disable the grouping properties of quotation marks.
 - ◆ Backslashes preceding a backslash disable that backslash.
 - ◆ Units of text that contain a double colon (::) are turned into extracted fields. The text on the left side of the double colon becomes the field name, and the right side becomes the value.

Note: Indexed fields with regex-extracted values containing quotation marks will generally not work, and backslashes might also have problems. Fields extracted at search time do not have these limitations.

Here's an example of a set of index-time extractions involving quotation marks and backslashes to disable quotation marks and backslashes:

```
WRITE_META = true
FORMAT = field1::value field2::"value 2" field3::"a field with a \"
quotation mark" field4::"a field which
ends with a backslash\"
```

When Splunk creates field names

Remember: When Splunk creates field names, it applies [field name syntax restrictions](#) to them.

1. All characters that are not in a-z,A-Z, and 0-9 ranges are replaced with an underscore (_).
2. All leading underscores are removed. In Splunk, leading underscores are reserved for **internal fields**.

Index-time field extraction examples

Here are a set of examples of configuration file setups for index-time field extractions.

Define a new indexed field

This basic example creates an indexed field called `err_code`.

transforms.conf

In `transforms.conf` add:

```
[netscreen-error]
REGEX = device_id=[\w+\] (?<err_code>[^:]+)
FORMAT = err_code::"$1"
WRITE_META = true
```

This stanza takes `device_id=` followed with a word within brackets and a text string terminating with a colon. The source type of the events is `testlog`.

Comments:

- The `FORMAT =` line contains the following values:
 - ♦ `err_code::` is the name of the field.
 - ♦ `$1` refers to the new field written to the index. It is the value extracted by `REGEX`.

- `WRITE_META = true` is an instruction to write the content of `FORMAT` to the index.

props.conf

Add the following lines to `props.conf`:

```
[testlog]
TRANSFORMS-netscreen = netscreen-error
```

fields.conf

Add the following lines to `fields.conf`:

```
[err_code]
INDEXED=true
```

Restart Splunk for your configuration file changes to take effect.

Define two new indexed fields with one regex

This example creates two indexed fields called `username` and `login_result`.

transforms.conf

In `transforms.conf` add:

```
[ftpd-login]
REGEX = Attempt to login by user: (.*) : login (.*)\.
FORMAT = username::"$1" login_result::"$2"
WRITE_META = true
```

This stanza finds the literal text `Attempt to login by user:`, extracts a username followed by a colon, and then the result, which is followed by a period. A line might look like:

```
2008-10-30 14:15:21 mightyhost awesomeftpd INFO Attempt to login by
user: root: login FAILED.
```

props.conf

Add the following lines to `props.conf`:

```
[ftpd-login]
TRANSFORMS-login = ftpd-login
```

fields.conf

Add the following lines to `fields.conf`:

```
[username]
INDEXED=true

[login_result]
INDEXED=true
```

Restart Splunk for your configuration file changes to take effect.

Concatenate field values from event segments at index time

This example shows you how an index-time transform can be used to extract separate segments of an event and combine them to create a single field, using the `FORMAT` option.

Let's say you have the following event:

```
20100126 08:48:49 781 PACKET 078FCFD0 UDP Rcv 127.0.0.0 8226 R Q [0084 A
NOERROR] A (4)www(8)google(3)com(0)
```

Now, what you want to do is get `(4)www(8)google(3)com(0)` extracted as a value of a field named `dns_requestor`. But you don't want those garbage parentheses and numerals, you just want something that looks like `www.google.com`. How do you achieve this?

transforms.conf

You would start by setting up a transform in `transforms.conf` named `dnsRequest`:

```
[dnsRequest]
REGEX = UDP[^\(]+\(\d\) (\w+)\(\d\) (\w+)\(\d\) (\w+)\(\d\) (\w+)
FORMAT = dns_requestor::$1.$2.$3
```

This transform defines a custom field named `dns_requestor`. It uses its `REGEX` to pull out the three segments of the `dns_requestor` value. Then it uses `FORMAT` to order those segments with periods between them, like a proper URL.

Note: This method of concatenating event segments into a complete field value is something you can *only* perform with index-time extractions; search-time extractions have practical restrictions that prevent it. If you find that you must use `FORMAT` in this manner, you will have to create a new indexed field to do it.

props.conf

Then, the next step would be to define a field extraction in `props.conf` that references the `dnsRequest` transform and applies it to events coming from the `server1` source type:

```
[server1]
TRANSFORMS-dnsExtract = dnsRequest
```

fields.conf

Finally, you would enter the following stanza in `fields.conf`:

```
[dns_requestor]
INDEXED = true
```

Restart Splunk for your configuration file changes to take effect.

Extract fields from files with structured data

Many structured data files, such as comma-separated value (CSV) files and Internet Information Server (IIS) web server logs, have information in the file header that Splunk Enterprise can use to extract fields during index-time event processing. You can configure Splunk Enterprise to automatically extract these fields.

For example, a CSV file starts with a header row that contains column headers for the values in subsequent rows, for example:

```
host,status,message,"start date"
srv1.splunk.com,error,"No space left on device",2013-06-10T06:35:00
srv2.splunk.com,ok,-,2013-06-11T06:00:00
```

- For information on how to add data to your Splunk Enterprise instance, see [How do you want to add data?](#)

- For information on how to set source types when importing structured data files, see [The "Set source type" page](#).
- For information on how to adjust timestamps when you preview how Splunk Enterprise will index your data, see [Adjust time stamps and event breaks](#).
- For more general information about configuration files, see About configuration files in the Admin manual.

Use Splunk Web to extract fields from structured data files

When you upload or monitor a structured data file, Splunk Enterprise loads the "Set Source type" page. This page lets you preview how Splunk Enterprise plans to index the data. See [The 'Set Source type' page](#).

To extract fields from structured data files with Splunk Web:

1. Choose the method that you want to add data.
2. Splunk Enterprise loads the "Set Source type" page. It sets the source type of the data based on its interpretation of that data. For example, if you upload a CSV file, it sets the source type to `csv`.
3. Review the events as Splunk Enterprise displays them in the preview pane on the right side of the page. Splunk Enterprise displays the events based on the current source type.
4. If the events appear to be formatted correctly, click "Next" to proceed to the "Modify input settings" page. Otherwise, change how Splunk Enterprise formats the events by modifying the time stamp, event breaking, and delimited settings.
5. Iterate until the previewed events look the way that you want.
6. If you don't want to save the settings as a new source type, proceed to Step 4. Otherwise, click the **Save As** button to save the settings as a new source type.
7. In the dialog that appears, type in a name and description for the new source type.
8. Select the category for the source type by selecting the category you want from the "Category" drop-down.
9. Select the application context that the new source type should apply to by choosing from the entries in the "App" drop-down.

10. Click "Save" to save the source type.

11. Proceed to Step 4a to proceed to the "Modify input settings" page.

Use configuration files to enable automatic header-based field extraction

You can also use a combination of `inputs.conf` and `props.conf` to extract fields from structured data files. Edit these files in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in

`$SPLUNK_HOME/etc/apps/<app_name>/local`. `Inputs.conf` specifies the files you want to monitor and the source type Splunk Enterprise should use to monitor them, and `props.conf` defines the source types themselves.

You must restart Splunk Enterprise for any changes that you make to `inputs.conf` and `props.conf` to take effect.

Props.conf attributes for structured data

Splunk Enterprise includes the following attributes in `props.conf` for working with files that contain headers. For additional attributes in `props.conf`, review the `props.conf` specification file.

Attribute	Description	Default
<code>INDEXED_EXTRACTIONS = {CSV W3C TSV PSV JSON}</code>	<p>Specifies the type of file and the extraction and/or parsing method that Splunk Enterprise should use on the file.</p> <p>Note: If you set <code>INDEXED_EXTRACTIONS=JSON</code>, check that you have not also set <code>KV_MODE = json</code> for the same source type, as this will make Splunk Enterprise extract the JSON fields twice, once at index time and again at search time.</p>	n/a (not set)
<code>PREAMBLE_REGEX</code>	<p>Some files contain preamble lines. This attribute specifies a regular expression which allows Splunk to ignore these preamble lines,</p>	n/a

	based on the pattern specified.	
FIELD_HEADER_REGEX	A regular expression that specifies a pattern for prefixed header line. Splunk Enterprise looks for the first line that matches that regex and parses it as header fields. Note that the actual header starts after the matching pattern which is not included in the parsed header fields. You can specify special characters in this attribute.	n/a
FIELD_DELIMITER	Specifies which character delimits or separates fields in the monitored file or source. You can specify special characters in this attribute.	n/a
FIELD_QUOTE	Specifies the character to use for quotes in the specified file or source. You can specify special characters in this attribute.	n/a
HEADER_FIELD_DELIMITER	Specifies which character delimits or separates field names in the header line. You can specify special characters in this attribute. If HEADER_FIELD_DELIMITER is not specified, FIELD_DELIMITER applies to the header line.	n/a
HEADER_FIELD_QUOTE	Specifies which character is used for quotes around field names in the header line. You can specify special characters in this attribute. If HEADER_FIELD_QUOTE is not specified, FIELD_QUOTE applies to the header line.	n/a
HEADER_FIELD_LINE_NUMBER	Specifies the line number of the line within the file that contains the header fields. If set to 0, Splunk attempts to locate the header fields within the file automatically.	0

<code>TIMESTAMP_FIELDS = field1,field2,...,fieldn</code>	Some CSV and structured files have their timestamp encompass multiple fields in the event separated by delimiters. This attribute tells Splunk to specify all such fields which constitute the timestamp in a comma-separated fashion.	Splunk Enterprise tries to automatically extract the timestamp of the event.
<code>FIELD_NAMES</code>	Some CSV and structured files might have missing headers. This attribute tells Splunk to specify the header field names directly.	n/a
<code>MISSING_VALUE_REGEX</code>	If Splunk Enterprise finds the specified regular expression in the structured data file, it considers the value for the field in the row to be empty.	n/a

Special characters or values are available for some attributes

You can use special characters or values such as spaces, vertical and horizontal tabs, and form feeds in some attributes. The following table lists these characters:

Special value	Props.conf representation
form feed	<code>\f</code>
space	<code>space</code> or <code>' '</code>
horizontal tab	<code>\t</code> or <code>tab</code>
vertical tab	<code>\v</code>
whitespace	<code>whitespace</code>
none	<code>none</code> or <code>\0</code>
file separator	<code>fs</code> or <code>\034</code>
group separator	<code>gs</code> or <code>\035</code>
record separator	<code>rs</code> or <code>\036</code>

unit separator	us or \037
----------------	------------

You can use these special characters for the following attributes only:

- FIELD_DELIMITER
- FIELD_HEADER_REGEX
- FIELD_QUOTE

Edit configuration files to create and reference source types

To create and reference the new source types to extract files with headers:

1. Using a text editor, open the file `props.conf` in the appropriate location as described in [Enable automatic header-based field extraction](#) earlier in this topic.

Note: If the `props.conf` file does not exist, you must create it.

2. Define a new sourcetype by creating a stanza which tells Splunk Enterprise how to extract the file header and structured file data, using the attributes described above. For example:

```
[HeaderFieldsWithFewEmptyFieldNamesWithSpaceDelim]
FIELD_DELIMITER=,
HEADER_FIELD_DELIMITER=\s
FIELD_QUOTE="
```

Note: You can define as many stanzas - and thus, as many sourcetypes - as you like in the file.

3. Save the `props.conf` file and close it.

4. Create a file `inputs.conf` in the same directory, if it does not already exist.

5. Open the file for editing.

6. Add a stanza which represents the file or files that you want Splunk Enterprise to extract file header and structured data from. For example:

```
[monitor:///opt/test/data/StructuredData/HeaderFieldsWithFewEmptyFieldNamesWithSpaceDelim]
sourcetype=HeaderFieldsWithFewEmptyFieldNamesWithSpaceDelim
```

Note: You can add as many stanzas as you wish for files or directories from which you want to extract header and structured data.

7. Save the `inputs.conf` file and close it.
8. Restart Splunk Enterprise for the changes to take effect.

Forward data extracted from structured data files

You can also forward fields extracted from a structured data file to another Splunk Enterprise instance. To do this, you must use either a full Splunk Enterprise instance that has been configured as a heavy forwarder, or a universal forwarder.

To forward fields extracted from structured data files:

1. Configure the Splunk Enterprise instance that monitors the files to forward data to another Splunk Enterprise instance.
2. On the instance that should receive the data, configure Splunk Enterprise to be a receiver.
3. On the monitoring instance, configure `props.conf` and `inputs.conf` to properly handle event breaking and time stamps for your data. You can do this in one of two ways:
 - To use Splunk Web, follow the instructions in [Use Splunk Web to extract fields from structured data files](#) earlier in this topic.
 - To use configuration files, follow the instructions in [Edit configuration files to create and reference sourcetypes](#) earlier in this topic.
4. Optionally, if you need to transform this data in any way prior to indexing it, edit `transforms.conf`.
- 5 Restart Splunk Enterprise on the receiving instance.
6. Restart Splunk Enterprise on the monitoring instance.
7. On the receiving instance, use the Search app to confirm that Splunk Enterprise has extracted the fields from the structured data files and indexed them properly.

Caveats to extracting fields from structured data files

Splunk Enterprise does not parse structured data that has been forwarded to an indexer

When you forward structured data to an indexer, Splunk Enterprise does not parse this data once it arrives at the indexer, even if you have configured `props.conf` on that indexer with `INDEXED_EXTRactions`. Forwarded data skips the following pipelines on the indexer, which precludes any parsing of that data on the indexer:

- parsing
- merging
- typing

The forwarded data must arrive at the indexer already parsed.

Field extraction settings for forwarded structured data must be configured on the forwarder

If you want to forward fields that you extract from structured data files to another Splunk Enterprise instance, you must configure the `props.conf` settings that define the field extractions on the forwarder that sends the data. This includes configuration of `INDEXED_EXTRactions` and any other parsing, filtering, anonymizing, and routing rules. Performing these actions on the instance that indexes the data will have no effect, as the forwarded data must arrive at the indexer already parsed.

When you use Splunk Web to modify event break and time stamp settings, it records all of the proposed changes as a stanza for `props.conf`. You can find those settings in the "Advanced" tab on the "Set Source type" page.

Use the "Copy to clipboard" link in the "Advanced" tab to copy the proposed changes to `props.conf` to the system clipboard. You can then paste this stanza into `props.conf` in a text editor on Splunk Enterprise instances that monitor and forward similar files.

Splunk Enterprise only indexes header fields whose rows contain data

When Splunk Enterprise extracts header fields from structured data files, it only extracts those fields where data is present in at least one row. If the header field has no data in any row, Splunk skips that field, and does not index it. Take, for example, the following csv file:

```
header1,header2,header3,header4,header5
```

```
one,1,won,,111
two,2,too,,222
three,3,thri,,333
four,4,fore,,444
five,5,faiv,,555
```

When Splunk Enterprise reads this file, it notes that the rows in the `header4` column are all empty, and does not index that header field or any of the rows in it. This means that neither `header4` nor any of the data in its row can be searched for in the index.

If, however, the `header4` field contains rows with empty strings (for example, ""), Splunk *does* index the field and all the rows underneath.

Splunk Enterprise does not support renaming of header fields mid-file

Some software, such as Internet Information Server, supports the renaming of header fields in the middle of the file. Splunk does not recognize changes such as this. If you attempt to index a file which has header fields renamed within the file, Splunk does not index the renamed header field.

Example configuration and data files

Following are an example `inputs.conf` and `props.conf` to give you an idea of how to use the file header extraction attributes.

To extract the data locally, edit `inputs.conf` and `props.conf` to define inputs and sourcetypes for the structured data files, and use the attributes described above to tell Splunk Enterprise how to deal with the files. To forward this data to another Splunk instance, edit `inputs.conf` and `props.conf` on the forwarding instance, and `props.conf` on the receiving instance.

Inputs.conf

```
[monitor:///opt/test/data/StructuredData/CSVWithFewHeaderFieldsWithoutAnyValues.csv]
sourcetype=CSVWithFewHeaderFieldsWithoutAnyValues
```

```
[monitor:///opt/test/data/StructuredData/VeryLargeCSVFile.csv]
sourcetype=VeryLargeCSVFile
```

```
[monitor:///opt/test/data/StructuredData/UselessLongHeaderToBeIgnored.log]
sourcetype=UselessLongHeaderToBeIgnored
```

```
[monitor:///opt/test/data/StructuredData/HeaderFieldsWithFewEmptyFieldNamesWithSpaceDelim]
sourcetype=HeaderFieldsWithFewEmptyFieldNamesWithSpaceDelim
```

Props.conf

```
[VeryLargeCSVFile]
FIELD_DELIMITER=,
```

```
[HeaderFieldsWithFewEmptyFieldNamesWithSpaceDelim]
FIELD_DELIMITER=,
HEADER_FIELD_DELIMITER=\s
FIELD_QUOTE="
```

Sample files

Note: You might need to scroll right quite a bit to see all of the content.

```
vqmcallsid,serialnumber,vqmvavgjbenvdelay,vqmvavgjbenvnegdelta,vqmvavgjbenvpodelta,  
99152,CFG0730084,-3,-2,356,64000,1,280,14,14.29,36,3499,201000,BW163736844290611-173170  
12:37:37.292,0,4.68,1.43,0.19,0,0,0,0,52,60,15,17,60,10,0,Loopback,0.48,48,46,0,30,1334  
0/1,2,0,54,80,80,18500,6096147089,48,1,0,2011-06-29  
12:41:47.303,2011-06-29 12:41:47.303  
99154,CFG0730084,-3,-1,251,64000,4,195,9,20.52,28,3494,359000,BW16350227029061159456629  
12:35:02.324,0,2.88,1.11,3.44,0,0,0,0,40,40,26,24,50,10,0,Loopback,0.31,54,46,0,31,2455  
0/1,2,0,48,60,70,30400,6096147089,54,1,0,2011-06-29  
12:41:47.342,2011-06-29 12:41:47.342
```

VeryLargeCSVFile.csv

IncidentNum,Category,Descript,DayOfWeek,Date,Time,PdDistrict,Resolution,Location,X,Y
030203898,FRAUD,"FORGERY, CREDIT
CARD",Tuesday,02/18/2003,16:30,NORTHERN,NONE,2800 Block of VAN NESS
AV,-122.424612993055,37.8014488257836
000038261,WARRANTS,WARRANT
ARREST,Thursday,04/17/2003,22:45,NORTHERN,"ARREST, BOOKED",POLK ST /
SUTTER ST,-122.420120319211,37.7877570602182
030203901,LARCENY/THEFT,GRAND THEFT
PICKPOCKET,Tuesday,02/18/2003,16:05,NORTHERN,NONE,VAN NESS AV /
MCALLISTER ST,-122.42025048261,37.7800745746105
030203923,DRUG/NARCOTIC,SALE OF BASE/ROCK
COCAINE,Tuesday,02/18/2003,17:00,BAYVIEW,"ARREST, BOOKED",1600 Block of
KIRKWOOD AV,-122.390718076188,37.7385560584619
030203923,OTHER
OFFENSES,CONSPIRACY,Tuesday,02/18/2003,17:00,BAYVIEW,"ARREST,
BOOKED",1600 Block of KIRKWOOD AV,-122.390718076188,37.7385560584619
030203923,OTHER OFFENSES,PROBATION
VIOLATION,Tuesday,02/18/2003,17:00,BAYVIEW,"ARREST, BOOKED",1600 Block
of KIRKWOOD AV,-122.390718076188,37.7385560584619

UselessLongHeaderToBeIgnored.log

***** Start Display Current Environment *****
WebSphere Platform 6.1 [ND 6.1.0.21 cf210844.13] running with process
name sammys_cell_A\fsgwsws189Node_A\sammys_A_c01_s189_m06 and process id
17904
Detailed IFix information: ID: 6.1.0-WS-WASSDK-AixPPC32-FP0000021
BuildVrsn: null Desc: Software Developer Kit 6.1.0.21
ID: 6.1.0-WS-WAS-AixPPC32-FP0000021 BuildVrsn: null Desc: WebSphere
Application Server 6.1.0.21
ID: 6.1.0-WS-WASSDK-AixPPC32-FP0000019 BuildVrsn: null Desc: Software
Developer Kit 6.1.0.19
ID: 6.1.0-WS-WAS-AixPPC32-FP0000019 BuildVrsn: null Desc: WebSphere
Application Server 6.1.0.19
ID: sdk.FP61021 BuildVrsn: null Desc: WebSphere Application Server
6.1.0.21
ID: sdk.FP61019 BuildVrsn: null Desc: WebSphere Application Server
6.1.0.19
ID: was.embed.common.FP61021 BuildVrsn: null Desc: WebSphere
Application Server 6.1.0.21
ID: was.embed.FP61021 BuildVrsn: null Desc: WebSphere Application
Server 6.1.0.21

HeaderFieldsWithFewEmptyFieldNamesWithSpaceDelim.csv

"Field 1" "Field 3" "Field 4" "Field 6"
Value11,Value12,Value13,Value14,Value15,Value16
Value21,Value22,Value23,Value24,Value25

```
Value31,Value32,Value33,Value34,Value35, Value36  
FieldHeaderRegex.log
```

```
Garbage  
Garbage  
Garbage  
Ignore_This_Stuff: Actual_Header1 Actual_Header2
```

Answers

Have questions? Visit [Splunk Answers](#) and see what questions and answers the Splunk community has around extracting fields.

Configure host values

About hosts

The **host** field value of an event is the name of the physical device from which the event originates. Because it is a **default field**, which means that Splunk Enterprise assigns a host to every event it indexes, you can use it to search for all events that have been generated by a particular host.

The host value is typically the hostname, IP address, or fully qualified domain name of the network host on which the event originated.

How Splunk Enterprise assigns the host value

Splunk Enterprise assigns a host value to each event by examining settings in the following order and using the first host setting it encounters:

1. Any event-specific host assignment that you specify in `transforms.conf`.
2. The default host value for the input that created the event, if any.
3. The default host value for the Splunk instance (indexer or forwarder) that initially consumes the data.

An overview of these assignment methods and their use cases follows. Subsequent topics describe the methods in greater detail.

The default host value

If no other host rules are specified for a source, Splunk Enterprise assigns the host field a default value that applies to all data coming into the instance from any input. The default host value is the hostname or IP address of the Splunk Enterprise instance (indexer or forwarder) initially consuming the data. When the Splunk Enterprise instance runs on the server where the event occurred, this is correct and no manual intervention is required.

For more information, see [Set a default host for a Splunk Enterprise server](#) in this manual.

The default host for a file or directory input

If you run Splunk Enterprise on a central log archive, or you are working with files forwarded from other hosts in your environment, you might need to override the default host assignment for events coming from particular inputs.

There are two methods for assigning a host value to data received through a particular input. You can define a static host value for all data coming through a specific input, or you can have Splunk Enterprise dynamically assign a host value to a portion of the path or filename of the source. The latter method can be helpful when you have a directory structure that segregates each host's log archive in a different subdirectory.

For more information, see [Set a default host for a file or directory input](#) in this manual.

Event-specific assignments

Some situations require you to assign host values by examining the event data. For example, If you have a central log host sending events to Splunk Enterprise, you might have several host servers that feed data to that main log server. To ensure that each event has the host value of its originating server, you need to use the event's data to determine the host value.

For more information, see [Set host values based on event data](#) in this manual.

Handle incorrectly-assigned host values

If your event data gets tagged with the wrong host value, don't worry. There are a number of ways to fix or work around the problem.

For details, see [Change host values after indexing](#) in this manual.

Tag host values

You can tag host values to aid in the execution of robust searches. Tags enable you to cluster groups of hosts into useful, searchable categories.

For details, see [About tags and aliases](#) in the Knowledge Manager manual.

Set a default host for a Splunk Enterprise server

An event host value is the IP address, host name, or fully qualified domain name of the physical device on the network from which the event originates. Because Splunk Enterprise assigns a `host` value at index time for every event it indexes, host value searches enable you to easily find data originating from a specific device.

Default host assignment

If you have not specified other host rules for a source (using the information in subsequent topics in this chapter), the default host value for an event is the hostname or IP address of the server running the Splunk instance (forwarder or indexer) consuming the event data. When the event originates on the server on which the Splunk Enterprise instance is running, that host assignment is correct and there's no need to change anything. However, if all your data is being forwarded from a different host or if you're bulk-loading archive data, you might want to change the default host value for that data.

To set the default value of the host field, you can use Splunk Web or edit `inputs.conf`.

Set the default host value using Splunk Web

1. In Splunk Web, click **Settings**.
3. On the Settings page, click **General settings**.
4. On the General settings page, scroll down to the **Index settings** section and change the **Default host name**.
5. Save your changes.

This sets the default value of the host field for all events coming into that Splunk Enterprise instance. You can override the value for individual sources or events, as described later in this chapter.

Set the default host value using inputs.conf

The default host assignment is set in `inputs.conf` during installation. You can modify the host value by editing that file in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`.

Splunk Enterprise places the host assignment in the `[default]` stanza.

This is the format of the default host assignment in `inputs.conf`:

```
[default]
host = <string>
```

Set `<string>` to your chosen default host value. `<string>` defaults to the IP address or domain name of the host where the data originated.

Warning: Do *not* put quotes around the `<string>` value: `host=foo`, not `host="foo"`.

Restart Splunk Enterprise to enable any changes you make to `inputs.conf`.

Note: By default, the `host` attribute is set to the variable `$decideOnStartup`, which means that it's set to the hostname of the machine `splunkd` is running on. The `splunkd` daemon re-interprets the value each time it starts up.

Override the default host value for data received from a specific input

If you are running Splunk Enterprise on a central log archive, or you are working with files forwarded from other hosts in your environment, you might need to override the default host assignment for events coming from particular inputs.

There are two methods for assigning a host value to data received through a particular input. You can define a static host value for all data coming through a specific input, or you can have Splunk Enterprise dynamically assign a host value to a portion of the path or filename of the source. The latter method can be helpful when you have a directory structure that segregates each host's log archive in a different subdirectory.

For more information, see [Set a default host for an file or directory input](#) in this manual.

Override the default host value using event data

Some situations require you to assign host values by examining the event data. For example, If you have a central log host sending events to Splunk Enterprise, you might have several host servers feeding data to that main log server. To ensure that each event has the host value of its originating server, you need to use the event's data to determine the host value.

For more information, see [Set host values based on event data](#) in this manual.

Set a default host for a file or directory input

You can set a host value for all data from a particular file or directory input. You can set the host statically or dynamically:

- If you **statically** set the host value, Splunk Enterprise assigns the same host to every event that comes in through a designated file or directory input.
- If you **dynamically** set the host value, Splunk Enterprise extracts the host name from a portion of the source input using a regular expression or segment of the source's full directory path.

You can also assign host values to events coming through a particular file or directory input based on their source or source type values (as well as other kinds of information). For more information, see [Set host values based on event data](#) in this manual.

Note: Splunk Enterprise currently does not enable the setting of default host values for event data received through TCP, UDP, or **scripted inputs**.

Statically set the default host value

This method applies a single default host value to each event received through a specific file or directory input.

Note: A static host value assignment only affects new data arriving through the input with which it's associated. You cannot assign a default host value to data that has already been indexed. Instead, you can tag the host value.

Use Splunk Web

You can define a host for a file or directory input whenever you add or edit an input of that type through the "Data inputs" page of Splunk Web's System interface.

To set the default host when creating a new input, see [Set a default host for a new input](#).

1. Click **Settings > Data Inputs**.

2. Click **Files & Directories**.
3. On the Files & directories page, click the name of an existing input to update it.
4. In the **Host** section, select the "constant value" option from the **Set host** dropdown.
5. Enter the static host value for the input in the **Host field value** field.
6. Click **Save**.

For more information about inputs and input types, see [What Splunk Enterprise can monitor](#) in this manual.

Set a default host for a new input

The process to set a default host is different when you create a new input.

1. Click **Settings > Data Inputs**.
 2. Click **Files & Directories**.
 3. On the Files & directories page, click **New** to add an input.
 4. Specify the file or directory that you want to monitor, and specify any whitelists or blacklists.
 5. Click **Next**.
 6. (Optional) Set the source type for your new input.
- Note:** If you specified a directory, the "Set Sourcetype" page does not appear.
7. Click **Next**.
 8. On the **Input Settings** page, in the **Host** section, click the **Constant Value** button.
 9. In the **Host field value** field, enter the host name for the input.
 10. Click **Review** to continue to the Review page.
 11. Click **Submit** to create the input.

Edit inputs.conf

You can directly edit `inputs.conf` to specify a host value for a monitored file or directory input. Set the `host` attribute in the appropriate stanza.

```
[monitor://<path>]
host = <your_host>
```

Edit `inputs.conf` in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For more information on configuration files in general, see [About configuration files in the Admin manual](#).

For more information about inputs and input types, see [What Splunk Enterprise can monitor](#) in this manual.

Example of static host value assignment

This example covers any events coming in from `/var/log/httpd`. Any events coming from this input will receive a `host` value of `webhead-1`.

```
[monitor:///var/log/httpd]
host = webhead-1
```

Dynamically set the default host value

This method dynamically extracts the host value for a file or directory input, either from a segment of the source input path or from a regular expression. For example, if you want to index an archived directory and the name of each file in the directory contains relevant host information, you can extract this information and assign it to the `host` field.

Note: For a primer on regular expression syntax and usage, see [Regular-Expressions.info](#). You can test regular expressions by using them in searches with the `rex search` command. Splunk Enterprise also maintains a list of useful third-party tools for writing and testing regular expressions.

Use Splunk Web

1. Click **Settings > Data Inputs**.
2. Click **Files & Directories**.

3. On the Files & directories page, click the name of an existing input to update it.

4. In the **Host** section, select one of the following two options from the **Set host** dropdown:

- **regex on path** - Choose this option if you want to extract the host name with a regular expression. Then enter the regex for the host you want to extract in the **Regular expression** field.
- **segment in path** - Choose this option if you want to extract the host name from a segment in your data source's path. Then enter the segment number in the **Segment number** field. For example, if the path to the source is `/var/log/<host server name>` and you want the third segment (the host server name) to be the host value, enter "3".

5. Click **Save**.

Dynamically set a default host for a new input

The process to set a default host dynamically is different when you create a new input.

1. Click **Settings > Data Inputs**.

2. Click **Files & Directories**.

3. On the Files & directories page, click **New** to add an input.

4. Specify the file or directory that you want to monitor, and specify any whitelists or blacklists.

5. Click **Next**.

6. (Optional) Set the source type for your new input.

Note: If you specified a directory, the "Set Sourcetype" page does not appear.

7. Click **Next**.

8. On the **Input Settings** page, in the **Host** section, click either **Regular expression on path** or **Segment in path**.

9. If you chose **Regular expression on path**, enter a regular expression that Splunk Enterprise should use to extract the hostname from the source path in the "Regular expression" field. Otherwise, enter the number for the source path segment that Splunk Enterprise should use to determine the hostname in the "Segment Number" field.

10. Click **Review** to continue to the Review page.

11. Click **Submit** to create the input.

Edit inputs.conf

You can set up dynamic host extraction rules by directly configuring `inputs.conf`.

Edit `inputs.conf` in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For more information on configuration files in general, see [About configuration files](#) in the Admin manual.

Use the `host_regex` attribute to override the `host` field with a value extracted through a regular expression:

```
[monitor://<path>]
host_regex = <your_regular_expression>
```

The regular expression extracts the `host` value from the filename of each input. The first capturing group of the regular expression is used as the host.

Note: If the regular expression fails to match, the default `host` attribute is set as the host.

Use the `host_segment` to override the `host` field with a value extracted from a segment in your data source's path. For example, if the path to the source is `/var/log/<host server name>` and you want the third segment (the host server name) to be the host value, your input stanza would look like:

```
[monitor://var/log/]
host_segment = 3
```

Examples of dynamic host assignment

In this example, the regular expression assigns all events from `/var/log/foo.log` a host value of "foo":

```
[monitor://var/log]
host_regex = /var/log/(\w+)
```

This example assigns the host value to the third segment in the path
apache/logs:

```
[monitor://apache/logs/]
host_segment = 3
```

Caveats

There are some caveats to using the `host_segment` attribute in an `inputs.conf` stanza:

- You cannot simultaneously specify the `host_regex` and `host_segment` attributes in the same stanza.
- When you simultaneously specify a `host_segment` and `source` attribute in the same stanza, the behavior of the `host_segment` attribute changes:
 - ◆ If the value you specify for the source contains a / (forward slash), then Splunk Enterprise extracts the host value based on the segment number you specify in `host_segment`.
 - ◆ If `source` does not contain a /, or you specify a `host_segment` value that is larger than the number of segments available in `source`, then Splunk Enterprise cannot extract the host value, and instead uses the name of the host that extracted the data. See the following examples:

Example 1: Host name is server01, source path is `/mnt/logs/server01`, `inputs.conf` contains:

```
[monitor:///mnt/logs/]
host_segment = 3
```

In this case, Splunk Enterprise extracts the host name `server01` from the source path `/mnt/logs/server01/*` and sets the host field to this value.

Example 2: Host name is server01, source path is `/mnt/logs/server01`, `inputs.conf` contains:

```
[monitor:///mnt/logs/server01]
source = /mnt/logs/server01
```

```
host_segment = 3
```

In this case, Splunk Enterprise extracts the host name `server01` from the `source` attribute and sets the host field to this value.

Example 3: Host name is `server02`, source path is `/mnt/logs/server02`, `inputs.conf` contains:

```
[monitor:///mnt/logs/server02]
source = serverlogs
host_segment = 3
```

In this case, Splunk Enterprise uses `server02` as the host field because it cannot extract the host segment value from the specified `source`.

Note: Do not explicitly specify `source` unless absolutely necessary. Consider using source types, tagging, and search wildcards instead of overriding the default source value by specifying this attribute.

Set host values based on event data

Splunk Enterprise can assign host names to your events based on the data in those events. This topic shows you how to use event data to override default host assignments with `props.conf`, `transforms.conf`, and regular expressions.

For a primer on regular expression syntax and usage, see [Regular-Expressions.info](#). You can test regular expressions by using them in searches with the `rex` search command. The Splunk community wiki also has a list of useful third-party tools for writing and testing regular expressions.

Configuration

To configure per-event overrides, you need to create two stanzas, one in `transforms.conf` and another in `props.conf`. Edit these files in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For more information about configuration files in general, see [About configuration files](#) in the Admin manual.

transforms.conf

Create a stanza in `transforms.conf` that follows this syntax:

```
[<unique_stanza_name>]
REGEX = <your_regex>
FORMAT = host::$1
DEST_KEY = MetaData:Host
```

Note the following:

- `<unique_stanza_name>` should reflect that it involves a host value. You'll use this name later in the `props.conf` stanza.
- `<your_regex>` is a regular expression that identifies where in the event you want to extract the host value.
- `FORMAT = host::$1` writes the `REGEX` value into the `host::` field.

props.conf

Next, create a stanza in `props.conf` that references the `transforms.conf` stanza:

```
[<spec>]
TRANSFORMS-<class> = <unique_stanza_name>
```

Note the following:

- `<spec>` can be:
 - ♦ `<sourcetype>`, the source type of an event.
 - ♦ `host::<host>`, where `<host>` is the host value for an event.
 - ♦ `source::<source>`, where `<source>` is the source value for an event.
- `<class>` is any unique identifier that you want to give to your transform.
- `<unique_stanza_name>` is the name of the stanza you created in `transforms.conf`.

Example

Assume that you're starting with the following set of events from the `houseness.log` file. The host is in the third position ("fflanda", etc.).

```
41602046:53 accepted fflanda
41602050:29 accepted rhallen
41602052:17 accepted fflanda
```

First, create a new stanza in `transforms.conf` with a regex that extracts the host value:

```
[houseness]
```

```
DEST_KEY = MetaData:Host
REGEX = \s(\w*)$
FORMAT = host::$1
```

Next, reference your `transforms.conf` stanza in a `props.conf` stanza. For example:

```
[source::.../housesness.log]
TRANSFORMS=rhallen=housesness
SHOULD_LINEMERGE = false
```

The above stanza has the additional attribute/value pair `SHOULD_LINEMERGE = false`. This specifies that Splunk Enterprise should break events at each newline.

The events will now appear in search results like this:



The screenshot shows three search results in a table. Each row has a checkbox on the left, a timestamp, a message, and a host field. The host field is highlighted in yellow and has a dropdown arrow. The first row has host=fflanda, the second has host=rhallen, and the third has host=fflanda. The source type is housesness and the source is ./housesness.log.

	Timestamp	Message	Host	Source Type	Source
<input type="checkbox"/>	6/22/09 4:44:44.000 PM	41602052:17 accepted fflanda	fflanda	housesness	./housesness.log
<input type="checkbox"/>	6/22/09 4:44:44.000 PM	41602050:29 accepted rhallen	rhallen	housesness	./housesness.log
<input type="checkbox"/>	6/22/09 4:44:44.000 PM	41602046:53 accepted fflanda	fflanda	housesness	./housesness.log

Change host values after indexing

At some point after indexing, you might discover that the host value for some of your events is not correct. For example, you might be collecting some Web proxy logs into a directory directly on your Splunk Enterprise server and you add that directory as an input without remembering to [override the value of the host field](#), which results in the host value being the same as your Splunk Enterprise host.

If something like that happens, here are your options, from easiest to hardest:

- Delete and reindex the data.
- Use a search to delete the specific events that have the incorrect host value, and reindex those events.
- Tag the incorrect host values, and use the tag to search.
- Set up a Comma-separated values (CSV) lookup to look up the host, map it in the lookup file to a new field name, and use the new name in searches.
- Alias the host field to a new field (such as `temp_host`), set up a CSV lookup to look up the correct host name using the name `temp_host`, then have the lookup overwrite the original `host` with the new lookup value (using the `OUTPUT` option when defining the lookup).

Of these options, deleting and reindexing gives you the best performance and is the easiest. If you cannot delete and reindex the data, then the last option provides the cleanest alternative.

Configure source types

Why source types matter

The **source type** is one of the **default fields** that Splunk Enterprise assigns to all incoming data. It tells Splunk Enterprise what kind of data you have, so that it can format the data intelligently during indexing. Source types also let you categorize your data for easier searching.

Source types determine how Splunk Enterprise formats incoming data

Because Splunk Enterprise uses the source type to decide how to format your data, it is important that you assign the correct source type to your data. That way, the indexed version of the data (the **event data**) looks the way you want, with appropriate **timestamps** and **event** breaks. This facilitates easier searching of the data later.

Splunk Enterprise comes with a large number of predefined source types. When consuming data, Splunk Enterprise will usually select the correct source type automatically. Sometimes, Splunk Enterprise needs your help. If your data is specialized, you might need to manually select a different predefined source type. If your data is unusual, you might need to create a new source type with customized event processing settings. And if your data source contains heterogeneous data, you might need to assign the source type on a per-event (rather than a per-source) basis.

Like any other field, you can also use the source type field to search event data, once the data has been indexed. You will use it a lot in your searches since the source type is a key way to categorize your data.

Typical source types

Any common data input format can be a source type. Most source types are log formats. For example, some common source types that Splunk Enterprise automatically recognizes include:

- **access_combined**, for NCSA combined format HTTP Web server logs.
- **apache_error**, for standard Apache Web server error logs.

- **cisco_syslog**, for the standard syslog produced by Cisco network devices (including PIX firewalls, routers, and ACS), usually via remote syslog to a central log host.
- **websphere_core**, a core file export from WebSphere.

For a longer list of source types that Splunk Enterprise automatically recognizes, see [List of pretrained source types](#) in this manual.

Configure source types

There are two basic types of configuration you can do with source types:

- Assign source types explicitly to your incoming data.
- Create new source types, either from scratch or by modifying an existing source type.

Assign source types

In most cases, Splunk Enterprise determines the best source type for your data and automatically assigns it to incoming events. In some cases, however, you might need to explicitly assign a source type to your data. You usually do this when defining the data input. For details on how to improve source type assignment, see:

- [Override automatic source type assignment](#)
- [Override source types on a per-event basis](#)
- [Configure rule-based source type recognition](#)
- [Create source types](#)
- [Rename source types](#)

Later in this topic, there is a section that explains [how Splunk Enterprise assigns source types](#).

Create new source types

If none of the existing source types fits the needs of your data, create a new one.

Splunk Web lets you adjust source type settings to fit your data. In essence, it is a visual source type editor. See [The Set Sourcetype page](#).

You can also create a new source type by directly editing props.conf and adding a source type stanza. See [Create source types](#).

Preview data to test and modify source types

Splunk Web lets you review the effects of applying a source type to an input. It lets you preview the resulting events without actually committing them to an index. You can also edit timestamp and event breaking settings interactively and then save the modifications as a new source type. For information on how data preview functions as a source type editor, see [The Set Sourcetype page](#).

Search on source types

`sourcetype` is the name of the source type search field. You can use the `sourcetype` field to find similar types of data from any source type. For example, you could search `sourcetype=weblogic_stdout` to find all of your WebLogic server events, even when WebLogic is logging from more than one domain (or "host," in Splunk terms).

How Splunk Enterprise assigns source types

Splunk Enterprise employs a variety of methods to assign source types to event data at index time. As it processes event data, Splunk Enterprise steps through these methods in a defined order of precedence. It starts with hardcoded source type configurations in `inputs.conf` and `props.conf`, moves on to rule-based source type association, and then works through methods like automatic source type recognition and automatic source type learning. This range of methods enables you to configure how Splunk Enterprise applies source type values to specific kinds of events, while letting Splunk Enterprise assign source type values to other events automatically.

The following list shows how Splunk Enterprise goes about determining the source type for a data input. Splunk Enterprise starts with the first method and then descends through the others as necessary, until it can determine the source type. The list also provides an overview on how you configure source type assignment for each level.

Explicit source type specification based on the data input

If Splunk Enterprise finds an explicit source type for the data input, it stops here.

You configure this in `inputs.conf` or [Splunk Web](#). Here is the `inputs.conf` syntax for assigning source types to a file input:

```
[monitor://<path>]
```

```
sourcetype=<sourcetype>
```

You can also assign a source type when defining an input in Splunk Web. For information on doing this for file inputs, see [Monitor files and directories with Splunk Web](#) in this manual. The process is similar for network or other types of inputs.

For more information, see [Specify source type for an input](#).

Explicit source type specification based on the data source

If Splunk Enterprise finds an explicit source type for the particular source, it stops here.

You configure this in props.conf, using this syntax:

```
[source::<source>]
sourcetype=<sourcetype>
```

For more information, see [Specify source type for a source](#).

Rule-based source type recognition

Splunk Enterprise looks next for any rules you've created for source types.

You can create source type classification rules in props.conf:

```
[rule::<rule_name>]
sourcetype=<sourcetype>
MORE_THAN_[0-100] = <regex>
LESS_THAN_[0-100] = <regex>
```

For information about setting up source type recognition rules, see [Configure rule-based source type recognition](#).

Automatic source type matching

Splunk Enterprise next attempts to use automatic source type recognition to match similar-looking files and assign a source type.

Splunk Enterprise calculates signatures for patterns in the first few thousand lines of any file or network input stream. These signatures identify things like repeating word patterns, punctuation patterns, line length, and so on. When Splunk Enterprise calculates a signature, it compares it to its set of signatures for

known, "pretrained" source types. If it identifies a match, it assigns that source type to the data.

See [List of pretrained source types](#) in this manual for a list of the source types that Splunk Enterprise can recognize out of the box.

Delayed rule-based source type association

If Splunk Enterprise hasn't identified a source type by now, it looks for any delayed rules.

This works like rule-based associations. You create a `delayedrule::` stanza in `props.conf`. This is a useful "catch-all" for source types, in case Splunk missed any with intelligent matching (see above).

A good use of delayed rule associations is for generic versions of very specific source types that were defined earlier with `rule::` in step 3, above. For example, you could use `rule::` to catch event data with specific syslog source types, such as "sendmail syslog" or "cisco syslog" and then have `delayedrule::` apply the generic "syslog" source type to the remaining syslog event data.

Here is the syntax:

```
[delayedrule::$RULE_NAME]
sourcetype=$SOURCETYPE
MORE_THAN_[0-100] = $REGEX
LESS_THAN_[0-100] = $REGEX
```

For more information about setting up or removing delayed rules for source type recognition, see [Configure rule-based source type recognition](#).

Automatic source type learning

If Splunk Enterprise is unable to assign a source type for the event using the preceding methods, it creates a new source type for the event signature (see step 4, above). Splunk Enterprise stores learned pattern information in `sourcetypes.conf`.

Override automatic source type assignment

Splunk Enterprise attempts to assign a source type to your data automatically.

You can specify what source type to assign. You can also configure Splunk Enterprise so that it assigns a source type based on either the data input or the data source.

For details on the precedence rules that Splunk Enterprise uses to assign source types to data, read [How Splunk assigns source types](#).

Overrides only work on file and directory monitoring inputs or files you have uploaded. You cannot override the source type on network inputs. Additionally, overrides only affect new data that arrives after you set up the override. To correct the source types of events that have already been indexed, create a tag for the source type instead.

This topic describes how to specify a source type based for data based on its:

- [input](#)
- [source](#)

Specify source type for an input

You can assign the source type for data coming from a specific input, such as `/var/log/`. You do this in either Splunk Web or the `inputs.conf` configuration file.

Note: While assigning source type by input seems like a simple way to handle things, it is not very granular--when you use it, Splunk Enterprise assigns the same source type to **all** data from an input, even if some of the data comes from different sources or hosts. To bypass automatic source type assignment in a more targeted manner, you can arrange for Splunk to assign source types based on the source of the data, as described [later](#) in this topic.

Use Splunk Web

When you [define a data input](#), you can set a source type value that Splunk Enterprise applies to all incoming data from that input. Splunk Enterprise gives you the option of picking a source type from a list or entering a unique source type value of your own.

To select a source type for an input, change the source type settings for the data input type you want to add. For example, for file inputs:

1. Click **Settings** in the upper right-hand corner of Splunk Web.

2. In the Data section of the Settings pop-up, click **Data Inputs**.
3. Click **Files & Directories**.
4. Click the **New** button to add an input.
5. In the "Add Data" page, browse or enter the name of the file you want to monitor, then click "Next".
6. In the "Set Sourcetype" page, click the "Sourcetype" drop-down and choose from the list of [pretrained source types](#). Splunk Enterprise updates the page to show how the data looks when it receives the new source type.
7. If you want to make changes to the source type, use the "Event Breaks", "Timestamp", and "Advanced" tabs to modify settings and refresh the data preview. See [The Set Sourcetype page](#) in this manual.
8. If you want to save the source type as a different name, click **Save As?** to open a dialog box to save the new source type. Otherwise, proceed to Step 10.
9. If you chose to save the source type, Splunk Enterprise displays the "Save Sourcetype" dialog. Enter the name, description, category, and app that the source type should apply to. See [Save modifications as a new source type](#).
10. Click "Next" to set the source type for the data and proceed to the [Input settings](#) page.

Splunk Enterprise now assigns your selected source type to all events it indexes for that input.

Use the inputs.conf configuration file

When you configure an input in `inputs.conf`, you can specify a source type for the input. Edit `inputs.conf` in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files in general, see [About configuration files](#) in the Admin manual.

To specify a source type, include a `sourcetype` attribute within the stanza for the input. For example:

```
[tcp://:9995]
connection_host=dns
sourcetype=log4j
```

```
source=tcp:9995
```

This example sets the source type to "log4j" for any events coming from your TCP input on port 9995.

Caution: Do not put quotes around the attribute value: `sourcetype=log4j`, not `sourcetype="log4j"`.

Specify source type for a source

Use `props.conf` to override automated source type matching and explicitly assign a single source type to all data coming from a specific source.

Edit `props.conf` in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files in general, see [About configuration files](#).

Note: If you forward data, and you want to assign a source type for a source, you must assign the source type in `props.conf` on the **forwarder**. If you do it in `props.conf` on the **receiver**, the override has no effect.

To override source type assignment, add a stanza for your source to `props.conf`. In the stanza, identify the source path, using regular expression (regex) syntax for flexibility if necessary. Then specify the source type by including a `sourcetype` attribute. For example:

```
[source:../../../../var/log/anaconda.log(.\d+)?]  
sourcetype=anaconda
```

This example sets the source type to "anaconda" for events from any sources containing the string `/var/log/anaconda.log` followed by any number of numeric characters.

Your stanza source path regexes (such as `[source:../../../../web/....log]`) should be as specific as possible. Avoid using a regex that ends in "...". For example, do not do this:

```
[source:~/home/fflanda/...]  
sourcetype=mytype
```

This is dangerous. It tells Splunk to process any gzip files in `/home/fflanda` as "mytype" files rather than gzip files.

Instead, write:

```
[source::/home/fflanda/....log(.\d+)?]  
sourcetype=mytype
```

Configure rule-based source type recognition

You can use rule-based source type recognition to expand the range of source types that Splunk Enterprise recognizes. In `props.conf`, you create a `rule::` stanza that associates a specific source type with a set of qualifying criteria. When consuming data, Splunk Enterprise assigns the specified source type to file inputs that meet the rule's qualifications.

You can create two kinds of rules in `props.conf`: rules and delayed rules. The only difference between the two is the point at which Splunk Enterprise checks them during the source typing process. As it processes each set of incoming data, [Splunk Enterprise uses several methods to determine source types](#):

- After [checking for explicit source type definitions based on the data input or source](#), Splunk Enterprise looks at any `rule::` stanzas defined in `props.conf` and tries to match source types to the data based on the classification rules specified in those stanzas.
- If Splunk Enterprise is unable to find a matching source type using the available `rule::` stanzas, it tries to use automatic source type matching, where it tries to identify patterns similar to source types it has learned in the past.
- If that method fails, Splunk Enterprise then checks any `delayedrule::` stanzas in `props.conf` and tries to match the data to source types using the rules in those stanzas.

For details on the precedence rules that Splunk Enterprise uses to assign source types to data, read [How Splunk Enterprise assigns source types](#).

For a primer on regular expression syntax and usage, see [Regular-Expressions.info](#). You can test regexes by using them in searches with the `rex` search command.

You can configure your system so that `rule::` stanzas contain classification rules for specialized source types, while `delayedrule::` stanzas contain classification rules for generic source types. That way, Splunk Enterprise applies the generic source types to broad ranges of events that haven't qualified for more specialized source types. For example, you could use `rule::` stanzas to catch data with specific syslog source types, such as `sendmail_syslog` or `cisco_syslog`, and then configure a `delayedrule::` stanza to apply the generic `syslog` source type

to any remaining syslog data.

Configuration

To set source typing rules, edit `props.conf` in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files in general, see [About configuration files in the Admin manual](#).

Create a rule by adding a `rule::` or `delayedrule::` stanza to `props.conf`. Provide a name for the rule in the stanza header, and declare the source type in the body of the stanza. After the source type declaration, list the source type assignment rules. These rules use one or more `MORE_THAN` and `LESS_THAN` statements to find patterns in the data that fit given regular expressions by specific percentages.

To create a rule, use this syntax:

```
[rule::<rule_name>] OR [delayedrule::<rule_name>]
sourcetype=<source_type>
MORE_THAN_[0-99] = <regex>
LESS_THAN_[1-100] = <regex>
```

You set a numerical value in the `MORE_THAN` and `LESS_THAN` attributes, corresponding to the percentage of input lines that must contain the string specified by the regular expression. For example, `MORE_THAN_80` means at least 80% of the lines must contain the associated expression. `LESS_THAN_20` means that less than 20% of the lines can contain the associated expression.

Note: Despite its nomenclature, the `MORE_THAN_` attribute actually means "more than or equal to". Similarly the `LESS_THAN_` attribute means "less than or equal to".

A rule can contain any number of `MORE_THAN` and/or `LESS_THAN` conditions. Splunk assigns the rule's source type to a data file only if the data qualifies all the statements in the rule. For example, you could define a rule that assigns a specific source type to a file input only if more than 60% of the lines match one regular expression and less than 20% match another regular expression.

Examples

Postfix syslog files

```
# postfix_syslog sourcetype rule
[rule::postfix_syslog]
sourcetype = postfix_syslog
# If 80% of lines match this regex, then it must be this type
MORE_THAN_80=^\w{3} +\d+ \d\d:\d\d:\d\d .* postfix(/\w+)?\[\d+\]:
```

Delayed rule for breakable text

```
# breaks text on ascii art and blank lines if more than 10% of lines
have
# ascii art or blank lines, and less than 10% have timestamps
[delayedrule::breakable_text]
sourcetype = breakable_text
MORE_THAN_10 = (^(?:---|==|\*\*\*|___|+=))|^s*$
LESS_THAN_10 = [: ][012]?[0-9]:[0-5][0-9]
```

List of pretrained source types

Splunk Enterprise ships with definitions for a large number of source types. These built-in source types are known as "pretrained" source types.

Splunk Enterprise can automatically recognize and assign many of these pretrained source types to incoming data. Splunk Enterprise also includes some pretrained source types that it does not automatically recognize but that you can manually assign via Splunk Web or `inputs.conf`, using methods described in earlier topics in this chapter, such as [Override automatic source type assignment](#).

It is a good idea to use a pretrained source type if it matches your data, as Splunk Enterprise already knows how to properly index pretrained source types. However, if your data does not fit any pretrained source types, you can create your own source types, as described in [Create source types](#). Splunk Enterprise can also index virtually any format of data even without custom properties.

For an introduction to source types, see [Why source types matter](#).

Automatically recognized source types

Source type name	Origin	Example
access_combined	NCSA combined format http web server logs (can be	10.1.1.43 - webdev [08/Aug/2005:10:04:42] "-" "check_http/1.10 (nagios

	generated by apache or other web servers)	
access_combined_wcookie	NCSA combined format http web server logs (can be generated by apache or other web servers), with cookie field added at end	"66.249.66.102.1124471045570513"-0700] "GET /themes/splunk_com/im "http://www.splunk.org/index.php/ en-US; rv:1.7.8) Gecko/20050524 F "61.3.110.148.1124404439914689"
access_common	NCSA common format http web server logs (can be generated by apache or other web servers)	10.1.1.140 - - [16/May/2005:15:01 /themes/ComBeta/images/bullet.png
apache_error	Standard Apache web server error log	[Sun Aug 7 12:17:35 2005] [error] exist: /home/reba/public_html/ima
asterisk_cdr	Standard Asterisk IP PBX call detail record	"", "5106435249", "1234", "default", Jesse""<5106435249>", "SIP/5249-1c 15:19:25", "2005-05-26 15:19:25", " 15:19:42", 17, 17, "ANSWERED", "DOCUM
asterisk_event	Standard Asterisk event log (management events)	Aug 24 14:08:05 asterisk[14287]: I 127.0.0.1
asterisk_messages	Standard Asterisk messages log (errors and warnings)	Aug 24 14:48:27 WARNING[14287]: C extension 's' in context 'default
asterisk_queue	Standard Asterisk queue log	1124909007 NONE NONE NONE CONFIGR
cisco_syslog	Standard Cisco syslog produced by all Cisco network devices including PIX firewalls, routers, ACS, etc., usually via remote syslog to a central log host	Sep 14 10:51:11 stage-test.splunk 00:08:49: %PIX-2-106001: Inbound to IP_addr/port flags TCP_flags o connection denied from 144.1.10.2 on interface outside
db2_diag	Standard IBM DB2 database administrative and error log	2005-07-01-14.08.15.304000-420 I2 : 4760 PROC : db2fmp.exe INSTANCI Automatic Table Maintenance, db2H Automatic Runstats: evaluation ha
exim_main	Exim MTA mainlog	2005-08-19 09:02:43 1E69KN-0001u6 R=send_to_relay T=remote_smtp H=m
exim_reject	Exim reject log	2005-08-08 12:24:57 SMTP protocol (input sent without waiting for g H=gate.int.splunk.com [10.2.1.254
linux_messages_syslog		

	Standard linux syslog (/var/log/messages on most platforms)	Aug 19 10:04:28 db1 sshd(pam_unix by (uid=0)
linux_secure	Linux securelog	Aug 18 16:19:27 db1 sshd[29330]: from ::ffff:10.2.1.5 port 40892 s
log4j	Log4j standard output produced by any J2EE server using log4j	2005-03-07 16:44:03,110 53223013 property...
mysqld_error	Standard mysql error log	050818 16:19:29 InnoDB: Started; /usr/libexec/mysqld: ready for con socket: '/var/lib/mysql/mysql.sock
mysqld	Standard MySQL query log; also matches the MySQL binary log following conversion to text	53 Query SELECT xar_dd_itemid, xar_dynamic_data WHERE xar_dd_pro
postfix_syslog	Standard Postfix MTA log reported via the Unix/Linux syslog facility	Mar 1 00:01:43 avas postfix/smtpd client=host76-117.pool80180.inter
sendmail_syslog	Standard Sendmail MTA log reported via the Unix/Linux syslog facility	Aug 6 04:03:32 nmrl100 sendmail[5: ctladdr=root (0/0), delay=00:00:0 min=00026, relay=[101.0.0.1] [101 (v00F3HmX004301 Message accepted
sugarcrm_log4php	Standard Sugarcrm activity log reported using the log4php utility	Fri Aug 5 12:39:55 2005,244 [2866 the application list language fil the default language(en_us)
weblogic_stdout	Weblogic server log in the standard native BEA format	####<Sep 26, 2005 7:27:24 PM MDT> <asiAdminServer> <ListenThread.De <HostName: 0.0.0.0, maps to multi addresses:169.254.25.129,169.254.
websphere_activity	Websphere activity log, also often referred to as the service log	----- ProcessId: 2580 ThreadId: 0000001 SourceId: com.ibm.ws.channel.frame ClassName: MethodName: Manufactur Platform 6.0 [BASE 6.0.1.0 o0510. nd6Cell101\was1Node01\TradeServer1 13:04:55.187000000 UnitOfWork: Se PrimaryMessage: CHFW0020I: The Tr Chain labeled SOAPAcceptorChain2 -----
websphere_core	Corefile export from Websphere	NULL----- subcomponent dump routine NULL==== signal 0 received 1TIDATETIME Data Javacore filename: /kmbcc/javacore

		----- subcomponent dump routine NULL == Tue Aug 2 10:19:24 20051XHSIGRECV Processing terminated. 1XHFULLVER ca131-20031105 NULL
websphere_trlog_syserr	Standard Websphere system error log in the IBM native trlog format	[7/1/05 13:41:00:516 PDT] 000003a com.ibm.ws.http.channel. inbound. (HttpICLReadCallback.java (Compiled
websphere_trlog_sysout	Standard Websphere system out log in the IBM native trlog format; similar to the log4j server log for Resin and Jboss, sample format as the system error log but containing lower severity and informational events	[7/1/05 13:44:28:172 PDT] 0000082 2005 TradeStreamerMDB: 100 Trade Statistics Total update Quote Pri stock update alerts messages (in 1.0365270454545454 The current pr s:393 old price = 15.47 new price
windows_snare_syslog	Standard windows event log reported through a 3rd party Intersect Alliance Snare agent to remote syslog on a Unix or Linuxserver	0050818050818 Sep 14 10:49:46 sta MSWinEventLog 0 Security 3030 Day admin4 User Success Audit Test_Ho Object Type: File Object Name: C: 1220 Operation ID: {0,117792} Pro Primary Domain: FLAME Primary Log Client Domain: - Client Logon ID: ListDirectory) Privileges -Sep

Special source types

Source type name	Origin	Examples
known_binary	The filename matches a pattern that is generally known to be a binary file, not a log file	mp3 files, images, .rdf, .dat, etc. This is intended to catch obvious non-text files

Pretrained source types

These are all the pretrained source types, including both those that are automatically recognized and those that are not.

Category	Source type(s)
Application servers	log4j, log4php, weblogic_stdout, websphere_activity, websphere_core, websphere_trlog, catalina, ruby_on_rails
Databases	db2_diag, mysqld, mysqld_error, mysqld_bin, mysqld_slow

E-mail	exim_main, exim_reject, postfix_syslog, sendmail_syslog, procmail
Operating systems	linux_messages_syslog, linux_secure, linux_audit, linux_bootlog, anaconda, anaconda_syslog, osx_asl, osx_crashreporter, osx_crash_log, osx_install, osx_secure, osx_daily, osx_weekly, osx_monthly, osx_window_server, windows_snare_syslog, dmesg, ftp, ssl_error, syslog, sar, rpmpkgs
Network	novell_groupwise, tcp
Printers	cups_access, cups_error, spooler
Routers and firewalls	cisco_cdr, cisco:asa, cisco_syslog, clavister
VoIP	asterisk_cdr, asterisk_event, asterisk_messages, asterisk_queue
Webservers	access_combined, access_combined_wcookie, access_common, apache_error, iis?
Splunk	splunk_com_php_error, splunkd, splunkd_crash_log, splunkd_misc, splunkd_stderr, splunk-blocksignature, splunk_directory_monitor, splunk_directory_monitor_misc, splunk_search_history, splunkd_remote_searches, splunkd_access, splunkd_ui_access, splunk_web_access, splunk_web_service, splunkd_conf?, django_access, django_service, django_error, splunk_help, mongod
Non-Log files	csv?, psv?, tsv?, _json?, json_no_timestamp, fs_notification, exchange?, generic_single_line
Miscellaneous / Other	snort, splunk_disk_objects?, splunk_resource_usage?, kvstore?

? These source types use the `INDEXED_EXTRactions` attribute, which sets other attributes in `props.conf` to specific defaults, and requires special handling to forward to another Splunk Enterprise instance. See [Forward data extracted from structured data files](#).

Finding out how a pretrained source type is configured to work

To find out what configuration information Splunk Enterprise uses to index a given source type, you can invoke the `btool` utility to list out the properties. For more information on using `btool`, refer to [Use btool to troubleshoot configurations](#)

in the Troubleshooting manual.

The following example shows how to list out the configuration for the `tcp` source type:

```
$ ./splunk btool props list tcp
[tcp]
BREAK_ONLY_BEFORE = (=\+)+
BREAK_ONLY_BEFORE_DATE = True
CHARSET = UTF-8
DATETIME_CONFIG = /etc/datetime.xml
KV_MODE = none
LEARN_SOURCETYPE = true
MAX_DAYS_AGO = 2000
MAX_DAYS_HENCE = 2
MAX_DIFF_SECS_AGO = 3600
MAX_DIFF_SECS_HENCE = 604800
MAX_EVENTS = 256
MAX_TIMESTAMP_LOOKAHEAD = 128
MUST_BREAK_AFTER =
MUST_NOT_BREAK_AFTER =
MUST_NOT_BREAK_BEFORE =
REPORT-tcp = tcpdump-endpoints, colon-kv
SEGMENTATION = inner
SEGMENTATION-all = full
SEGMENTATION-inner = inner
SEGMENTATION-outer = foo
SEGMENTATION-raw = none
SEGMENTATION-standard = standard
SHOULD_LINEMERGE = True
TRANSFORMS =
TRANSFORMS-baindex = banner-index
TRANSFORMS-dlindex = download-index
TRUNCATE = 10000
maxDist = 100
pulldown_type = true
```

Override source types on a per-event basis

This topic shows you how to configure Splunk Enterprise to override source types on a per-event basis. You do this at parse-time, after Splunk Enterprise has made its initial assignment as described in [How Splunk Enterprise assigns source types](#).

To configure per-event overrides, you use `transforms.conf` in tandem with `props.conf`.

Since this type of override occurs at parse-time, it works only on an indexer or heavy forwarder, not on a universal forwarder. See Configuration parameters and the data pipeline in the Admin manual for more information on what configurations are available at different points in the input/parsing/indexing process.

For information about configuring basic (not per-event) source type overrides for event data that comes from specific inputs or that has a particular source, see [Override automatic source type assignment](#) in this manual.

Configuration

To configure per-event overrides, you need to create two stanzas, one in `transforms.conf` and another in `props.conf`. Edit these files in `$SPLUNK_HOME/etc/system/local/` or in your own custom application directory in `$SPLUNK_HOME/etc/apps/`. For more information about configuration files in general, see About configuration files in the Admin manual.

transforms.conf

Create a stanza in `transforms.conf` that follows this syntax:

```
[<unique_stanza_name>]
REGEX = <your_regex>
FORMAT = sourcetype::<your_custom_sourcetype_value>
DEST_KEY = MetaData:Sourcetype
```

Note the following:

- `<unique_stanza_name>` should reflect that it involves a source type. You'll use this name later in the `props.conf` stanza.
- `<your_regex>` is a regular expression that identifies the events that you want to apply a custom source type to (such as events carrying a particular hostname or other field value).
- `<your_custom_sourcetype_value>` is the source type that you want to apply to the regex-selected events.

Note: For a primer on regular expression syntax and usage, see [Regular-Expressions.info](#). You can test regexes by using them in searches with the `rex` search command. Splunk also maintains a list of useful third-party tools for writing and testing regular expressions.

props.conf

Next, create a stanza in `props.conf` that references the `transforms.conf` stanza:

```
[<spec>]
TRANSFORMS-<class> = <unique_stanza_name>
```

Note the following:

- `<spec>` can be:
 - ♦ `<sourcetype>`, the source type of an event.
 - ♦ `host::<host>`, where `<host>` is the host value for an event.
 - ♦ `source::<source>`, where `<source>` is the source value for an event.
- `<class>` is any unique identifier that you want to give to your transform.
- `<unique_stanza_name>` is the name of the stanza you created in `transforms.conf`.

Example: Assign a source type to events from a single input but different hosts

Let's say that you have a shared UDP input, "UDP514". Your Splunk Enterprise instance indexes a wide range of data from a number of hosts through this input. You've found that you need to apply a particular source type called "my_log" to data originating from three specific hosts (host1, host2, and host3) reaching Splunk through UDP514.

To start, you can use the regex that Splunk typically uses to extract the host field for syslog events. You can find it in `system/default/transforms.conf`:

```
[syslog-host]
REGEX
= : \d\d\s+(?:\d+\s+|(?:(?:user|daemon|local.?)\.\w+\s+))*\[?(?(\w[\w\.-]{2,})\])?\s
FORMAT = host::$1
DEST_KEY = MetaData:Host
```

You can easily modify this regex to only match events from the hostnames you want (in this example, host1, host2, and host3):

```
REGEX
= : \d\d\s+(?:\d+\s+|(?:(?:user|daemon|local.?)\.\w+\s+))*\[?(host1|host2|host3)[\w\.-]*\]?
```

Now you can use the modified regex in a transform that applies the `my_log` source type to events that come from those three hosts:


```
[set_sourcetype_my_log_for_some_hosts]
REGEX
= : \d\d\d\s+(?:\d+\s+|(?:(?:user|daemon|local.?)\.\w+\s+))*\[?(?:host1|host2|host3) [\w\.-]*\]?
FORMAT = sourcetype::my_log
DEST_KEY = MetaData:Sourcetype
```

Then you can specify that transform in a `props.conf` stanza that identifies the specific input for the events:

```
[source::udp:514]
TRANSFORMS-changesourcetype = set_sourcetype_my_log_for_some_hosts
```

Create source types

You can create new source types in several ways:

- Use the "Set Sourcetype" page in Splunk Web as part of adding the data.
- Create a source type in the "Source types" management page, as described in [Add source type](#).
- Edit the `props.conf` configuration file directly.

Set the source type in Splunk Web

The "Set Sourcetype" page in Splunk Web provides an easy way to view the effects of applying a source type to your data and to make adjustments to the source type settings as necessary. You can save your changes as a new source type, which you can then assign to data inputs.

The page lets you make the most common types of adjustments to **timestamps** and **event** breaks. For other modifications, it lets you edit the underlying `props.conf` file directly. As you change settings, you can immediately see the changes to the event data.

The page appears only when you specify or upload a single file. It does not appear when you specify any other type of source.

To learn more about the page, see [The "Set Sourcetype" page](#) in this manual.

Create a source type

You can use the "Source types" management page to create a new source type. See [Add source type](#) in this manual.

Edit props.conf

You can also create a new source type by editing `props.conf` and adding a new stanza. For detailed information on `props.conf`, read the `props.conf` specification in the Admin manual. For information on configuration files in general, see About configuration files in the Admin manual.

The following is an example of an entry in `props.conf`. This entry defines the `access_combined` source type and then assigns that source type to files that match the specified source. You can specify multiple files or directories in a source by using a regular expression.

```
[access_combined]
pulldown_type = true
maxDist = 28
MAX_TIMESTAMP_LOOKAHEAD = 128
REPORT-access = access-extractions
SHOULD_LINEMERGE = False
TIME_PREFIX = \[
category = Web
description = National Center for Supercomputing Applications (NCSA)
combined fo
rmat HTTP web server logs (can be generated by apache or other web
servers)
```

```
[source::/opt/weblogs/apache.log]
sourcetype = iis
```

To edit `props.conf`:

1. On the host where you want to create a source type, make a copy of

`$SPLUNK_HOME/etc/system/default/props.conf` and save it in

`$SPLUNK_HOME/etc/system/local`.

Note: You might need to create the `local` directory. If you use an app, go to the app directory in `$SPLUNK_HOME/etc/apps`.

2. Using a text editor, open the `props.conf` file in

`$SPLUNK_HOME/etc/system/local`.

3. Add a stanza for the new source type and specify any attributes that Splunk Enterprise should use when handling the source type.

```
[my_sourcetype]
attributel = value
```

```
attribute2 = value
```

Note: See the `props.conf` specification for a list of attributes and how they should be used.

4. Optionally, if you know the name of the file (or files) that Splunk Enterprise should apply the source type to, you can specify them with the

`[source::<source>]` stanza:

```
[my_sourcetype]
attribute1 = value
attribute2 = value
```

```
[source::.../my/logfile.log]
sourcetype = my_sourcetype
```

5. Save the `props.conf` file.

6. Restart Splunk Enterprise. The new source types take effect after the restart completes.

Specify event breaks and time stamping

When you create a source type, there are some key attributes that you should specify:

- **Event breaks.** To learn how to use `props.conf` to specify event breaks, see [Configure event linebreaking](#).
- **Timestamps.** To learn how to use `props.conf` to specify timestamps, see [Configure timestamp recognition](#), as well as other topics in the "Configure timestamps" chapter of this manual.

There are also a number of additional settings that you can configure. See the `props.conf` specification for more information.

Manage source types

This topic discusses the source types management page. This page lets you create, edit, and delete source types in Splunk Enterprise. It loads when you select "Source types" from the Settings menu.

Name ^	Actions	Category	App
json JavaScript Object Notation format. For more information, visit http://json.org/	Edit	Structured	system
access_combined National Center for Supercomputing Applications (NCSA) combined format HTTP web server logs (can be generated by apache or other web servers)	Edit	Web	ao-bog
apache_error Error log format produced by the Apache web server (typically error_log on *nix systems)	Edit	Web	system
catalina Output produced by Apache Tomcat Catalina (System.out and System.err)	Edit	Application	system
ciscoasa Output produced by the Cisco Adaptive Security Appliance (ASA) Firewall	Edit	Network & Security	system
csv Comma-separated value format. Set header and other settings in "Delimited Settings"	Edit	Structured	system

The Source Types page displays all source types that have been configured on the instance. It shows the default source types that come with every installation of Splunk Enterprise as well as any source types that you have added.

Sort source types

By default, the Source Types management page sorts source types alphabetically. You can change how the page sorts by clicking the header bar for the "Name", "Category", and "App" columns.

Each header bar (except for "Actions") acts as a toggle. Click once to sort in ascending order and click again to sort in descending order.

Filter source types

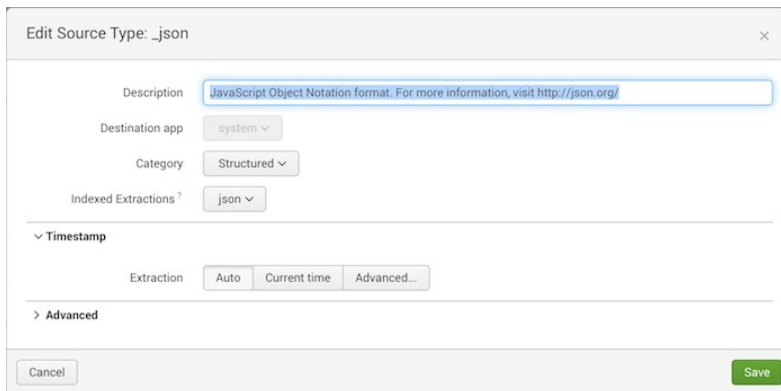
You can filter the number of source types you see on the Source Type management page.

- To see only source types that belong to a certain category, click the "Category" drop-down and select the category you want. Only source types that belong to that category will display. To see all source types again, select "All" from the "Category" drop-down.
- To see only source types that belong in a certain application context, click the "App" drop-down and select the application context that the source type applies to. Only source types that apply to that application context will display. To see all source types again, select "All" from the "App" drop-down.
- To see only source types whose names contain a certain string, type that string in the "Filter" text box next to the "App" drop-down, then press Enter. Only source types whose names or descriptions match what you have typed in the "Filter" box display. To see all source types again, click the "x" button on the right side of the "Filter" text box.

By default, the Source Types management page shows up to 20 source types on a page. If you want to see more or less, click the "20 per page" link on the right side of the page and select the number of source types you want to see. Choices are 10, 20, 50, or 100 listings on a page.

Modify source types

To modify a source type, click its name in the list, or click its "Edit" link in the "Actions" column. The "Edit Source type" page appears.



The "Edit Source Type" dialog lets you change the configuration of a source type. You can change the following:

Description: Type in the description of the source type in the "Description" field.

Destination app: The application context that the source type applies to.

Note: You cannot change the app destination for source types included with Splunk Enterprise.

Category: The category that the source type is a member of. Click the button to select from the list of categories and choose the one you want. When you save, the source type appears in the category you selected.

Indexed Extractions: A format for extracting fields at index time from files with structured data. Select the type of indexed extraction that best represents the contents of the file:

- none: The file does not contain structured data.
- json: The file is in JavaScript Object Notation (json) format.
- csv: The file has comma-separated values.
- tsv: The file has tab-separated values.

- psv: The file has pipe (|) separated values.
- w3c: The file conforms to the World Wide Web Consortium (W3C) logging format.

Timestamp:

The Timestamp section of the dialog controls how Splunk Enterprise extracts timestamps for events from the source file.

- Auto: Splunk Enterprise extracts time stamps from the source file using automatic methods.
- Current Time: Splunk Enterprise uses the current time for each event it extracts from the source file.
- Advanced: Splunk Enterprise uses advanced methods to extract time stamps from the source file.

Advanced time stamp extraction configurations

The following advanced configurations are available when you select "Advanced" in the "Timestamp Extraction" section:

- Time zone: Select the time zone that you want Splunk Enterprise to use when it extracts the timestamps.
- Timestamp format: Enter a string that represents the time stamp format that Splunk Enterprise should expect when it reads the source file. The available formats come from the properties of the `strptime()` programming function. For example, if the source file contains logs with timestamps in this format:

```
6 Jun 2015 18:35:05
```

then, to tell Splunk Enterprise to extract timestamps like this, you would specify the following in the "Timestamp format" field:

```
%d %b %Y %H:%M:%S
```

Another example:

```
Tue Jun 4 2:55:18 PM 2015
```

maps to

```
%a %b %d %I:%M:%S %p %Y
```

For a list of the strings that you can use to define the time stamp format, see `strptime(3)` (<http://linux.die.net/man/3/strptime>) on the die.net Linux man page site.

- **Timestamp prefix:** A regular expression that represents the characters that come before a time stamp. When Splunk Enterprise sees this set of characters in an event, it expects a time stamp to occur after that.
- **Lookahead:** This option tells Splunk Enterprise to scan no more than the number of characters specified into an event for the time stamp. If you specified a regular expression in the "Timestamp prefix" field, it looks no more than the number of characters specified past the string that the regular expression represents for the time stamp.

Advanced

The Advanced section of the dialog shows you all of the configurations for the source type, in key/value format. This represents what is in the `props.conf` file that defines the source type. You can edit each setting directly, or add and delete settings. To delete settings, click the "x" on the right side of each setting. To add an entry, click the "New setting" link at the bottom of the dialog. This exposes a key/value pair of fields. Enter the key name in the "Name" field and its value in the "Value" field.

Caution: Use the "Advanced" section with care. Adding or changing values here can cause Splunk Enterprise to index data incorrectly.

Add Source Type

To create a new source type, click the "New Source Type" button at the top right of the screen. The "Create Source Type" dialog opens.

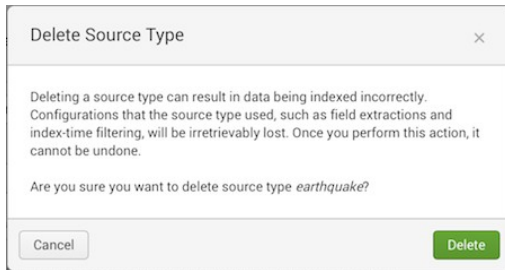
This dialog is exactly the same as the "Edit Source Type" dialog. See "[Managesourcetypes](#)" for information on the controls in the dialog.

When you have finished configuring the source type, click "Save." Splunk Enterprise saves the source type and the dialog box closes.

Delete Source Type

To delete a source type, click the "Delete" link in the "Actions" column for the source type that you want to delete. You cannot delete source types that come with Splunk Enterprise, only source types that you create or that come with apps.

When you delete a source type, the following dialog appears:



Caution: Deleting a source type has significant consequences, especially if Splunk Enterprise uses the source type currently:

- Data can be indexed incorrectly once you delete the source type. Making the data searchable in the way you want later can take a lot of effort. Many apps and add ons use source types to look for data, and data indexed under a missing source type is data those apps and add-ons do not see.
- Any configurations that the source type uses, such as field extractions, index time filtering, and time stamp formats, are irretrievably lost.
- You can not undo a source type deletion. The only options available in this case are to restore the `props.conf` file that defines the source type from a backup, or recreate the source type manually.

If you are sure you want to delete the source type, click "Delete". The dialog closes and Splunk Enterprise returns you to the Source Types management page.

Rename source types at search time

You might want to rename a source type in certain situations. For example, say you accidentally assigned an input to the wrong source type. Or you realize that two differently named source types should be handled exactly the same at search time.

You can use the `rename` attribute in `props.conf` to assign events to a new source type at search time. In case you ever need to search on it, Splunk Enterprise moves the original source type to a separate field, `_sourcetype`.

Note: The indexed events still contain the original source type name. The renaming occurs only at search time. Also, renaming the source type does only that; it does not fix any problems with the indexed format of your event data caused by assigning the wrong source type in the first place.

To rename the source type, add the `rename` attribute to your source type stanza:

```
rename = <string>
```

Note: A source type name can only contain the letters `a` through `z`, the numerals `0` through `9`, and the `_` (underscore) character.

For example, say you're using the source type `"cheese_shop"` for your application server. Then, accidentally, you index a pile of data as source type `"whoops"`. You can rename `"whoops"` to `"cheese_shop"` with this `props.conf` stanza:

```
[whoops]
rename=cheese_shop
```

Now, a search on `"cheese_shop"` will bring up all the `"whoops"` events as well as any events that had a `"cheese_shop"` source type from the start:

```
sourcetype=cheese_shop
```

If you ever need to single out the `"whoops"` events, you can use `_sourcetype` in your search:

```
_sourcetype=whoops
```

Important: Data from a renamed source type will only use the search-time configuration for the target source type (`"cheese_shop"` in this example). Any field extractions for the original source type (`"whoops"` in the example) will be ignored.

Manage event segmentation

About event segmentation

Segmentation is what Splunk Enterprise uses to break events up into searchable **segments** at **index** time, and again at **search** time. Segments can be classified as **major** or **minor**. Minor segments are breaks within major segments. For example, the IP address `192.0.2.223` is a major segment. But this major segment can be broken down into minor segments, such as `"192"`, as well as groups of minor segments like `"192.0.2"`.

You can define how detailed the event segmentation should be. This is important because **index-time segmentation** affects indexing and search speed, storage size, and the ability to use typeahead functionality (where Splunk Web provides items that match text you type into the Search bar). **Search-time segmentation**, on the other hand, affects search speed and the ability to create searches by selecting items from the results displayed in Splunk Web.

For more information about the distinction between "index time" and "search time," see "Index time versus search time" in the Managing Indexers and Clusters manual.

You can assign segmentation to specific categories of events in `props.conf`, as described in ["Set the segmentation for event data"](#).

Types of event segmentation

There are three main types, or levels, of segmentation, configurable at index or search time:

- **Inner segmentation** breaks events down into the smallest minor segments possible. For example, when an IP address such as `192.0.2.223` goes through inner segmentation, it is broken down into `192`, `0`, `2`, and `223`. Setting inner segmentation at index time leads to faster indexing and searching and reduced disk usage. However, it restricts the typeahead functionality, so that a user can only type ahead at the minor segment level.
- **Outer segmentation** is the opposite of inner segmentation. Under outer segmentation, Splunk Enterprise only indexes major segments. For example, the IP address `192.0.2.223` gets indexed as `192.0.2.223`, which means that you cannot search on individual pieces of the phrase. You can

still use wildcards, however, to search for pieces of a phrase. For example, you can search for `192.0*` and you will get any events that have IP addresses that start with `192.0`. Also, outer segmentation disables the ability to click on different segments of search results, such as the `192.0` segment of the same IP address. Outer segmentation tends to be marginally more efficient than full segmentation, while inner segmentation tends to be much more efficient.

- **Full segmentation** is a combination of inner and outer segmentation. Under full segmentation, the IP address is indexed both as a major segment and as a variety of minor segments, including minor segment combinations like `192.0` and `192.0.2`. This is the least efficient indexing option, but it provides the most versatility in terms of searching.

The `segmenters.conf` file, located in `$SPLUNK_HOME/etc/system/default`, defines all available segmentation types. By default, index-time segmentation is set to the `indexing` type, which is a combination of inner and outer segmentation. Search-time segmentation is set to full segmentation.

No segmentation

The most space-efficient segmentation setting is to disable segmentation completely. This has significant implications for search, however. By setting Splunk Enterprise to index with no segmentation, you restrict searches to indexed fields, such as time, source, host, and source type. Searches for keywords will return no results. You must pipe your searches through the search command to further restrict results. Use this setting only if you do not need any advanced search capability.

Configure segmentation types

`segmenters.conf` defines segmentation types. You can define custom segmentation types, if necessary.

For information on the types of segmentation available by default, look at the `segmenters.conf` file in `$SPLUNK_HOME/etc/system/default`.

Important: Do not modify the default file. If you want to make changes to the existing segmentation stanzas or create new ones altogether, you can copy the default file to `$SPLUNK_HOME/etc/system/local/` or to a custom app directory in `$SPLUNK_HOME/etc/apps/`. For information on configuration files and directory locations, see "About configuration files".

Set segmentation types for specific hosts, sources, or source types

You can configure index-time and search-time segmentation to apply to specific hosts, sources, or source types. If you run searches that involve a particular source type on a regular basis, you could use this capability to improve the performance of those searches. Similarly, if you typically index a large number of `syslog` events, you could use this feature to help decrease the overall disk space that those events take up.

For details about how to apply segmentation types to specific event categories, see ["Set the segmentation for event data"](#).

Set the segmentation for event data

By default, Splunk Enterprise segments events during indexing to allow for the most flexible searching. There are numerous types of segmentation available, and you can create others if necessary. The type of segmentation that you employ affects indexing speed, search speed, and the amount of disk space the indexes occupy. To learn more about segmentation and the trade-offs between the various types of segmentation, refer to ["About segmentation"](#).

Splunk Enterprise can also segment events at search time. You can set search-time segmentation in Splunk Web, as described in ["Set search-time segmentation in Splunk Web"](#).

If you know how you want to search for or process events from a specific host, source, or source type, you can configure index-time segmentation for that specific type of event. You can also configure search-time segmentation options for specific types of events.

Specify segmentation in props.conf

Specify segmentation for events of particular hosts, sources, or source types by assigning segmentation types to the appropriate stanzas in `props.conf`. In the stanzas, you assign segmentation types, or "rules", that have been defined in `segmenters.conf`. These can either be predefined types (such as `inner`, `outer`, or `full`), or custom types that you've defined. For more information on defining custom types, read ["Configure segmentation types"](#).

The attribute you configure in `props.conf` to use these types depends on whether you're configuring index-time or search-time segmentation:

- For index-time segmentation, use the `SEGMENTATION` attribute.
- For search-time segmentation, use the `SEGMENTATION-<segment_selection>` attribute.

You can define either one of the attributes or both together in the stanza.

Add your stanza to `$SPLUNK_HOME/etc/system/local/props.conf`.

Index-time segmentation

The `SEGMENTATION` attribute determines the segmentation type used at index time. Here's the syntax:

```
[<spec>]
SEGMENTATION = <seg_rule>
```

[<spec>] can be:

- `<sourcetype>`: A source type in your event data.
- `host::<host>`: A host value in your event data.
- `source::<source>`: A source of your event data.

```
SEGMENTATION = <seg_rule>
```

- This specifies the type of segmentation to use at index time for [<spec>] events.
- `<seg_rule>`
 - ◆ A segmentation type, or "rule", defined in `segmenters.conf`
 - ◆ Common settings are `inner`, `outer`, `none`, and `full`, but the default file contains other predefined segmentation rules as well.
 - ◆ Create your own custom rule by editing `$SPLUNK_HOME/etc/system/local/segmenters.conf`, as described in ["Configure segmentation types"](#).

Search-time segmentation

The `SEGMENTATION-<segment_selection>` attribute helps determine the segmentation type used at search time. Here's the syntax:

[<spec>]
SEGMENTATION-<segment_selection> = <seg_rule>

[<spec>] can be:

- <sourcetype>: A source type in your event data.
- host::<host>: A host value in your event data.
- source::<source>: A source of your event data.

SEGMENTATION-<segment_selection> = <seg_rule>

- This specifies the type of segmentation to use at search time in Splunk Web for [<spec>] events.
- <segment_selection> can be one of the following: `full`, `inner`, `outer`, or `raw`.
 - ◆ These four values are the set of options displayed in the **Event segmentation** dropdown box in the **Results display options** panel, invoked from **Options** directly above search results in Splunk Web.
 - ◆ Note that these values are just the set of available Splunk Web dropdown options. You use this attribute to specify the actual segmentation type that the option invokes, which might not be of the same name as the dropdown option itself. For example, you could even define the "inner" dropdown option to invoke the "outer" segmentation type, not that you'd likely want to.
 - ◆ By mapping the dropdown option to a <seg_rule>, a user can later specify the option when looking at search results to set search-time segmentation, as described in ["Set search-time segmentation in Splunk Web"](#).
- <seg_rule>
 - ◆ A segmentation type, or "rule", defined in `segmenters.conf`
 - ◆ Common settings are `inner`, `outer`, `none`, and `full`, but the default file contains other predefined segmentation rules as well.
 - ◆ Create your own custom rule by editing `$SPLUNK_HOME/etc/system/local/segmenters.conf`, as described in ["Configure segmentation types"](#).

Example

This example sets both index-time and search-time segmentation rules for `syslog` events.

Add the following to the `[syslog]` source type stanza in `props.conf`:

```
[syslog]
SEGMENTATION = inner
SEGMENTATION-full= inner
```

This stanza changes the index-time segmentation for all events with a `syslog` source type to inner segmentation. It also causes the `full` radio button in Splunk Web to invoke inner segmentation for those same events.

Note: You must restart Splunk Enterprise to apply changes to search-time segmentation. You must re-index your data to apply index-time segmentation changes to existing data.

Set search-time event segmentation in Splunk Web

Splunk Web allows you to set segmentation for search results. While this has nothing to do with index-time segmentation, search-time segmentation in Splunk Web affects browser interaction and can speed up search results.

To set search-result segmentation:

1. Perform a search. Look at the results.
2. Click **Options...** above the returned set of events.
3. In the **Event Segmentation** dropdown box, choose from the available options: full, inner, outer, or raw. The default is "full".

You can configure the meaning of these dropdown options, as described in ["Set the segmentation for event data"](#).

Improve the data input process

Use a test index to test your inputs

Before adding new inputs to your production index, it is best to test them out. Add the inputs to a test index. Once you've verified that you're receiving the right data inputs and that the resulting events are in a usable form, you can point the inputs to your default "main" index. You can continue to test new inputs this way over time.

If you find that the inputs you started with are not the ones you want, or that the indexed events don't appear the way you need them to, you can keep working with the test index until you get results you like. When things start looking good, you can edit the inputs to point to your main index instead.

You can preview how Splunk Enterprise will index your data into a test index. During preview, you can adjust some event processing settings interactively. See ["The "Set Sourcetype" page"](#) for details.

Use a test index

To learn how to create and use custom indexes, read "Set up multiple indexes" in the Managing Indexers and Clusters manual. There are a few basic steps, described in detail in that topic:

1. Create the test index, using Splunk Web or the CLI or by editing `indexes.conf` directly. See "Set up multiple indexes" for details.
2. When [configuring the data inputs](#), route events to the test index. You can usually do this in Splunk Web. For each input:
 - a. When configuring the input from the **Add data** page, check the **More settings** option. It reveals several new fields, including one called **Index**.
 - b. In the **Index** dropdown box, select your test index. All events for that data input will now go to that index.
 - c. Repeat this process for each data input that you want to send to your test index.

You can also specify an index when configuring an input in `inputs.conf`, as described here.

3. When you search, specify the test index in your search command. (By default, Splunk Enterprise searches on the "main" index.) Use the `index=` command:

```
index=test_index
```

Note : When searching a test index for events coming in from your newly created input, Splunk recommends that you use the *Real-time > All time(real-time)* time range for the fields sidebar. The resulting **real-time search** will show all events being written to that index regardless of the value of their extracted time stamp. This is particularly useful if you are indexing historical data into your index that a search for "Last hour" or "Real-time > 30 minute window" would not show.

Delete indexed data and start over

If you want to clean out your test index and start over again, use the CLI `clean` command, described here.

Point your inputs at the default index

Once you're satisfied with the results and are ready to start indexing for real, you'll want to edit your data inputs so that they point to the default, "main" index, instead of the test index. This is a simple process, just the reverse of the steps you took to use the test index in the first place. For each data input that you've already set up:

1. Go back to the place where you initially configured the input. For example, if you configured the input from the **Add data** page in Splunk Web, return to the configuration screen for that input:

a. Select **System > System configurations > Data inputs**.

b. Select the input's data type to see a list of all configured inputs of that type.

c. Select the specific data input that you want to edit. This will take you to a screen where you can edit it.

d. Select the **Display advanced settings** option. Go to the field named **Index**.

e. In the **Index** dropdown box, select the **main** index. All events for that data input will now go to that index.

If you instead used `inputs.conf` to configure an input, you can change the index directly in that file, as described here.

2. Now when you search, you no longer need to specify an index in your search command. By default, Splunk Enterprise searches on the "main" index.

Use persistent queues to help prevent data loss

Persistent queuing lets you store data in an input queue to disk. This can help prevent data loss if the **forwarder** or **indexer** gets backed up.

By default, forwarders and indexers have an in-memory input queue of 500KB. If the input stream runs at a faster rate than the forwarder or indexer can process, to a point where the queue is maxed out, undesired consequences occur. In the case of UDP, data drops off the queue and gets lost. For other input types, the application generating the data gets backed up.

By implementing persistent queues, you can help prevent this from happening. With persistent queuing, once the in-memory queue is full, the forwarder or indexer writes the input stream to files on disk. It then processes data from the queues (in-memory and disk) until it reaches the point when it can again start processing directly from the data stream.

Note: While persistent queues help prevent data loss if Splunk Enterprise gets backed up, you can still lose data if Splunk Enterprise crashes. For example, Splunk holds some input data the in-memory queue, as well as in the persistent queue files. The in-memory data can get lost if a crash occurs. Similarly, data that is in the parsing or indexing pipeline but that has not yet been written to disk can get lost in the event of a crash.

When can you use persistent queues?

Persistent queuing is available for certain types of inputs, but not all. Generally speaking, it is available for inputs of an ephemeral nature, such as network inputs, but not for inputs that have their own form of persistence, such as file monitoring.

Persistent queues are available for these input types:

- TCP
- UDP

- FIFO
- Scripted inputs
- Windows Event Log inputs

Persistent queues are not available for these input types:

- Monitor
- Batch
- File system change monitor
- splunktcp (input from Splunk forwarders)

Configure a persistent queue

Use the inputs.conf file to configure a persistent queue.

Inputs do not share queues. You configure a persistent queue in the stanza for the specific input.

Syntax

To create the persistent queue, specify these two attributes within the particular input's stanza:

```
persistentQueueSize = <integer>(KB|MB|GB|TB)
* Max size of the persistent queue file on disk.
* Defaults to 0 (no persistent queue).
```

Example

Here's an example of specifying a persistent queue for a tcp input:

```
[tcp://9994]
persistentQueueSize=100MB
```

Persistent queue location

The persistent queue has a hardcoded location, which varies according to the input type.

For network inputs, the persistent queue is located here:

```
$SPLUNK_HOME/var/run/splunk/[tcpin|udpin]/pq__<port>
```

Note: There are *two* underscores in the file name: `pq__<port>`, *not* `pq_<port>`.

For example:

- The persistent queue for TCP port 2012:
`$SPLUNK_HOME/var/run/splunk/tcpin/pq__2012`
- The persistent queue for UDP port 2012:
`$SPLUNK_HOME/var/run/splunk/udpin/pq__2012`

For FIFO inputs, the persistent queue resides under
`$SPLUNK_HOME/var/run/splunk/fifoin/<encoded path>`.

For scripted inputs, it resides under
`$SPLUNK_HOME/var/run/splunk/exec/<encoded path>`. The FIFO/scripted input stanza in `inputs.conf` derives the `<encoded path>`.

Troubleshoot the input process

This topic discusses some initial steps you can take to troubleshoot the data input process.

Determine why you do not find the events you expect

When you add an input to Splunk Enterprise, that input gets added relative to the app you are in. Some apps write input data to a specific index. If you cannot find data that you are certain is in Splunk Enterprise, confirm that you are looking at the right index. You might want to add indexes to the list of default indexes for the role you are using.

- For more information about roles, refer to the topic about roles in the *Securing Splunk Enterprise* manual.
- For more information about troubleshooting data input issues, read the rest of this topic or see *I can't find my data!* in the *Troubleshooting Manual*.

Note: When you add inputs by editing `inputs.conf`, Splunk Enterprise might not immediately recognize them unless you restart. Splunk Enterprise looks for inputs every 24 hours, starting from the time it was last restarted. This means that if you add a new stanza to monitor a directory or file, it could take up to 24 hours for Splunk Enterprise to start indexing the contents of that directory or file. To ensure that your input is immediately recognized and indexed, add the input

through Splunk Web or CLI, or restart Splunk services after making edits to `inputs.conf`.

Troubleshoot your tailed files

You can use the `FileStatus` Representational State Transfer (REST) endpoint to get the status of your tailed files. For example:

```
curl  
https://serverhost:8089/services/admin/inputstatus/TailingProcessor:FileStatus
```

You can also monitor the fishbucket.

The fishbucket is a subdirectory within the Splunk Enterprise directory that keeps a record about each file input. The fishbucket keeps track of how far into a file that Splunk Enterprise has read, so that if you stop and restart `splunkd`, it knows where in each file input to resume reading. The fishbucket is at

`$SPLUNK_DB/fishbucket/splunk_private_db`.

To monitor the fishbucket, use the REST endpoint. Review the REST API Reference manual for additional information.

Troubleshoot monitor inputs

For a variety of information on dealing with monitor input issues, read "Troubleshooting Monitor Inputs" in the Community Wiki.

Can't find forwarded data?

Confirm that the forwarder functions properly and is visible to the indexer. You can use the Distributed Management Console (DMC) to troubleshoot Splunk topologies and get to the root of any forwarder issues. Read the *Distributed Management Console* manual for details.