Splunk® Enterprise Admin Manual 9.2.2

Generated: 7/26/2024 12:03 pm

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Welcome to Splunk Enterprise administration

How to use this manual

This manual provides information about the different ways you can administer Splunk Enterprise. It also introduces you to some initial administration tasks for Windows and *nix.

> Unless otherwise stated, tasks and processes in this manual are suitable for both Windows and *nix operating systems.

For a bigger picture overview of the Splunk administration process, including tasks not described in this manual (such as setting up users or data and security configuration), see "Splunk Administration: The big picture," in this manual.

For a list and simple description of the other manuals available to Splunk users, see "Other manuals for the Splunk administrator".

What you can do with the Administration Manual

Task:	Look here:
Start Splunk and do some initial configuration	All the things you need to do to get started on Splunk, from starting Splunk and installing your license, to binding Splunk to an IP. See: "What to do first" for more information.
Use Splunk Web to configure and administer Splunk	An overview of Splunk Web and how you can use it to administer Splunk. See "Use Splunk Web" for more information.
Use configuration files to configure and administer Splunk	A discussion about configuration files: where to find them, how to create and edit them, and some important stuff about file precedences. See "About configuration files" to get started.
Use the Splunk command line interface (CLI) to configure and administer Splunk	An overview of how to use the Command Line Interface to configure Splunk. See "About the CLI" for more information.
Optimize Splunk on Windows	Some Windows-specific things you should know about working with Splunk, including some tips for optimal deployment and information about working with system images. See "Introduction for Windows admins" for more information.
Learn about Splunk licenses	Install your license then go here to learn everything you need to know about Splunk licenses: "Manage Splunk licenses" for more information.
Get familiar with Splunk apps	An introduction and overview of Splunk Apps and how you might integrate them into your Splunk configuration. See "Meet Splunk apps" for more information.
Manage user settings	The Manage users chapter shows you how to manage settings for users. For more information about creating users, see Users and role-based access control in the Securing Splunk Enterprise manual.

Splunk platform administration: the big picture

The Admin Manual provides information about the initial administration tasks as well as information about the different methods you can use to administer your Splunk software. For a more specific overview of what you can do with the Admin Manual, see How to use this manual.

Below are administration tasks you might want to do after initial configuration and where to go to learn more.

Task:	Look here:
Perform backups	Back up configuration information Back up indexed data Set a retirement and archiving policy
Define alerts	The *Alerting Manual*
Manage search jobs	Manage search jobs

For more administration help, see the manuals described below.

Install and upgrade Splunk Enterprise

The Installation Manual describes how to install and upgrade Splunk Enterprise. For information on specific tasks, start here.

Task:	Look here:
Understand installation requirements	Plan your installation
Estimate hardware capacity needs	Estimate hardware requirements
Install Splunk	Install Splunk Enterprise on Windows Install Splunk Enterprise on Unix, Linux, or MacOS
Upgrade Splunk Enterprise	Upgrade from an earlier version

Get data in

Getting Data In is the place to go for information about data inputs: how to consume data from external sources and how to enhance the value of your data.

Task:	Look here:
Learn how to consume external data	How to get data into Splunk
Configure file and directory inputs	Get data from files and directories
Configure network inputs	Get network events
Configure Windows inputs	Get Windows data
Configure miscellaneous inputs	Other ways to get data in
Enhance the value of your data	Configure event processing Configure timestamps Configure indexed field extraction Configure host values Configure source types Manage event segmentation

Task:	Look here:
	Use lookups and workflow actions
See how your data will look after indexing	Preview your data
Improve the process	Improve the data input process

Manage indexes and indexers

Managing Indexers and Clusters tells you how to configure indexes. It also explains how to manage the components that maintain indexes: indexers and clusters of indexers.

Task:	Look here:
Learn about indexing	Indexing overview
Manage indexes	Manage indexes
Manage index storage	Manage index storage
Back up indexes	Back up indexed data
Archive indexes	Set a retirement and archiving policy
Learn about clusters and index replication	About clusters and index replication
Deploy clusters	Deploy clusters
Configure clusters	Configure clusters
Manage clusters	Manage clusters
Learn about cluster architecture	How clusters work

Scale Splunk platform deployments

The Distributed Deployment Manual describes how to distribute Splunk platform functionality across multiple components, such as forwarders, indexers, and search heads. Associated manuals cover distributed components in detail:

- The Forwarding Data Manual describes forwarders.
- The Distributed Search Manual describes search heads.
- The Updating Splunk Components Manual explains how to use the deployment server and forwarder management to manage your deployment.

Task:	Look here:
Learn about distributed Splunk platform deployments	Scale deployments
Perform capacity planning for Splunk platform deployments	Estimate hardware requirements
Learn how to forward data	Forward data
Distribute searches across multiple indexers	Search across multiple indexers
Update the deployment	Deploy configuration updates across your environment

Secure Splunk Enterprise

Securing Splunk tells you how to secure your Splunk Enterprise deployment.

Task:	Look here:
Authenticate users and edit roles	User and role-based access control
Secure data with SSL	Secure authentication and encryption
Audit Splunk software	Audit system activity
Use Single Sign-On (SSO) with Splunk software	Configure Single Sign-on
Use Splunk software with LDAP	Set up user authentication with LDAP

Troubleshoot Splunk software

The Troubleshooting Manual provides overall guidance on Splunk platform troubleshooting. In addition, topics in other manuals provide troubleshooting information on specific issues.

Task:	Look here:
Learn about Splunk platform troubleshooting tools	First steps
Learn about Splunk log files	Splunk log files
Work with Splunk support	Contact Splunk support
Resolve common problems	Some common scenarios

References and other information

The Splunk documentation includes several useful references, as well as some other sources of information that might be of use to the Splunk software administrator.

Reference:	Look here:
Configuration file reference	Configuration file reference in the Admin Manual
REST API reference	REST API Reference Manual
CLI help	Available through installed instances of Splunk Enterprise. For details on how to invoke it, read Get help with the CLI in the Admin Manual.
Release information	Release Notes
Information on managing Splunk platform knowledge objects	Knowledge Manager Manual

Other manuals for the Splunk platform administrator

The *Admin Manual* is one of several books with important information and procedures for the Splunk Enterprise administrator. But it's just the beginning of what you can do with Splunk Enterprise.

If you need to configure, run, or maintain Splunk Enterprise as a service for yourself or other users, start with this book. Then go to these other manuals for details on specific areas of Splunk Enterprise administration.

Manual	What it covers	Key topic areas
Getting Data In	Specifying data inputs and improving how Splunk software handles data	**How to get data into Splunk** **Configure event processing** **Preview your data**

Manual	What it covers	Key topic areas
Managing Indexers and Clusters	Managing Splunk indexers and clusters of indexers	**About indexing and indexers** **Manage indexes** **Back up and archive your indexes** **About clusters and index replication** **Deploy clusters**
Distributed Deployment	Scaling your deployment to fit the needs of your enterprise.	**Distributed Splunk overview**
Forwarding Data	Forwarding data into Splunk.	**Forward data**
Distributed Search	Using search heads to distribute searches across multiple indexers.	**Search across multiple indexers**
Updating Splunk Components	Using the deployment server and forwarder management to update Splunk components such as forwarders and indexers.	**Deploy updates across your environment**
Securing Splunk	Data security and user authentication	**User authentication and roles** **Encryption and authentication with SSL** **Auditing**
Monitoring Splunk Enterprise	Use included dashboards and alerts to monitor and troubleshoot your Splunk Enterprise deployment	**About the monitoring console**
Troubleshooting	Solving problems	**First steps** **Splunk log files** **Some common scenarios**
Installation	Installing and upgrading Splunk	**System requirements** **Step by step installation procedures** **Upgrade from an earlier version**

The topic "Learn to administer Splunk" provides more detailed guidance on where to go to read about specific admin tasks.

Other books of interest to the Splunk administrator

In addition to the manuals that describe the primary administration tasks, you might want to visit other manuals from time to time, depending on the size of your Splunk Enterprise installation and the scope of your responsibilities. These are other manuals in the Splunk Enterprise documentation set:

- **Search Tutorial**. This manual provides an introduction to searching with Splunk.
- **Knowledge Manager**. This manual describes how to manage Splunk knowledge objects, such as event types, tags, lookups, field extractions, workflow actions, saved searches, and views.
- **Alerting**. This manual describes Splunk's alerting and monitoring functionality.
- **Data Visualizations**. This manual describes the range of visualizations that Splunk provides.
- **Search Manual**. This manual tells you how to search and how to use the Splunk search language.
- **Search Reference**. This reference contains a detailed catalog of the Splunk search commands.
- **Developing Views and Apps for Splunk Web**. This manual explains how to develop views and apps using advanced XML. It also contains other developer topics, such as custom scripts and extending Splunk.
- **REST API Reference**. This manual provides information on all publicly accessible REST API endpoints.
- **Release Notes**. Look here for information about new features, known issues, and fixed problems.

The larger world of Splunk documentation

For links to the full set of Splunk Enterprise documentation, including the manuals listed above, visit: **Splunk Enterprise documentation**.

To access all the Splunk documentation, including manuals for apps, go to this page: **Welcome to Splunk documentation**.

Make a PDF

If you'd like a PDF version of this manual, click the red **Download the Admin Manual as PDF** link below the table of contents on the left side of this page. A PDF version of the manual is generated on the fly. You can save it or print it to read later.

Introduction for Windows admins

Welcome!

Splunk is a powerful, effective tool for Windows administrators to resolve problems that occur on their Windows networks. Its out-of-the-box feature set positions it to be the secret weapon in the Windows administrator's toolbox. The ability to add apps that augment its functionality makes it even more extensible. And it has a growing, thriving community of users.

How to use this manual as a Windows user

This manual has topics that will help you experiment with, learn, deploy, and get the most out of Splunk.

Unless otherwise specified, the information in this manual is helpful for both Windows and *nix users. If you are unfamiliar with Windows or *nix operational commands, we strongly recommend you check out Differences between *nix and Windows in Splunk operations.

We've also provided some extra information in the chapter "get the most out of Splunk on Windows". This chapter is intended for Windows users to help you make the most of Splunk and includes the following information.

Deploy Splunk on Windows provides some considerations and preparations specific to Windows users. Use this topic when you plan your deployment.

Optimize Splunk for peak performance describes ways to keep your Splunk on Windows deployment running properly, either during the course of the deployment, or after the deployment is complete.

Put Splunk onto system images helps you make Splunk a part of every Windows system image or installation process. From here you can find tasks for installing Splunk and Splunk forwarders onto your system images.

For more information

Here's some additional Windows topics of interest in other Splunk manuals:

- An overview of all of the installed Splunk for Windows services (from the Installation Manual)
- What Splunk can monitor (from the Getting Data In Manual)
- Considerations for deciding how to monitor remote Windows data (from the Getting Data In Manual). Read this

topic for important information on how to get data from multiple machines remotely.
- About the universal forwarder (from the Universal Forwarder Manual)

Other useful information:

- Where is my data? (from the Getting Data In Manual)
- Use Splunk's Command Line Interface (CLI) (from the Getting Data In Manual)
- Sources, sourcetypes and fields (from the Getting Data In Manual)
- Fields and field extraction (from the Knowledge Manager Manual)

If you need help

If you are looking for in-depth Splunk knowledge, a number of education programs are available.

When you get stuck, Splunk has a large free support infrastructure that can help:

- Splunk Answers.
- The Splunk Community Wiki.
- The Splunk Internet Relay Chat (IRC) channel (EFNet #splunk). (IRC client required)

If you still don't have an answer to your question, you can get in touch with Splunk's support team. The Support Contact page tells you how to do that.

Note: Levels of support above the community level require an Enterprise license. To get one, you'll need to speak with the Sales team.

Optimize Splunk Enterprise for peak performance

This topic discusses standards that assist the system administrator when implementing or expanding their Splunk Enterprise infrastructure, and in maintaining consistent performance:

- **Designate one or more machines solely for Splunk Enterprise components.** Splunk scales horizontally. Adding more physical machines dedicated to Splunk Enterprise translates into better performance than having more resources in a single machine. Where possible, split up your indexing and searching activities across a number of machines, and only run one Splunk Enterprise component on each machine. Performance is reduced when you run Splunk Enterprise on machines that share resources with other services.

- **Provide dedicated, fast storage to your Splunk Enterprise indexers.** Insufficient storage I/O is the most commonly encountered limitation in a Splunk software infrastructure. For guidance on storage for indexers, see What storage type should I use for a role? in the *Capacity Planning Manual*.

- **Don't allow anti-virus programs to scan disks used for Splunk services.** When an anti-virus product scans files for viruses on access, performance of Splunk services is significantly reduced, especially as the recently indexed data ages. If you use anti-virus programs on the servers running Splunk Enterprise, make sure that all Splunk software directories and programs are excluded from on-access file scans.

- **Use multiple indexes, where possible.** Distribute the data that in indexed by Splunk into different indexes. Sending all data to one index can cause I/O bottlenecks on your system and complicate retention calculations and access controls. For information on how to configure indexes, see Configure your indexes in the *Managing Indexers and Clusters of Indexers* manual.

- **Don't store your indexes on the same physical disk or volume as the operating system.** The disk that holds your operating system or its swap file is not a recommended place for Splunk Enterprise data storage. Put your indexes on other disks or volumes mounted on the machine. For more information on how indexes are stored, including information on database bucket types and how Splunk stores and ages them, see How Splunk stores indexes in the *Managing Indexers and Clusters of Indexers* manual.

- **Don't store the hot and warm buckets of your indexes on network volumes.** Network latency will decrease indexing performance significantly. Always use fast, local disk for the index hot and warm buckets. You can specify network shares for the cold and frozen buckets of an index using Distributed File System (DFS) volumes or Network File System (NFS) mounts. But searches that include data stored on network volumes will be slower.

- **Maintain disk availability, bandwidth, and space on your indexers.** Make sure that the disk volumes or mounts that hold the indexes maintain free space at all times. Disk performance decreases as available space decreases, and disk seek times will increase. Slow storage affects how efficiently Splunk Enterprise indexes data, and will also impact how quickly search results, reports and alerts are returned. The volume or mount that contains your indexes must have approximately 5 gigabytes of free disk space by default, or indexing will stop.

Differences between *nix and Windows in Splunk operations

This topic clarifies the functional differences that you'll encounter between *nix and Windows operating systems, under the context in which they matter in Splunk operations. It does not delve into technical comparisons of - or advocacy for - either flavor of OS, but rather explains why you'll see things referenced one way or another on various OS-specific Splunk manual pages.

Paths

A major difference in the way that *nix operating systems handle files and directories is the type of slash used to separate files or directories in the pathname. *nix systems use the forward slash, ("/"). Windows, on the other hand, uses the backslash ("\").

An example of a *nix path:

```
/opt/splunk/bin/splunkd
```

An example of a Windows path:

```
C:\Program Files\Splunk\bin\splunkd.exe
```

Environment variables

Another area where the operating systems differ is in the representation of environment variables. Both systems have a way to temporarily store data in one or more environment variables. On *nix systems, this is shown by using the dollar sign ("$") in front of the environment variable name, like so:

```
# SPLUNK_HOME=/opt/splunk; export $SPLUNK_HOME
```

On Windows, it's a bit different - to specify an environment variable, you need to use the percent sign ("%"). Depending on the type of environment variable you are using, you may need to place one or two percent signs before the environment name, or on either side of the name.

```
> set SPLUNK_HOME="C:\Program Files\Splunk"
> echo %SPLUNK_HOME%
C:\Program Files\Splunk
>
```

To set the %SPLUNK_HOME% variable in the Windows environment, you can do one of two things:

- Edit `splunk-launch.conf` in %SPLUNK_HOME%\etc.
- Set the variable by accessing the "Environment Variables" window. Open an Explorer window, and on the left pane, right-click "My Computer", then select "Properties" from the window that appears. Once the System Properties window appears, select the "Advanced" tab, then click on the "Environment Variables" button that appears along the bottom window of the tab.

Configuration files

Splunk Enterprise works with configuration files that use ASCII/UTF-8 character set encoding. When you edit configuration files on Windows, configure your text editor to write files with this encoding. On some Windows versions, UTF-8 is not the default character set encoding. See How to edit a configuration file.

Ways you can configure Splunk software

Splunk software maintains its configuration information in a set of **configuration files**. You can configure Splunk by using any (or all!) of these methods:

- Use Splunk Web.
- Use Splunk's Command Line Interface (CLI) commands.
- Edit Splunk's configuration files directly.
- Use App setup screens that use the Splunk REST API to update configurations.

All of these methods change the contents of the underlying configuration files. You may find different methods handy in different situations.

Use Splunk Web

You can perform most common configuration tasks in Splunk Web. Splunk Web runs by default on port 8000 of the host on which it is installed:

- If you're running Splunk on your local machine, the URL to access Splunk Web is `http://localhost:8000`.
- If you're running Splunk on a remote machine, the URL to access Splunk Web is `http://<hostname>:8000`, where `<hostname>` is the name of the machine Splunk is running on.

Administration menus can be found under **Settings** in the Splunk Web menu bar. Most tasks in the Splunk documentation set are described for Splunk Web. For more information about Splunk Web, see Meet Splunk Web.

Edit configuration files

Most of Splunk's configuration information is stored in .conf files. These files are located under your Splunk installation directory (usually referred to in the documentation as `$SPLUNK_HOME`) under `/etc/system`. In most cases you can copy

these files to a local directory and make changes to these files with your preferred text editor.

Before you begin editing configuration files, read "About configuration files".

Use Splunk CLI

Many configuration options are available via the CLI. These options are documented in the CLI chapter in this manual. You can also get CLI help reference with the `help` command while Splunk is running:

```
./splunk help
```

For more information about the CLI, refer to "About the CLI" in this manual. If you are unfamiliar with CLI commands, or are working in a Windows environment, you should also check out Differences between *nix and Windows in Splunk operations.

Setup pages for an app

Developers can create setup pages for an app that allow users to set configurations for that app without editing the configuration files directly. Setup pages make it easier to distribute apps to different environments, or to customize an app for a particular usage.

Setup pages use Splunk's REST API to manage the app's configuration files.

For more information about setup pages, refer to Enable app configuration with setup pages in Splunk Cloud Platform or Splunk Enterprise on the Splunk Developer Portal.

Managing a distributed environment

The Splunk deployment server provides centralized management and configuration for distributed environments. You can use it to deploy sets of configuration files or other content to groups of Splunk instances across the enterprise.

For information about managing deployments, refer to the "Updating Splunk Components" manual.

Get the most out of Splunk Enterprise on Windows

Deploy Splunk Enterprise on Windows

You can integrate Splunk into your Windows environment in any number of ways. This topic discusses some of those scenarios and offers guidelines on how to best adapt your Splunk for Windows deployment to your enterprise.

While this topic is geared more toward deploying Splunk in a Windows environment, Splunk itself also has distributed deployment capabilities that you should be aware of, even as you integrate it into your Windows enterprise. The Distributed Deployment Manual has lots of information on spreading Splunk services across a number of computers.

When deploying Splunk on Windows on a large scale, you can rely completely on your own deployment utilities (such as System Center Configuration Manager or Tivoli/BigFix) to place both Splunk and its configurations on the machines in your enterprise. Or, you can integrate Splunk into system images and then deploy Splunk configurations and apps using Splunk's deployment server.

Concepts

When you deploy Splunk into your Windows network, it captures data from the machines and stores it centrally. Once the data is there, you can search and create reports and dashboards based on the indexed data. More importantly, for system administrators, Splunk can send alerts to let you know what is happening as the data arrives.

In a typical deployment, you dedicate some hardware to Splunk for indexing purposes, and then use a combination of universal forwarders and Windows Management Instrumentation (WMI) to collect data from other machines in the enterprise.

Considerations

Deploying Splunk in a Windows enterprise requires a number of planning steps.

First, you must inventory your enterprise, beginning at the physical network, and leading up to how the machines on that network are individually configured. This includes, but is not limited to:

- Counting the number of machines in your environment and defining a subset of those which need Splunk installed. Doing this defines the initial framework of your Splunk topology.
- Calculating your network bandwidth, both in your main site and at any remote or external sites. Doing this determines where you will install your main Splunk instance, and where and how you will use Splunk forwarders.
- Assessing the current health of your network, particularly in areas where networks are separated. Making sure your edge routers and switches are functioning properly will allow you to set a baseline for network performance both during and after the deployment.

Then, you must answer a number of questions prior to starting the deployment, including:

- **What data on your machines needs indexing? What part of this data do you want to search, report, or alert across?** This is probably the most important consideration to review. The answers to these questions determine how you address every other consideration. It determines where to install Splunk, and what types of Splunk you use in those installations. It also determines how much computing and network bandwidth Splunk will potentially use.

- **How is the network laid out? How are any external site links configured? What security is present on those links?** Fully understanding your network topology helps determine which machines you should install Splunk on, and what types of Splunk (indexers or forwarders) you should install on those machines from a networking standpoint.

A site with thin LAN or WAN links makes it necessary to consider how much Splunk data should be transferred between sites. For example, if you have a hub-and-spoke type of network, with a central site connected to branch sites, it might be a better idea to deploy forwarders on machines in the branch sites, which send data to an intermediate forwarder in each branch. Then, the intermediate forwarder would send data back to the central site. This is a less costly move than having all machines in a branch site forward their data to an indexer in the central site.

If you have external sites that have file, print or database services, you'll need to account for that traffic as well.

- **How is your Active Directory (AD) configured?** How are the operations masters roles on your domain controllers (DCs) defined? Are all domain controllers centrally located, or do you have controllers located in satellite sites? If your AD is distributed, are your bridgehead servers configured properly? Is your Inter-site Topology Generator (ISTG)-role server functioning correctly? If you are running Windows Server 2008 R2, do you have read-only domain controllers (RODCs) in your branch sites? If so, then you have to consider the impact of AD replication traffic as well as Splunk and other network traffic.

- **What other roles are the servers in your network playing?** Splunk indexers need resources to run at peak performance, and sharing servers with other resource-intensive applications or services (such as Microsoft Exchange, SQL Server and even Active Directory itself) can potentially lead to problems with Splunk on those machines. For additional information on sharing server resources with Splunk indexers, see "Introduction to capacity planning for Splunk Enterprise" in the Capacity Planning Manual.

- **How will you communicate the deployment to your users?** A Splunk installation means the environment is changing. Depending on how Splunk is rolled out, some machines will get new software installed. Users might incorrectly link these new installs to perceived problems or slowness on their individual machine. You should keep your user base informed of any changes to reduce the number of support calls related to the deployment.

Prepare your Splunk on Windows deployment

How you deploy Splunk into your existing environment depends on the needs you have for Splunk, balanced with the available computing resources you have, your physical and network layouts, and your corporate infrastructure. As there is no one specific way to deploy Splunk, there are no step-by-step instructions to follow. There are, however, some general guidelines to observe.

For a more successful Splunk deployment:

- **Prepare your network.** Before integrating Splunk into your environment:
    - Make sure that your network is functioning properly, and that all switches, routers and cabling are correctly configured.
    - Replace any broken or failing equipment.
    - Ensure any virtual LANs (VLANs) are properly set up.
    - Test network throughput, particularly between sites with thin network links.

- **Prepare your Active Directory.** While AD is not a requirement to run Splunk, it's a good idea to ensure that it is functioning properly prior to your deployment. This includes but is not limited to:
    - Identifying all of your domain controllers, and the operations master roles any of them might perform. If you have RODCs at your branch sites, make sure that they have the fastest connections as possible to

operations masters DCs.
    - Ensuring that AD replication is functioning correctly, and that all site links have a DC with a copy of the global catalog.
    - If your forest is divided into multiple sites, make sure your ISTG role server is functioning properly, or that you have assigned at least two bridgehead servers in your site (one primary, one backup).
    - Ensuring that your DNS infrastructure is working properly.

You might need to place DCs on different subnets on your network, and seize flexible single master operations (FSMO, or operations master) roles as necessary to ensure peak AD operation and replication performance during the deployment.

- **Define your Splunk deployment.** Once your Windows network is properly prepared, you must now determine where Splunk will go in the network. Consider the following:
    - Determine the set(s) of data that you want Splunk to index on each machine, and whether or not you need for Splunk to send alerts on any collected data.
    - Dedicate one or more machines in each network segment to handle Splunk indexing, if possible. For additional information on capacity planning for a distributed Splunk deployment, review "Introduction to capacity planning for Splunk Enterprise" in the Capacity Planning Manual.
    - Don't install full Splunk on machines that run resource-intensive services like AD (in particular, DCs that hold FSMO roles), any version of Exchange, SQL Server, or machine virtualization product such as Hyper-V or VMWare. Instead, use a universal forwarder, or connect to those machines using WMI.
    - If you're running Windows Server 2008/2008 R2 Core, remember that you'll have no GUI available to make changes using Splunk Web when you install Splunk on those machines.
    - Arrange your Splunk layout so that it uses minimal network resources, particularly across thin WAN links. Universal forwarders greatly reduce the amount of Splunk-related traffic sent over the wire.

- **Communicate your deployment plans to your users.** It's important to advise your users about the status of the deployment, throughout the course of it. This will significantly reduce the amount of support calls you receive later.

Put Splunk Enterprise onto system images

This topic explains the concepts of making Splunk Enterprise a part of every Windows system image or installation process. It also guides you through the general process of integration, regardless of the imaging utilities that you use.

- For more specific information about getting Windows data into the Splunk platform, review Monitoring Windows data with Splunk Enterprise in the *Getting Data In* manual.
- For information on distributed Splunk Enterprise deployments, read Distributed overview in the *Distributed Deployment Manual*. This overview is essential reading for understanding how to set up Splunk platform deployments, irrespective of the operating system that you use. For information about the distributed deployment capabilities of Splunk Enterprise, see About deployment server and forwarder management in *Updating Splunk Enterprise Instances*.
- For information about planning larger Splunk platform deployments, read Introduction to capacity planning for Splunk Enterprise in the *Capacity Planning Manual* and Deploy Splunk Enterprise on Windows in this manual.

Concepts for system integration on Windows

The main reason to integrate Splunk Enterprise into Windows system images is to ensure that Splunk Enterprise is available immediately when the machine is activated for use in the enterprise. This frees you from having to install and configure Splunk Enterprise after activation.

In this scenario, when a Windows system is activated, it immediately launches Splunk Enterprise after booting. Then, depending on the type of Splunk Enterprise instance installed and the configuration given, Splunk Enterprise either collects data from the machine and forwards it to an indexer (in many cases), or begins indexing data that is forwarded from other Windows machines.

System administrators can also configure Splunk Enterprise instances to contact a **deployment server**, which allows for further configuration and update management.

In many typical environments, universal forwarders on Windows machines send data to a central indexer or group of indexers, which then allow that data to be searched, reported and alerted on, depending on your specific needs.

Considerations for system integration

Integrating Splunk Enterprise into your Windows system images requires planning.

In most cases, the preferred Splunk Enterprise component to integrate into a Windows system image is a **universal forwarder**. The universal forwarder is designed to share resources on computers that perform other roles, and does much of the work that an indexer can, at much less cost. You can also modify the forwarder's configuration using the deployment server or an enterprise-wide configuration manager with no need to use Splunk Web to make changes.

In some situations, you may want to integrate a full instance of Splunk Enterprise into a system image. Where and when this is more appropriate depends on your specific needs and resource availability.

You should not include a full version of Splunk Enterprise in an image for a server that performs any other type of role, unless you have specific need for the capability that an indexer has over a forwarder. Installing multiple indexers in an enterprise does not give you additional indexing power or speed, and can lead to undesirable results.

Before integrating Splunk Enterprise into a system image, consider:

- **the amount of data you want Splunk Enterprise to index, and where you want it to send that data, if applicable.** This feeds directly into disk space calculations, and should be a top consideration.
- **the type of Splunk Enterprise instance to install on the image or machine.** Universal forwarders have a significant advantage when installing on workstations or servers that perform other duties, but might not be appropriate in some cases.
- **the available system resources on the imaged machine.** How much disk space, RAM and CPU resources are available on each imaged system? Will it support a Splunk Enterprise installation?
- **the resource requirements of your network.** Splunk Enterprise needs network resources, whether you're using it to connect to remote machines using WMI to collect data, or you're installing forwarders on each machine and sending that data to an indexer.
- **the system requirements of other programs installed on the image.** If Splunk Enterprise is sharing resources with another server, it can take available resources from those other programs. Consider whether or not you should install other programs on a workstation or server that is running a full instance of Splunk Enterprise. A universal forwarder will work better in cases like this, as it is designed to be lightweight.
- **the role that the imaged machine plays in your environment.** Will it be a workstation only running productivity applications like Office? Or will it be an operations master domain controller for your Active Directory forest?

Integrate Splunk Enterprise into a system image

Once you have determined the answers to the questions in the checklist above, the next step is to integrate Splunk Enterprise into your system images. The steps listed are generic, allowing you to use your favorite system imaging or configuration tool to complete the task.

Choose one of the following options for system integration:

- Integrate a universal forwarder into a system image
- Integrate a full version of Splunk Enterprise into a system image

Integrate a universal forwarder onto a system image

This topic discusses the procedure to integrate a Splunk universal forwarder into a Windows system image. For additional information about integrating Splunk Enterprise into images, see Integrate Splunk Enterprise into system images.

Install and configure Windows and applications

1. On a reference computer, install and configure Windows the way that you want, including installing Windows features, service packs, and other components.
2. Install and configure necessary applications, taking into account Splunk's system and hardware capacity requirements.
3. Install and configure the universal forwarder from the command line. You must supply at least the `LAUNCHSPLUNK=0` command line flag when you perform the installation.
4. Proceed through the graphical portion of the install, selecting the inputs, deployment servers, and/or forwarder destinations you want.
5. After the installation has completed, open a command prompt or PowerShell window.

Edit configurations and run clone-prep-clear-config

1. (Optional) Edit configuration files that were not configurable in the installer.
2. Change to the universal forwarder `bin` directory.
3. Run `./splunk clone-prep-clear-config`.
4. Exit the command prompt or PowerShell window.
5. In the Services Control Panel, configure the `splunkd` service to start automatically by setting its startup type to 'Automatic'.
6. Prepare the system image for domain participation using a utility such as Windows System Image Manager (WSIM). Microsoft recommends using `SYSPREP` or WSIM as the method to change machine Security Identifiers (SIDs) prior to cloning, as opposed to using third-party tools (such as Ghost Walker or NTSID.)

Clone and restore the image

1. Restart the machine and clone it with your favorite imaging utility.
2. After cloning the image, use the imaging utility to restore it into another physical or virtual machine.
3. Run the cloned image. Splunk services start automatically.
4. Use the CLI to restart Splunk Enterprise to remove the `cloneprep` information:
   splunk restart

   > You must restart Splunk Enterprise from the CLI to delete the cloneprep file. Restarting the Splunk service does not perform the deletion.
5. Confirm that the `$SPLUNK_HOME\cloneprep` file has been deleted.

The image is now ready for deployment.

Integrate full Splunk Enterprise onto a system image

This topic discusses the procedure to integrate a full version of Splunk into a Windows system image. For additional information about integrating Splunk into images, see "Put Splunk onto system images" in this manual.

To integrate a full version of Splunk into a system image:

1. Using a reference computer, install and configure Windows to your liking, including installing any needed Windows features, patches and other components.

2. Install and configure any necessary applications, taking into account Splunk's system and hardware capacity requirements.

3. Install and configure Splunk.

Important: You can install using the GUI installer, but more options are available when installing the package from the command line.

4. Once you have configured Splunk inputs, open a command prompt.

5. From this prompt, stop Splunk by changing to the `%SPLUNK_HOME%\bin` directory and issuing a `.\splunk stop`

6. Clean any event data by issuing a `.\splunk clean eventdata`.

7. Close the command prompt window.

8. Ensure that the `splunkd` and `splunkweb` services are set to start automatically by setting their startup type to 'Automatic' in the Services Control Panel.

9. Prepare the system image for domain participation using a utility such as SYSPREP (for Windows XP and Windows Server 2003/2003 R2) and/or Windows System Image Manager (WSIM) (for Windows Vista, Windows 7, and Windows Server 2008/2008 R2).

Note: Microsoft recommends using SYSPREP and WSIM as the method to change machine Security Identifiers (SIDs) prior to cloning, as opposed to using third-party tools (such as Ghost Walker or NTSID.)

10. Once you have configured the system for imaging, reboot the machine and clone it with your favorite imaging utility.

The image is now ready for deployment.

Administer Splunk Enterprise with Splunk Web

Launch Splunk Web

After Splunk is running, you can launch the Web interface, **Splunk Web**. To learn more about Splunk Web, see:

- Admin tasks with Splunk Web
- Navigating Splunk Web
- Using Splunk Search

To launch Splunk Web, navigate to:

```
http://mysplunkhost:<port>
```

Use the host and port you chose during installation.

The first time you log in to Splunk with an Enterprise license, login as the administrator you created at installation time.:
Username - *admin*
Password - <password>

> Splunk Free does not have access controls, so you will not be prompted for login information.

> You cannot access Splunk Free from a remote browser until you have edited $SPLUNK_HOME/etc/local/server.conf and set allowRemoteLogin to Always. If you are running Splunk Enterprise, remote login is disabled by default (set to requireSetPassword) for the admin user until you change the default password.

Admin tasks with Splunk Web

Splunk Web is the browser-based interface for the Splunk platform. Here are just a few of the things you can do in Splunk Web:

- Configure your data inputs
- Search data and report and visualize results
- Investigate problems
- Manage users natively or via LDAP strategies
- Troubleshoot Splunk deployments
- Manage clusters and peers

Refer to the system requirements for a list of supported operating systems and browsers.

Splunk Settings menu

Splunk Web provides a convenient interface for managing most aspects of Splunk platform operations. Most of the functions can be accessed by clicking **Settings** in the menu. From here you can:

Manage your data

Under **Settings > Data** you can do the following:

- **Data Inputs** Lets you view a list of data types and configure them. To add an input, click the **Add data** button in the Data Inputs page. For more information about how to add data, see the *Getting Data In* manual.
- **Forwarding and receiving** lets you set up your forwarders and receivers. For more information about setting up forwarding and receiving, see the Forwarding Data manual.
- **Indexes** lets you add, disable, and enable indexes.
- **Report acceleration summaries** takes you to the searching and reporting app to lets you review your existing report summaries. For more information about creating report summaries, see the *Knowledge Manager Manual*.

Manage users and user authentication

By navigating to **Settings > Users and Authentication > Access Control** you can do the following:

- Create and manage users
- Define and assign roles
- Set up LDAP authentication strategies

For more information about working with users and authentication, see *Securing Splunk Enterprise*.

Work with Apps

To see your installed **apps**, select **Apps** in the menu bar.

From this page, you can select an app from a list of those you have already installed and are currently available to you. From here you can also access the following menu options:

- **Find more Apps** lets you search for and install additional apps.
- **Manage Apps** lets you manage your existing apps.

You can also access all of your apps in the Home page.

For more information about apps, see Developing views and apps for Splunk Web.

Manage aspects of your system

The options under **Settings > System** let you do the following:

- **Server settings** lets you manage Splunk platform settings like ports, host name, index paths, email server, and system logging and deployment client information. For more about configuring and managing distributed environments with Splunk Web, see the Updating Splunk Components manual.
- **Server controls** lets you restart the Splunk platform.
- **Licensing** lets you manage and renew your Splunk licenses.

Splunk Enterprise summary dashboard

The summary dashboard is the first thing you see as you enter the Search & Reporting app. It provides a search bar and time range picker which you can use to input and run your initial search.

When you add an input to Splunk, that input gets added relative to the app you're in. Some apps, like the *nix and Windows apps, write input data to a specific index (in the case of *nix and Windows, that is the **os** index). If you review the summary dashboard and you don't see data that you're certain is in Splunk, be sure that you're looking at the right index.

You may want to add the index that an app uses to the list of default indexes for the role you're using. For more information about roles, refer to this topic about roles in Securing Splunk.For more information about Summary Dashboards, see the Search Tutorial.

Configure Dashboards Trusted Domains List

The Dashboards Trusted Domains List is a list of authorized domains and URLs that aid the management of external content. For example, external images without a domain or URL specified in the list will not render in the dashboard. To permit external content, you can add the content's domain or URL to the list. You can turn off the enforcement of the domain list by configuring your web-features.conf file.

Create or edit a local web-features.conf file

The following are prerequisites for editing configuration files:

- You must be a user with file system access. For example, a system administrator can edit configuration files.
- You must understand how the configuration system works across your deployment, including where to make the changes safely.

For steps on how to safely create or edit a configuration file, see Customize a configuration file.

Add the Dashboards Trusted Domains List to the web-features.conf file

Use the REST API to update the web-features.conf setting. Updates to web-features.conf are replicated across search heads and don't require a restart.

You must add your domain names under the system level local file. The following steps create a Dashboards Trusted Domains List:

1. Write your initial REST command. The command uses the following structure: `https://<host>:<mPort>/servicesNS/nobody/system/web-features/feature:dashboards_csp`. For example, your REST command might look like the following:

   ```
   curl -k -u admin:password
   https://localhost:8089/servicesNS/nobody/system/web-features/feature:dashboards_csp
   ```
2. Add the domains to the Dashboards Trusted Domains List. The setting name in your command must follow this format: `dashboards_trusted_domain.<label name>`. You must use unique label names for each URL. Using the same label name will overwrite any previously attached URL. Your command might look like the following:

   ```
   -d dashboards_trusted_domain.exampleLabel=https://example.com
   ```

Remove a domain

To remove a domain with the API, use the same label name and attach it with an empty string. The empty string will overwrite the previous domain URL.

To remove a domain without the API, you can edit the file manually. For more details, see How to edit a configuration file.

Example of configured dashboards_trusted_domains settings

Add authorized domains and URLs to the web-features.conf file, instead of the previously used web.conf file.

If you want to troubleshoot the Dashboards Trusted Domains List or add to the list directly, you can add authorized domains and URLs to the [feature:dashboards_csp] stanza in the web-features.conf file. Each setting will start with the syntax `dashboards_trusted_domain.` followed by the domain or URL name.

Domain and URL names can be specific or use an asterisk wildcard. The asterisk wildcard must be the leftmost domain in the domain name system. Asterisk wildcards in the middle or end of a domain name system do not work. For example, the domain name `*.buttercup-games.com` loads content from any subdomain under `buttercup-games.com`. The domain name `www.*.buttercup-games.com` is invalid.

The following is an example of configured dashboards_trusted_domains settings.

```
[feature:dashboards_csp]
  dashboards_trusted_domain.everything=*.buttercup-games.com
  dashboards_trusted_domain.example=example.buttercup-games.com
```

Subdomains allowed by default

The Dashboards Trusted Domains List (DTDL) allows select subdomains by default without adding the domains to the DTDL. Additionally, the subdomains do not trigger the content warning modals. The subdomains are part of an internal Splunk software list that is not visible to users.

The following lists the subdomains allowed by default:

- apps.splunk.com
- dev.splunk.com
- docs.flowmill.com
- docs.splunk.com
- help.rigor.com
- help.victorops.com
- lantern.splunk.com
- splunkbase.com
- splunkbase.splunk.com
- splunkui.splunk.com
- splunk.com/download
- splunk.com/products

External content and redirection feature settings

Do not set the feature settings to false. Turning the feature settings to false removes safeguards for external content and external redirection modals.

Dashboard Studio and Classic SimpleXML dashboards use feature settings in web-features.conf to turn the enforcement of the Dashboards Trusted Domains List on and off.

`Enable_dashboards_external_content_restriction` is true by default and shows the external content warning if a domain or URL is not in the Dashboards Trusted Domains List.

`Enable_dashboards_redirection_restriction` is true by default and shows the redirection warning modal if a domain or URL is not in the Dashboards Trusted Domains List.

The following is an example of configured external content and redirection feature settings set to true:

```
[feature:dashboards_csp]
enable_dashboards_external_content_restriction=true
enable_dashboards_redirection_restriction=true
```

Dashboard Studio dashboards

The warning modals for Dashboard Studio dashboards differ in how they handle external or redirection content. Both modals have configurable feature settings that default to true for enablement.

External content warning modal

Dashboard Studio dashboards that attempt to load external content not listed in the Trusted Domains List receive an error message and the content doesn't load.

To avoid the error, you can do one of the following:

- Add the domain or URL to the Dashboards Trusted Domains List.
- Upload external content to your app directory and reference the content locally.
- Upload images directly with the Dashboard Studio UI. For more details, see Add an image.

Redirection content warning modal

Dashboard Studio dashboards that attempt to redirect to external content not listed in the Trusted Domains List receives a warning message confirming that you want to leave the Splunk Platform.

To avoid the warning modal, you can add the domain or URL to the Dashboards Trusted Domains List.

Classic SimpleXML dashboards

The warning modals for Classic SimpleXML dashboards differ in how they handle external or redirection content. Both modals have configurable feature settings that default to true for enablement.

External content warning modal

When viewing SimpleXML dashboards that attempt to load external content, a warning modal prompts the following:

- Load content by acknowledging the external domain or URL is trusted.
- Not load content by selecting **Cancel** because the external domain or URL is not trusted.

To avoid the warning modal, you can do one of the following:

- Add the domain or URL to the Dashboards Trusted Domains List.
- Upload external content to your app directory and reference the content locally.

Tags that load external content

The warning modal checks HTML tags that load external content. The following is a list of HTML tags in SimpleXML that load external content:

- applet
- audio
- base
- embed
- form
- frame
- iframe
- img
- object
- script
- style
- track
- video

Redirection content warning modal

The redirection content warning modal applies to any links in HTML tags or custom URLs. When viewing Classic SimpleXML dashboards that attempt to redirect to external content, a warning modal prompts the following:

- Redirect to the content by acknowledging the external domain or URL is trusted.
- Not redirect to the content by selecting Cancel because the external domain or URL is not trusted.

Tags that load external content

The warning modal checks HTML tags that redirect to external content. The following is a list of HTML tags in SimpleXML that redirect to external content:

- a
- link

Customize Splunk Web messages

You can modify notifications that display in Splunk Web in one of two ways:

- You can add and edit the text of custom notifications that display in the **Messages** menu.
- You can set the audience for certain error or warning messages generated by Splunk Enterprise.

Add or edit a custom notification

Add or edit a custom notification in Splunk Web or using the Splunk platform REST API.

Add a custom notification in Splunk Web

You can add a custom message to Splunk Web, for example to notify your users of scheduled maintenance. You need admin or system user level privileges to add or edit a custom notification.

To add or change a custom notification:

1. Select **Settings > User Interface**.
2. Click **New** to create a new message, or click **Bulletin Messages** and select the message you want to edit.
3. Give your new message a name and message text, or edit the existing text.
4. Click **Save.** The message will now appear when the user accesses **Messages** in the menu.

Add a custom notification using the Splunk platform REST API

For information on how to add a custom notification using the Splunk platform REST API, see Message users in apps for Splunk Cloud Platform and Splunk Enterprise in the *Splunk Developer Guide*.

Set audience for a Splunk Enterprise message

For some messages that appear in Splunk Web, you can control which users see the message.

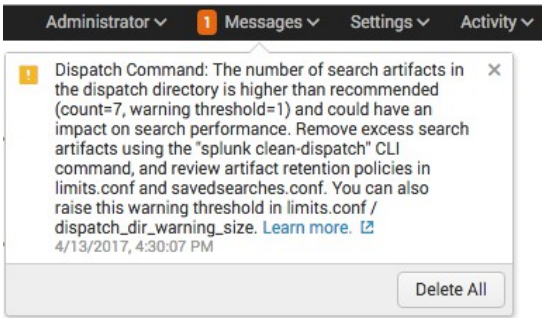
If by default a message displays only for users with a particular capability, such as `admin_all_objects`, you can display the message to more of your users, without granting them the `admin_all_objects` capability. Or you can have fewer users see a message.

The message you configure must exist in `messages.conf`. You can set the audience for a message by role or by capability, by modifying settings in `messages.conf`.

Identify a message available for audience scoping

The message you restrict must exist in `messages.conf`. Not all messages reside in `messages.conf`. If a message contains a Learn more link it resides in `messages.conf` and is configurable. If a message does not contain a Learn more link, it might or might not reside in `messages.conf` and be configurable.

For example, the message in the following image contains a Learn more link:



Once you have chosen a message that you want to configure, check whether it is configurable. Search for parts of the message string in `$SPLUNK_HOME/etc/system/default/messages.conf` on *nix or `%SPLUNK_HOME%\etc\system\default\messages.conf` on Windows. The message string is a setting within a stanza. The stanza name is a message identifier. Make note of the stanza name to use in your customized copy of `messages.conf`. Never edit the configuration files that are in the `default` directory.

For example, searching the default messages.conf for text from the sample message shown above, such as "artifacts," leads you to the following stanza:

```
[DISPATCHCOMM:TOO_MANY_JOB_DIRS__LU_LU]
message      = The number of search artifacts in the dispatch directory is higher than recommended
(count=%lu, warning threshold=%lu) and could have an impact on search performance.
action       = Remove excess search artifacts using the "splunk clean-dispatch" CLI command, and review
artifact retention policies in limits.conf and savedsearches.conf. You can also raise this warning threshold
in limits.conf / dispatch_dir_warning_size.
severity     = warn
capabilities = admin_all_objects
help         = message.dispatch.artifacts
```

The stanza name for this message is `DISPATCHCOMM:TOO_MANY_JOB_DIRS__LU_LU`.

Scope a message by capability

Set the capabilities required to view a message by editing the `capabilities` attribute in the `messages.conf` stanza for the message. A user must have all the listed capabilities to view the message.

For example,

```
[DISPATCHCOMM:TOO_MANY_JOB_DIRS__LU_LU]
capabilities = admin_all_objects, can_delete
```

For a list of capabilities and their definitions, see About defining roles with capabilities in *Securing Splunk Enterprise*.

If a role attribute is set for the message, that attribute takes precedence over the capabilities attribute. The capabilities attribute for the message is ignored.

See messages.conf.spec.

Scope a message by role

Set the roles required to view a message by editing the `roles` attribute in the `messages.conf` stanza for the message. If a user belongs to any of these roles, the message is visible to them.

If a role attribute is set for the message, that attribute takes precedence over the capabilities attribute. The capabilities attribute for the message is ignored.

For example:

```
[DISPATCHCOMM:TOO_MANY_JOB_DIRS__LU_LU]
roles = admin
```

See About configuring role-based user access in *Securing Splunk Enterprise*.

Display global banner

Splunk Enterprise lets you display a global banner that remains visible to all users on all UI pages across the product. The global banner feature gives organizations with strict security concerns the ability to display a site classification message that is required to run Splunk software in some environments. For example, you can display a global banner that tells users they are using a secure or classified site.

While the primary use case for the global banner is the persistent display of a site classification message, you can use it to display any type of notification that requires a persistent, highly visible message. For example, you can use the global banner to notify users about:

- New features
- Software version upgrades
- Scheduled maintenance or downtime
- Data outages

Splunk Web bulletin messages are suitable for most in-product notifications that do not require a persistent global message. For more information on bulletin messages, see Customize Splunk Web messages.

Splunk Enterprise supports the display of a single global banner only. The global banner does not appear on the Splunk Enterprise login page and users cannot dismiss the banner inside the product.

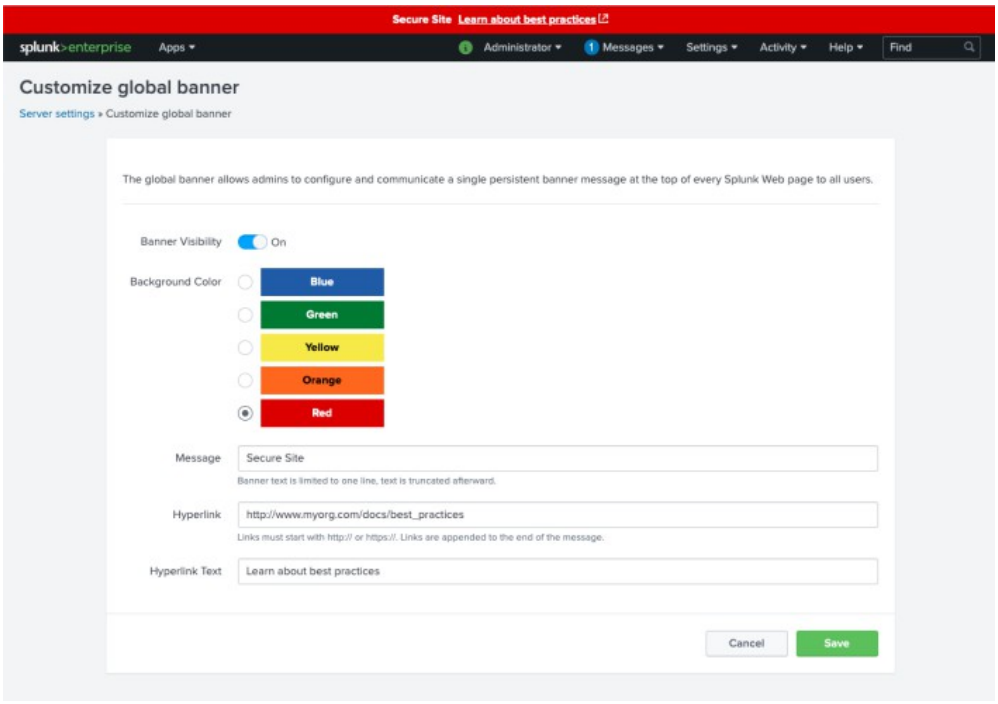
Customize the global banner

You can enable and customize the global banner using Splunk Web, REST, or configuration files.

To customize the global banner a role must have the edit_global_banner capability. This capability is provided to admin and sc_admin roles by default.

Customize global banner using Splunk Web

1. In Splunk Web, click **Settings > Server Settings > Global Banner**.
2. Toggle the **Banner Visibility** switch to On.
3. Select a background color for your global banner.
4. Enter your message text.
5. (optional) Specify a URL to generate a hyperlink to additional information, such as relevant best practices documentation.
6. Enter text for the hyperlink. For example, "Learn about best practices".
   Your customized global banner now appears on all UI pages in Splunk Enterprise.



Deploy global banner in a search head cluster

In a search head cluster environment, some sections of the Settings menu in Splunk Web are hidden by default. To deploy the global banner using Splunk Web in a search head cluster, you must first show the full Settings menu, as follows:

1. On any cluster member, in Splunk Web, click **Settings > Show All Settings > Show**.
   The full Settings menu now appears in Splunk Web.
2. Click **Server settings > Global banner**.
3. Customize the global banner as shown in the preceding section Customize global banner using Splunk Web.
4. Click **Save**.
   The search head cluster automatically replicates the global banner configuration to each cluster member and the global banner now appears in Splunk Web on each search head.

Customize global banner using REST

To customize the global banner using REST, send a POST request to the following endpoint:

`data/ui/global-banner`
For endpoint details, see data/ui/global-banner in the *REST API Reference Manual*.

Customize global banner using configuration files

You can customize the global banner by specifying settings in `$SPLUNK_HOME/etc/system/local/global-banner.conf`.

For detailed information on `global-banner.conf` settings, see global-banner.conf.

Administer Splunk Enterprise with configuration files

About configuration files

Splunk Enterprise configuration settings are stored in **configuration files**. These files are identified by the `.conf` extension. Types of configuration settings include:

- System settings
- Authentication and authorization information
- Index-related settings
- Deployment and cluster configurations
- Knowledge objects and saved searches

For a list of configuration files and an overview of the area that each file covers, see List of configuration files in this manual.

Default configuration files are stored in the `$SPLUNK_HOME/etc/system/default/` directory.

Use Splunk Web to manage configuration files

When you change your configuration in Splunk Web, that change is written to a copy of the configuration file for that setting. Splunk software creates a copy of this configuration file (if it does not exist), writes the change to that copy, and adds it to a directory under `$SPLUNK_HOME/etc/...`. The directory that the new file is added to depends on a number of factors that are discussed in Configuration file directories in this manual. The most common directory is `$SPLUNK_HOME/etc/system/local`, which is used in the example.

If you add a new index in Splunk Web, the software performs the following actions:

1. Checks for a copy of the file.

2. If no copy exists, the software creates a copy of `indexes.conf` and adds it to a directory, such as `$SPLUNK_HOME/etc/system/local`.

3. Writes the change to the copy of `indexes.conf`.

4. Leaves the default file unchanged in `$SPLUNK_HOME/etc/system/default`.

Edit the configuration file settings directly

While you can perform a lot of configuration with Splunk Web or CLI commands, you can also edit the configuration files directly. Some advanced configurations are not exposed in Splunk Web or the CLI and can only be changed by editing the configuration files directly.

Never change, copy, or move the configuration files that are in the default directory. Default files must remain intact and in their original location. When you upgrade your Splunk software, the default directory is overwritten. Any changes that you make in the default directory are lost when you upgrade to a newer version of the software. Changes that you make in non-default configuration directories persist when you upgrade.

To change settings for a particular configuration file, you must first create a new version of the file in a non-default directory and then add the settings that you want to change. When you first create this new version of the file, start with an empty file. Do not start from a copy of the file in the default directory. For information on the directories where you can manually change configuration files, see Configuration file directories.

Before you change any configuration files:

- Learn about how the default configuration files work, and where to put the files that you edit. See Configuration file directories.
- Learn about the structure of the stanzas that comprise configuration files and how the attributes you want to edit are set up. See Configuration file structure.
- Learn how different versions of the same configuration files in different directories are layered and combined so that you know the best place to put your file. See Configuration file precedence.
- Consult the product documentation, including the .spec and .example files for the configuration file. These documentation files reside in the file system in `$SPLUNK_HOME/etc/system/README`, as well as in the last chapter of this manual.

After you are familiar with the configuration file content and directory structure, and understand how to leverage Splunk Enterprise configuration file precedence, see How to edit a configuration file to learn how to safely change your files.

Configuration file directories

A Splunk Enterprise installation can have multiple versions of a configuration file located across several directories. For example, you might have the same configuration file with different settings located in each of the default, local, and app directories. Splunk Enterprise uses a layering scheme and rules to evaluate overlapping configurations and prioritize them.

When you need to override a setting that's been defined as a default, you can place a customized configuration file in a different folder path under the Splunk Enterprise installation. For a description and examples of how precedence is determined, see Configuration file precedence.

A detailed list of settings for each configuration file is provided in the `.spec` file named for that configuration file. You can find the latest version of the `.spec` and `.example` files in the `$SPLUNK_HOME/etc/system/README` folder of your Splunk Enterprise installation, or in the documentation at the configuration file reference.

About the default files

The default directory contains preconfigured versions of the configuration files with default settings. The location of the default directory in a Splunk Enterprise installation is `$SPLUNK_HOME/etc/system/default`.

"all these worlds are yours, except /default - attempt no editing there" -- duckfez, 2010

You should never change a configuration file that's located in the `$SPLUNK_HOME/etc/system/default` directory. The Splunk Enterprise upgrade process overwrites the contents in that folder automatically, which will remove any changes. If you want to retain a setting you've changed through an upgrade, place your configuration file into a `local` folder path such as `$SPLUNK_HOME/etc/system/local` or `$SPLUNK_HOME/etc/apps/$app_name/local` as described below.

The upgrade process also inspects the content in the $SPLUNK_HOME/etc/system/local folder path. An upgrade usually does not make changes to the local configuration files, but if changes are made they are noted in the

configuration file or in the migration log. You can choose to preview the changes to your customized configuration files as part of the upgrade process before any changes are made.

Where you can place (or find) your modified configuration files

To change the settings in a particular configuration file, you must first create a new file of the same name in a non-default directory, and add the required settings and changed values to your new configuration file. A setting with a new value defined in a non-default directory will take precedence over a setting defined in the default directory.

When changing a default setting using a new configuration file, you only need to define the stanza category, the setting, and update the value. Do not make a complete copy of the configuration file from the default directory into another folder, as the settings in that copy will take precedence and override changes made during an upgrade.

The following is the configuration directory structure in `$SPLUNK_HOME/etc`:

$SPLUNK_HOME/etc/system/local

Local changes on a site-wide basis go here; for example, settings you want to make available to all apps. If the configuration file you're looking for doesn't already exist in this directory, create it and verify the service account has permissions to it.

$SPLUNK_HOME/etc/peer-apps/[_cluster|<app_name>]/[local|default]

For indexer cluster peer nodes only.

The subdirectories under `$SPLUNK_HOME/etc/peer-apps` contain configuration files that are common across all peer nodes.

DO NOT change the content of these subdirectories on the cluster peer itself. Instead, use the cluster manager node to distribute any new or modified files to them.

The `_cluster` directory contains configuration files that are not part of real apps but that still need to be identical across all peers. A typical example is the `indexes.conf` file.

For more information, see Update common peer configurations in the *Managing Indexers and Clusters* manual.

$SPLUNK_HOME/etc/apps/<app_name>/[local|default]

If you're in an app when a configuration change is made, the setting goes into a configuration file in the app's `/local` directory. For example, edits for search-time settings in the Search app go here: `$SPLUNK_HOME/etc/apps/search/local/`.

If you want to edit a configuration file so that the change only applies to a certain app, copy the file to the app's `/local` directory, verify permissions, and make your changes there.

$SPLUNK_HOME/etc/users

User-specific configuration changes go here.

$SPLUNK_HOME/etc/system/README

This directory contains supporting reference documentation. For most configuration files, there are two reference files: `.spec` and `.example`; for example, `inputs.conf.spec` and `inputs.conf.example`. The `.spec` file specifies the syntax, including a list of available attributes and variables. The `.example` file contains examples of real-world usage.

Configuration file structure

Before you edit configuration files, you should familiarize yourself with the structure of the files.

Stanzas

Configuration files consist of one or more **stanzas**, or sections. Each stanza begins with a stanza header in square brackets. This header identifies the settings held within that stanza. Each setting is an attribute value pair that specifies particular configuration settings.

For example, `inputs.conf` provides an `[SSL]` that includes settings for the server certificate and password (among other things):

```
[SSL]
serverCert = <pathname>
password = <password>
```

Depending on the stanza type, some of the attributes might be required, while others could be optional.

Setting up a new stanza

When you edit a configuration file, you might be changing the default stanza, like above, or you might need to add a brand-new stanza.

Here's the basic pattern:

```
[stanza1_header]
<attribute1> = <val1>
# comment
<attribute2> = <val2>
...

[stanza2_header]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

Important: Attributes are case-sensitive. For example, `sourcetype = my_app` is **not** the same as `SOURCETYPE = my_app`. One will work; the other won't.

Stanza scope

Configuration files frequently have stanzas with varying scopes, with the more specific stanzas taking precedence. For example, consider this example of an `outputs.conf` configuration file, used to configure **forwarders**:

```
[tcpout]
indexAndForward=true
```

```
compressed=true

[tcpout:my_indexersA]
compressed=false
server=mysplunk_indexer1:9997, mysplunk_indexer2:9997

[tcpout:my_indexersB]
server=mysplunk_indexer3:9997, mysplunk_indexer4:9997
```

Note that this example file has two levels of stanzas:

- The global `[tcpout]`, with settings that affect all tcp forwarding.
- Two `[tcpout:<target_list>]` stanzas, whose settings affect only the indexers defined in each target group.

The setting for `compressed` in `[tcpout:my_indexersA]` overrides that attribute's setting in `[tcpout]`, *for the indexers in the my_indexersA target group only.*

For more information on forwarders and `outputs.conf`, see Configure forwarders with outputs.conf.

Configuration file precedence

Splunk software uses **configuration files** to determine nearly every aspect of its behavior. A Splunk platform deployment can have many copies of the same configuration file. These file copies are usually layered in directories that affect either the users, an **app**, or the system as a whole.

When editing configuration files, it is important to understand how Splunk software evaluates these files and which ones take precedence.

When incorporating changes, Splunk software does the following to your configuration files:

- It merges the settings from all copies of the file, using a location-based prioritization scheme.
- When different copies have conflicting attribute values (that is, when they set the same attribute to different values), it uses the value from the file with the highest priority.
- It determines the priority of configuration files by their location in the directory structure, according to the rules described in this topic.

Note: Besides resolving configuration settings among multiple copies of a file, Splunk software sometimes needs to resolve settings within a single file. See Attribute precedence within a single props.conf file.

About configuration file context

To determine the order of directories for evaluating configuration file precedence, Splunk software considers each file's context. Configuration files operate in either a global context or in the context of the current app and user:

- **Global.** Activities like indexing take place in a global context. They are independent of any app or user. For example, configuration files that determine monitoring or indexing behavior occur outside of the app and user context and are global in nature.
- **App/user.** Some activities, like searching, take place in an app or user context. The app and user context is vital to search-time processing, where certain knowledge objects or actions might be valid only for specific users in

specific apps.

The precedence order for configuration file directories varies according to the context of the particular configuration file. To learn the context of each file, see List of configuration files and their context.

How Splunk determines precedence order

Configuration file precedence order depends on the location of file copies within the directory structure. Splunk software considers the context of each file to determine the precedence order of the directories.

Precedence within global context

When the file context is global, directory priority descends in this order:

1. System local directory -- highest priority
2. App local directories
3. App default directories
4. System default directory -- lowest priority

When consuming a global configuration, such as `inputs.conf`, Splunk software first uses the attributes from any copy of the file in `system/local`. Then it looks for any copies of the file located in the app directories, adding any attributes found in them, but ignoring attributes already discovered in system/local. As a last resort, for any attributes not explicitly assigned at either the system or app level, it assigns default values from the file in the `system/default` directory.

Note: As the next section describes, cluster peer nodes have an expanded order of precedence.

Precedence within global context, indexer cluster peers only

There is an expanded precedence order for indexer cluster peer configurations, which are considered in the global context. This is because some configuration files, like `indexes.conf`, must be identical across peer nodes.

To keep configuration settings consistent across peer nodes, configuration files are managed from the cluster manager node, which pushes the files to the `peer-app` directories on the peer nodes. Files in the `peer-app` directories have the highest precedence in a cluster peer's configuration. These directories exist only on indexer cluster peer nodes.

Here is the expanded precedence order for cluster peers:

1. Peer-app local directories -- highest priority
2. System local directory
3. App local directories
4. Peer-app default directories
5. App default directories
6. System default directory -- lowest priority

Precedence within app or user context

For files with an app/user context, directory priority descends from user to app to system:

1. User directories for current user -- highest priority
2. App directories for currently running app (local, followed by default)
3. App directories for all other apps (local, followed by default) -- for exported settings only

4. System directories (local, followed by default) -- lowest priority

An attribute in `savedsearches.conf`, for example, might be set at all three levels: the user, the app, and the system. Splunk will always use the value of the user-level attribute, if any, in preference to a value for that same attribute set at the app or system level.

How app directory names affect precedence

> For most practical purposes, the information in this subsection probably won't matter, but it might prove useful if you need to force a certain order of evaluation or for troubleshooting.

The effect of app directory names varies depending on whether the context is global or local.

App directory names in the global context

When determining priority in the global context, Splunk software uses lexicographical order to determine priority among the collection of apps directories. For example, files in an apps directory named "A" have a higher priority than files in an apps directory named "B", and so on.

App directory names in the app/user context

When determining priority in the app/user context, Splunk software uses reverse-lexicographical order to determine priority among the collection of apps directories, For example, files in an apps directory named "B" have a higher priority than files in an apps directory named "A", and so on.

> When determining precedence in the app/user context, directories for the currently running app take priority over those for all other apps, independent of how they're named. Furthermore, other apps are only examined for exported settings.

The finer points of lexicographical order

In the global context only, lexicographical order determines precedence. Thus, files in an apps directory named "A" have a higher priority than files in an apps directory named "B", and so on. Also, all apps starting with an uppercase letter have precedence over any apps starting with a lowercase letter, due to lexicographical order. ("A" has precedence over "Z", but "Z" has precedence over "a", for example.)

In addition, numbered directories have a higher priority than alphabetical directories and are evaluated in lexicographic, not numerical, order. For example, in descending order of precedence:

```
$SPLUNK_HOME/etc/apps/myapp1
$SPLUNK_HOME/etc/apps/myapp10
$SPLUNK_HOME/etc/apps/myapp2
$SPLUNK_HOME/etc/apps/myapp20
...
$SPLUNK_HOME/etc/apps/myappApple
$SPLUNK_HOME/etc/apps/myappBanana
$SPLUNK_HOME/etc/apps/myappZabaglione
...
$SPLUNK_HOME/etc/apps/myappapple
$SPLUNK_HOME/etc/apps/myappbanana
$SPLUNK_HOME/etc/apps/myappzabaglione
...
```

Lexicographical order sorts items based on the values used to encode the items in computer memory. In Splunk software, this is almost always UTF-8 encoding, which is a superset of ASCII.

- Numbers are sorted before letters. Numbers are sorted based on the first digit. For example, the numbers 10, 9, 70, 100 are sorted lexicographically as 10, 100, 70, 9.
- Uppercase letters are sorted before lowercase letters.
- Symbols are not standard. Some symbols are sorted before numeric values. Other symbols are sorted before or after letters.

In the app/user context, precedence is determined instead by reverse-lexicographical order. Therefore, the order of precedence is exactly opposite the lexicographical order described above, which is used in the global context only. For example, files in an apps directory named "B" have a higher priority than files in an apps directory named "A", files in app "a" have precedence over files in apps "B" or "A", and so on. Similarly, numerical app directories have a lower precedence than alphabetical directories.

Summary of the effect of directories on configuration precedence

Putting this all together, the order of directory priority, from highest to lowest, goes like this:

Global context

```
$SPLUNK_HOME/etc/system/local/*

$SPLUNK_HOME/etc/apps/A/local/* ... $SPLUNK_HOME/etc/apps/z/local/*

$SPLUNK_HOME/etc/apps/A/default/* ... $SPLUNK_HOME/etc/apps/z/default/*

$SPLUNK_HOME/etc/system/default/*
```

Global context, cluster peer nodes only

```
$SPLUNK_HOME/etc/peer-apps/A/local/* ... $SPLUNK_HOME/etc/peer-apps/z/local/*

$SPLUNK_HOME/etc/system/local/*

$SPLUNK_HOME/etc/apps/A/local/* ... $SPLUNK_HOME/etc/apps/z/local/*

$SPLUNK_HOME/etc/peer-apps/A/default/* ... $SPLUNK_HOME/etc/peer-apps/z/default/*

$SPLUNK_HOME/etc/apps/A/default/* ... $SPLUNK_HOME/etc/apps/z/default/*

$SPLUNK_HOME/etc/system/default/*
```

Within the peer-apps/[local|default] directories, the special _cluster subdirectory has a higher precedence than any app subdirectories starting with a lowercase letter (for example, anApp). However, it has a lower precedence than any apps starting with an uppercase letter (for example, AnApp). This is due to the location of the underscore ("_") character in the lexicographical order.

App/user context

```
$SPLUNK_HOME/etc/users/*
```

```
$SPLUNK_HOME/etc/apps/Current_running_app/local/*

$SPLUNK_HOME/etc/apps/Current_running_app/default/*

$SPLUNK_HOME/etc/apps/z/local/*, $SPLUNK_HOME/etc/apps/z/default/*, ... $SPLUNK_HOME/etc/apps/A/local/*,
$SPLUNK_HOME/etc/apps/A/default/*

$SPLUNK_HOME/etc/system/local/*

$SPLUNK_HOME/etc/system/default/*
```

In the app/user context, all configuration files for the currently running app take priority over files from all other apps. This is true for both the app's local and default directories. So, if the current context is app C, Splunk evaluates both $SPLUNK_HOME/etc/apps/C/local/* and $SPLUNK_HOME/etc/apps/C/default/* before evaluating the local and default directories for any other apps. Furthermore, Splunk software only looks at configuration data for other apps if that data has been exported globally through the app's default.meta file. Also note that /etc/users/ is evaluated only when the particular user logs in or performs a search.

Example of how attribute precedence works

This example of attribute precedence uses `props.conf`. The `props.conf` file is unusual, because its context can be either global or app/user, depending on when Splunk is evaluating it. Splunk evaluates `props.conf` at both index time (global) and search time (apps/user).

Assume `$SPLUNK_HOME/etc/system/local/props.conf` contains this stanza:

```
[source::/opt/Locke/Logs/error*]
sourcetype = fatal-error
```

and `$SPLUNK_HOME/etc/apps/t2rss/local/props.conf` contains another version of the same stanza:

```
[source::/opt/Locke/Logs/error*]
sourcetype = t2rss-error
SHOULD_LINEMERGE = True
BREAK_ONLY_BEFORE_DATE = True
```

The line merging attribute assignments in `t2rss` always apply, as they only occur in that version of the file. However, there's a conflict with the `sourcetype` attribute. In the `/system/local` version, the `sourcetype` has a value of "fatal-error". In the `/apps/t2rss/local` version, it has a value of "t2rss-error".

Since this is a `sourcetype` assignment, which gets applied at index time, Splunk uses the global context for determining directory precedence. In the global context, Splunk gives highest priority to attribute assignments in `system/local`. Thus, the `sourcetype` attribute gets assigned a value of "fatal-error".

The final, internally merged version of the file looks like this:

```
[source::/opt/Locke/Logs/error*]
sourcetype = fatal-error
SHOULD_LINEMERGE = True
BREAK_ONLY_BEFORE_DATE = True
```

List of configuration files and their context

As mentioned, Splunk decides how to evaluate a configuration file based on the context that the file operates within, global or app/user. Generally speaking, files that affect data input, indexing, or deployment activities are global; files that affect search activities usually have a app/user context.

The `props.conf` and `transforms.conf` files can be evaluated in either a app/user or a global context, depending on whether Splunk is using them at index or search time. The `limits.conf` file is evaluated in a global context except for a few settings, which are tunable by app or user.

The following sections describe the global or app/user status of some important configuration files.

Global configuration files

```
authentication.conf
authorize.conf
deploymentclient.conf
distsearch.conf
indexes.conf
inputs.conf
limits.conf, except for indexed_realtime_use_by_default
outputs.conf
procmonfilters.conf
props.conf -- global and app/user context
pubsub.conf
restmap.conf
searchbnf.conf
segmenters.conf
server.conf
serverclass.conf
serverclass.seed.xml.conf
source-classifier.conf
sourcetypes.conf
transforms.conf  -- global and app/user context
user-seed.conf -- special case: Must be located in /system/default
web.conf
wmi.conf
```

App/user configuration files

```
alert_actions.conf
app.conf
audit.conf
commands.conf
eventdiscoverer.conf
event_renderers.conf
eventtypes.conf
fields.conf
literals.conf
macros.conf
multikv.conf
props.conf -- global and app/user context
savedsearches.conf
tags.conf
times.conf
transactiontypes.conf
```

```
transforms.conf  -- global and app/user context
user-prefs.conf
workflow_actions.conf
```

Troubleshooting configuration precedence and other issues

Splunk's configuration file system supports many overlapping configuration files in many different locations. The price of this level of flexibility is that figuring out which value for which configuration option is being used in your Splunk installation can sometimes be quite complex. If you're looking for some tips on figuring out what configuration setting is being used in a given situation, read Use btool to troubleshoot configurations in the *Troubleshooting Manual*.

Attribute precedence within a single props.conf file

In addition to understanding how attribute precedence works across files, you also sometimes need to consider attribute priority within a single props.conf file.

Precedence within sets of stanzas affecting the same target

When two or more **stanzas** specify a behavior that affects the same item, items are evaluated by the stanzas' ASCII order. For example, assume you specify in `props.conf` the following stanzas:

```
[source::.../bar/baz]
attr = val1

[source::.../bar/*]
attr = val2
```

The second stanza's value for `attr` will be used, because its path is higher in the ASCII order and takes precedence.

Overriding default attribute priority in props.conf

There's a way to override the default ASCII priority in `props.conf`. Use the `priority` key to specify a higher or lower priority for a given stanza.

For example, suppose we have a source:

```
source::az
```

and the following patterns:

```
[source::...a...]
sourcetype = a

[source::...z...]
sourcetype = z
```

In this case, the default behavior is that the settings provided by the pattern "source::...a..." take precedence over those provided by "source::...z...". Thus, sourcetype will have the value "a".

To override this default ASCII ordering, use the `priority` key:

```
[source::...a...]
sourcetype = a
priority = 5

[source::...z...]
sourcetype = z
priority = 10
```

Assigning a higher priority to the second stanza causes `sourcetype` to have the value "z".

There's another attribute precedence issue to consider. By default, stanzas that match a string literally ("literal-matching stanzas") take precedence over regex pattern-matching stanzas. This is due to the default values of their `priority` keys:

- 0 is the default for pattern-matching stanzas
- 100 is the default for literal-matching stanzas

So, literal-matching stanzas will always take precedence over pattern-matching stanzas, unless you change that behavior by explicitly setting their `priority` keys.

You can use the `priority` key to resolve collisions between patterns of the same type, such as `sourcetype` patterns or `host` patterns. The `priority` key does not, however, affect precedence across spec types. For example, `source` patterns take priority over `host` and `sourcetype` patterns, regardless of priority key values.

Precedence for events with multiple attribute assignments

The `props.conf` file sets attributes for processing individual events by host, source, or sourcetype (and sometimes event type). So it's possible for one event to have the same attribute set differently for the **default fields**: host, source or sourcetype. The precedence order is:

- source
- host
- sourcetype

You might want to override the default `props.conf` settings. For example, assume you are tailing `mylogfile.xml`, which by default is labeled `sourcetype = xml_file`. This configuration will re-index the entire file whenever it changes, even if you manually specify another sourcetype, because the property is set by source. To override this, add the explicit configuration by source:

```
[source::/var/log/mylogfile.xml]
CHECK_METHOD = endpoint_md5
```

How to edit a configuration file

To customize a Splunk platform instance to meet your specific needs, you can edit the built-in configuration settings.

Prerequisites

You must meet the following prerequisites to edit configuration files:

- You must be a user with file system access, such as a system administrator.

- You must understand how the configuration system works across your deployment and where to make the changes.

The following table describes what you need to know and where to find that information:

What to know	Documentation
You can have configuration files with the same name in your default, local, and app directories. This layering effect that allows your Splunk platform deployment to determine configuration priorities. Before you edit a configuration file, you need to know where to create the custom version of the configuration file.	See Configuration file directories.
Configuration files consist of stanzas. Each stanza identifies settings that specify the configuration. Before you edit a configuration file, you need to understand how the file's stanzas are structured.	See Configuration file structure.
Splunk software uses configuration files to set defaults and limitations. A Splunk platform deployment can have multiple copies of the same configuration file in different directories. The ways these copies are layered in the directories affect either the user, an app, or the system as a whole. When you are editing a configuration file, you need to understand how Splunk software evaluates files in order of importance.	See Configuration file precedence.
If your deployment includes search head or indexer clusters, some changes you make to configuration files may need to be made to configuration files on every search head cluster member or indexer peer node. Before you edit a configuration file, you need to understand whether to make the same change to the configuration file on every search head or indexer in the cluster.	See Use the deployer to distribute apps and configuration updates in *Distributed Search* and Update common peer configurations and apps in *Managing Indexers and Clusters of Indexers*.

Customize a configuration file

To customize a configuration file, create a new file with the same name in a local or app directory. Then, add the specific settings that you want to customize to the local configuration file.

Never change or copy the configuration files in the default directory. The files in the default directory must remain intact and in their original location. The Splunk Enterprise upgrade process overwrites the default directory. Any changes that you make in the default directory are lost on upgrade. Changes that you make in non-default configuration directories, such as $SPLUNK_HOME/etc/system/local or $SPLUNK_HOME/etc/apps/<app_name>/local, persist through upgrades.

1. Determine whether the configuration file already exists in your preferred directory. For example, if you want to make changes to a configuration file in your local directory, open the $SPLUNK_HOME/etc/system/local directory.
2. If the configuration file does not exist in your preferred directory, create the file. You are creating an empty file.

3. Edit the configuration file in the preferred directory and add only the stanzas and settings that you want to customize in the local file.

Clear a setting

You can clear a setting to override any previous value that the setting held, including the value set in the default directory. Clearing a setting causes the system to consider the value entirely unset.

You clear a setting by changing its value to null.

For example, suppose you want to clear the `forwardedindex.0.whitelist` setting in the output.conf file that is in your local directory. Follow these steps to clear the setting:

1. Open the outputs.conf file in your local directory.
2. Find the `forwardedindex.0.whitelist` setting and change the value to null. For example:

```
forwardedindex.0.whitelist =
```

3. Save the outputs.conf file.

Because the settings in the local directory take precedence over the settings in the default directory, when Splunk software reads the settings, the null setting for `forwardedindex.0.whitelist` is used.

Insert a comment

When you customize a setting, it is useful to explain why you customized the setting. Adding comments to configuration files in your local or apps directory is a great way to add these explanations, both for you and for others who view these files.

To add a comment to a configuration file, insert the pound sign (#) before the comment. Start the comment at the beginning of a line.

The best location to put your comment is either before the stanza that setting is within or before the setting itself. For example:

```
# This stanza forwards some log files.
[monitor:///var/log]
```

If you have multiple settings in a stanza, then add the comments before each setting. Consider including a date in your comment or placing your comments in all capital letters. For example:

```
[stanza_name]

# 1/30/2020 - 5 is optimal for our current configuration.
# This was discussed with both Alex Garcia and Wei Zhang.
a_setting = 5

# 9/15/2019 - WE'VE CHANGED THIS SETTING TO "TRUE" BECAUSE IT ALLOWS US TO  <your_reason_goes_here>.
b_setting = true
```

Where not to put your comments

> Do not put comments on the same line as the stanza or the setting.

This example shows where not to place your comments.

```
[monitor:///var/log]    # This is a bad place to put your comment.
a_setting = 5  # This is a bad place too.
```

Placing comments on the same line as a stanza or setting might cause unexpected results. In the following example, the comment is placed on the same line as the setting:

```
a_setting = 5  #5 is the best number
```

This sets `a_setting` to the value `5 #5 is the best number` and not to `5` as intended.

Creating and editing configuration files on Windows and other non-UTF-8 operating systems

The Splunk platform works with configuration files with ASCII/UTF-8 encoding.

On operating systems where UTF-8 is not the default character set, such as Windows, configure your text editor to write files in the default character set for that operating system.

Best practices for editing configuration files

When editing your configuration files, follow these best practices:

Best practice	Description
Spell the name of the setting correctly.	For best results, copy the setting name from the corresponding specification file.
Verify the capitalization is correct in the names of settings.	Setting names are case-sensitive. That is, for a setting named `someAttribute`, you cannot substitute `SomeAttribute` or `someattribute`.
Place the setting in the stanza where it belongs. Many settings only operate within a particular context and require you to place them within the stanza for that context.	Refer to the specification files for guidance on stanza requirements. As an example, indexer clustering settings in the server.conf file must be placed within the `[clustering]` stanza.
Place the setting so that it applies to the desired scope. Some settings can be applied either globally or within a specific scope.	To apply the setting globally, place the setting towards the start of the configuration file, prior to any stanza. When the setting has a specific scope, place the setting within the stanza for that scope. For example, in the indexes.conf file, some settings can be applied either on a per-index basis or globally, for all indexes. If you want a particular value for the setting to apply to just a single index, place the setting under that index's stanza. Similarly, if you want a setting to apply to all indexes, place the setting above all stanzas. You can also place a setting with one value above the stanzas and then add the setting with a different value to one or more index stanzas. That way, each index uses the global value except where the setting's value has been modified for a specific index.

Best practice	Description

Do not add the same setting twice within the same context. If you do, the final instance of the setting will take effect.	If you add the same setting twice within the same context, you might find yourself confused at some later date.
	
	For example, say you add this stanza and setting:
	
	`[some stanza]`
	`setting=foo`
	Then, someone later adds a stanza with the same name but a different value for the setting further down in the file:
	
	`[some stanza]`
	`setting=bar`
	The setting now has a value of `bar`, because the second instance is further down in the file. However, this can cause confusion if someone later tries to change the setting and encounters the first instance of the setting but not the second.

When to restart Splunk Enterprise after a configuration file change

When you make changes to a Splunk Enterprise instance by manually editing the configuration files, you might need to restart Splunk Enterprise for the changes to take effect.

Note: Updates made through Splunk Web, REST API endpoints, or the CLI are less likely to require restarts. This is because the instance automatically reloads the changed configurations after such updates.

This topic provides guidelines to help you determine whether to restart after a change. Whether a change requires a restart depends on a number of factors, and this topic does not provide a definitive authority. Always check the configuration file or its reference topic to see whether a particular change requires a restart. For a full list of configuration files and an overview of the area each file covers, see List of configuration files in this manual.

When to restart forwarders

If you make a configuration file change to a heavy forwarder, you must restart the forwarder, but you do not need to restart the receiving indexer. If the changes are part of a deployed app already configured to restart after changes, then the forwarder restarts automatically.

When to restart splunkweb

You must restart splunkweb to enable or disable SSL for Splunk Web access.

When to restart splunkd

As a general rule, restart splunkd after making the following types of changes.

Indexer changes

- Index time field extractions
- Time stamp properties

For information on changes to `indexes.conf` settings that necessitate a restart, see Determine which indexes.conf changes require restart in *Managing Indexers and Clusters of Indexers*. In addition, for information on configuration bundle changes that initiate a restart, see Update common peer configurations and apps in *Managing Indexers and Clusters of Indexers*.

Note: When settings that affect indexing are changed through Splunk Web and the CLI, they do not require restarts and take place immediately.

User and role changes

Any user and role changes made in configuration files require a restart, including:

- LDAP configurations (If you make these changes in Splunk Web you can reload the changes without restarting.)
- Password changes
- Changes to role capabilities
- Changes to existing roles, such as settings for role-based field filters or search filters (see Manage an existing role in *Securing the Splunk Platform*).
- Splunk Enterprise native authentication changes, such as user-to-role mappings.

System changes

Changes that affect the system settings or server state require restart, such as:

- Licensing changes
- Web server configuration updates
- Turning on or off role-based field filtering
- Changes to general indexer settings (minimum free disk space, default server name, etc.)
- Changes to General settings (e.g., port settings).
- Changing a forwarder's output settings
- Changing the time zone in the OS of a Splunk Enterprise instance (Splunk Enterprise retrieves its local time zone from the underlying OS at startup)
- Installing some apps may require a restart. Consult the documentation for each app you are installing.

Splunk Enterprise changes that do not require a restart

Search-time processing settings

Settings that apply to search-time processing take effect immediately and do not require a restart. This is because searches run in a separate process that reloads configurations. For example, lookup tables, tags, and event types are re-read for each search.

This includes (but is not limited to) changes to:

- Lookup tables
- Field extractions
- Knowledge objects
- Tags
- Event types

Files that contain search-time operations include (but are not limited to):

- `macros.conf`

- `props.conf`
- `transforms.conf`
- `savedsearches.conf` (If a change creates an endpoint you must restart.)

To reload your endpoints type the following into your browser:

```
http://<yoursplunkserver>:8000/en-US/debug/refresh
```

Index-time settings

Index-time props and transforms do not require restarts, as long as your indexers are receiving the data from forwarders. That is to say:

- Changes to `props.conf` and `transforms.conf` on an indexer do not require restarts.
- In an indexer cluster, changes to `props.conf` and `transforms.conf` are automatically reloaded when the peers receive the changes from the manager node.
- On a non-clustered indexer, changes to `props.conf` and `transforms.conf` require a reload.
- On either a clustered or non-clustered indexer, once the .conf files have reloaded, the changes take effect after a forwarder auto-LB time period.

Workload management settings

Changes to the workload management configuration files workload_rules.conf and workload_pools.conf do not require a restart.

How to reload files

To reload `transforms.conf`:

```
http://<yoursplunkserver>:8000/en-US/debug/refresh?entity=admin/transforms-lookup
for new lookup file definitions that reside within transforms.conf

http://<yoursplunkserver>:8000/en-US/debug/refresh?entity=admin/transforms-extract
for new field transforms/extractions that reside within transforms.conf
```

To reload `authentication.conf`, use Splunk Web. Go to **Settings > Access controls > Authentication method** and click **Reload authentication configuration**. This refreshes the authentication caches, but does not disconnect current users.

Restart an indexer cluster

To learn about restarts in an indexer cluster, and when and how to use a rolling restart, see Restart the entire indexer cluster or a single peer node in *Managing Indexers and Clusters of Indexers*.

Use cases

In complex situations, restarting Splunk Enterprise is the safest practice. Here are a few scenarios where you might (or might not) be able to avoid a restart.

Scenario: You edit search- or index-time transforms in `props.conf` and `transforms.conf`

Whether to restart depends on whether the change is related to a index-time setting or a search-time setting. Index-time settings include:

- line breaking
- timestamp parsing

Search-time settings relate mainly to field extraction and creation and do not require a restart. Any index-time changes still require a restart. For example:

1. If `props.conf` and `transforms.conf` are configured as search-time transforms on the index, you do not have to restart. For search-time changes, each time you run a search, Splunk software reloads the `props.conf` and `transforms.conf`.

2. If the search-time changes are on a heavy forwarder, you must restart that forwarder. (If the changes are part of a deployed app configured to restart after changes, then this happens automatically.)

3. If it is an index-time transform on the indexer, you must restart the indexer.

List of configuration files

The following is a list of some of the available spec and example files associated with each conf file. Some conf files do not have spec or example files. Contact Support before editing a conf file that does not have an accompanying spec or example file.

Do not edit the default copy of any conf file in `$SPLUNK_HOME/etc/system/default/`. See How to edit a configuration file.

File	Purpose
alert_actions.conf	Create an alert.
app.conf	Configure app properties
audit.conf	Configure auditing and event hashing. This feature is not available for this release.
authentication.conf	Toggle between Splunk's built-in authentication or LDAP, and configure LDAP.
authorize.conf	Configure roles, including granular access controls.
bookmarks.conf	Bookmark monitoring console URLs.
checklist.conf	Customize monitoring console health check.
collections.conf	Configure KV Store collections for apps.
commands.conf	Create custom search commands for apps in Splunk Cloud Platform or Splunk Enterprise using in the Developer Guide on the Developer Portal.
datamodels.conf	Attribute/value pairs for configuring data models.
default.meta	Set permissions for objects in a Splunk app.
deploymentclient.conf	Specify behavior for clients of the deployment server.
distsearch.conf	Specify behavior for distributed search.
event_renderers.conf	Configure event-rendering properties.
eventtypes.conf	Create event type definitions.
federated.conf	Search data outside of your own Splunk platform deployment.
fields.conf	Create multivalue fields and add search capability for indexed fields.
global-banner.conf	Display a global banner on all pages in Splunk Web.

File	Purpose
health.conf	Set the default thresholds for proactive Splunk component monitoring.
indexes.conf	Manage and configure index settings.
inputs.conf	Set up data inputs.
instance.cfg	Designate and manage settings for specific instances of Splunk. This can be handy, for example, when identifying forwarders for internal searches.
limits.conf	Set various limits (such as maximum result size or concurrent real-time searches) for search commands.
literals.conf	Customize the text, such as search error strings, displayed in Splunk Web.
macros.conf	Define search macros in Settings.
messages.conf	Customize Splunk Web messages.
metric_rollups.conf	Set attribute/value pairs for metric rollup policy entries.
multikv.conf	Configure extraction rules for table-like events (ps, netstat, ls).
outputs.conf	Set up forwarding behavior.
passwords.conf	Maintain the credential information for an app.
procmon-filters.conf	Monitor Windows process data.
props.conf	Set indexing property configurations, including timezone offset, custom source type rules, and pattern collision priorities. Also, map transforms to event properties.
pubsub.conf	Define a custom client of the deployment server.
restmap.conf	Create custom REST endpoints.
rolling_upgrade.conf	Configure an automated search head cluster rolling upgrade.
savedsearches.conf	Define ordinary reports, scheduled reports, and alerts.
searchbnf.conf	Configure the search assistant.
segmenters.conf	Configure segmentation.
server.conf	Contains a variety of settings for configuring the overall state of a Splunk Enterprise instance. For example, the file includes settings for enabling SSL, configuring nodes of an **indexer cluster** or a **search head cluster**, configuring **KV store**, and setting up a **license manager**.
serverclass.conf	Define deployment server classes for use with deployment server.
serverclass.seed.xml.conf	Configure how to seed a deployment client with apps at start-up time.
source-classifier.conf	Terms to ignore (such as sensitive data) when creating a source type.
sourcetypes.conf	Machine-generated file that stores source type learning rules.
tags.conf	Configure tags for fields.
telemetry.conf	Enable apps to collect telemetry data about app usage and other properties.
times.conf	Define custom time ranges for use in the Search app.
transactiontypes.conf	Add additional transaction types for transaction search.
transforms.conf	Configure regex transformations to perform on data inputs. Use in tandem with props.conf.
ui-prefs.conf	Change UI preferences for a view. Includes changing the default earliest and latest values for the time range picker.
user-prefs.conf	Configure settings on a per-user basis for use by Splunk Web.

File	Purpose
user-seed.conf	Set a default user and password.
visualizations.conf	List the visualizations that an app makes available to the system.
viewstates.conf	Use this file to set up UI views (such as charts).
web.conf	Configure Splunk Web, enable HTTPS.
web-features.conf	Configure some Splunk Web settings.
wmi.conf	Set up Windows management instrumentation (WMI) inputs.
workflow_actions.conf	Configure workflow actions.
workload_policy.conf	Enable or disable admission rules in workload management.
workload_pools.conf	Configure workload pools (compute and memory resource groups) that you can assign to searches in workload management.
workload_rules.conf	Configure workload rules to define access and priority for workload pools in workload management.

Configuration parameters and the data pipeline

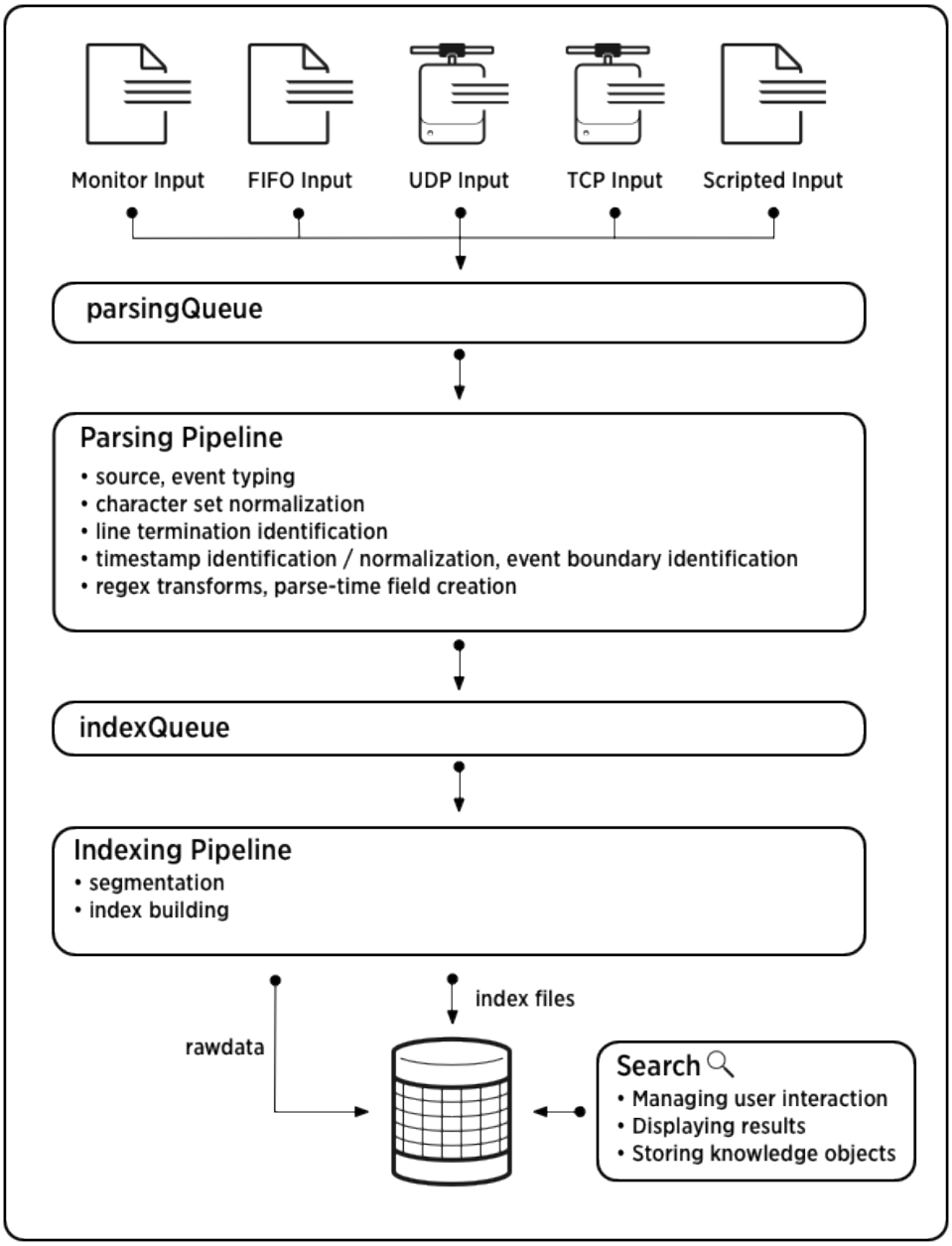
Data goes through several phases as it transitions from raw input to searchable events. This process is called the **data pipeline** and consists of four phases:

- **Input**
- **Parsing**
- **Indexing**
- **Search**

Each phase of the data pipeline relies on different configuration file parameters. Knowing which phase uses a particular parameter allows you to identify where in your Splunk deployment topology you need to set the parameter.

What the data pipeline looks like

This diagram outlines the data pipeline:


Monitor Input
FIFO Input
UDP Input
TCP Input
Scripted Input
parsingQueue
Parsing Pipeline
• source, event typing
• character set normalization
• line termination identification
• timestamp identification / normalization, event boundary identification
• regex transforms, parse-time field creation
indexQueue
Indexing Pipeline
• segmentation
• index building
index files
rawdata
Search
• Managing user interaction
• Displaying results
• Storing knowledge objects

The Distributed Deployment manual describes the data pipeline in detail, in "How data moves through Splunk: the data pipeline".

How Splunk Enterprise components correlate to phases of the pipeline

One or more Splunk Enterprise components can perform each of the pipeline phases. For example, a universal forwarder, a heavy forwarder, or an indexer can perform the input phase.

Data only goes through each phase once, so each configuration belongs on only one component, specifically, the first component in the deployment that handles that phase. For example, say you have data entering the system through a set of universal forwarders, which forward the data to an intermediate heavy forwarder, which then forwards the data onwards to an indexer. In that case, the input phase for that data occurs on the universal forwarders, and the parsing phase occurs on the heavy forwarder.

Data pipeline phase	Components that can perform this role
Input	**indexer** **universal forwarder** **heavy forwarder**
Parsing	indexer heavy forwarder light/universal forwarder (in conjunction with the `INDEXED_EXTRACTIONS` attribute only)
Indexing	indexer
Search	indexer **search head**

Where to set a configuration parameter depends on the components in your specific deployment. For example, you set parsing parameters on the indexers in most cases. But if you have heavy forwarders feeding data to the indexers, you instead set parsing parameters on the heavy forwarders. Similarly, you set search parameters on the search heads, if any. But if you aren't deploying dedicated search heads, you set the search parameters on the indexers.

For more information, see "Components and the data pipeline" in the *Distributed Deployment Manual.*

How configuration parameters correlate to phases of the pipeline

This is a non-exhaustive list of configuration parameters and the pipeline phases that use them. By combining this information with an understanding of which Splunk component in your particular deployment performs each phase, you can determine where to configure each setting.

For example, if you are using universal forwarders to consume inputs, you need to configure `inputs.conf` parameters on the forwarders. If, however, your indexer is directly consuming network inputs, you need to configure those network-related `inputs.conf` parameters on the indexer.

The following items in the phases below are listed in the order Splunk applies them (ie `LINE_BREAKER` occurs before `TRUNCATE`).

Input phase

- inputs.conf
- props.conf
  - `CHARSET`
  - `NO_BINARY_CHECK`

    - CHECK_METHOD
    - CHECK_FOR_HEADER (deprecated)
    - PREFIX_SOURCETYPE
    - sourcetype
- wmi.conf
- regmon-filters.conf

Structured parsing phase

- props.conf
    - INDEXED_EXTRACTIONS, and all other structured data header extractions

Parsing phase

- props.conf
    - LINE_BREAKER, TRUNCATE, SHOULD_LINEMERGE, BREAK_ONLY_BEFORE_DATE, and all other line merging settings
    - TIME_PREFIX, TIME_FORMAT, DATETIME_CONFIG (datetime.xml), TZ, and all other time extraction settings and rules
    - TRANSFORMS which includes per-event queue filtering, per-event index assignment, per-event routing
    - SEDCMD
    - MORE_THAN, LESS_THAN
- transforms.conf
    - stanzas referenced by a TRANSFORMS clause in props.conf
    - LOOKAHEAD, DEST_KEY, WRITE_META, DEFAULT_VALUE, REPEAT_MATCH

Indexing phase

- props.conf
    - SEGMENTATION
- indexes.conf
- segmenters.conf

Search phase

- props.conf
    - EXTRACT
    - REPORT
    - LOOKUP
    - KV_MODE
    - FIELDALIAS
    - EVAL
    - rename
- transforms.conf
    - stanzas referenced by a REPORT clause in props.conf
    - filename, external_cmd, and all other lookup-related settings
    - FIELDS, DELIMS
    - MV_ADD
- lookup files in the lookups folders
- search and lookup scripts in the bin folders
- search commands and lookup scripts
- savedsearches.conf
- eventtypes.conf
- tags.conf
- commands.conf

- alert_actions.conf
- macros.conf
- fields.conf
- transactiontypes.conf
- multikv.conf

Other configuration settings

There are some settings that don't work well in a distributed Splunk environment. These tend to be exceptional and include:

- props.conf
  - `CHECK_FOR_HEADER` (deprecated), `LEARN_MODEL`, `maxDist`. These are created in the parsing phase, but they require generated configurations to be moved to the search phase configuration location.

Back up configuration information

All Splunk's configuration information is contained in **configuration files**. To back up the set of configuration files, make an archive or copy of `$SPLUNK_HOME/etc/`. This directory, along with its subdirectories, contains all the default and custom settings for your Splunk install, and all apps, including saved searches, user accounts, tags, custom source type names, and other configuration information.

Copy this directory to a new Splunk instance to restore. You don't have to stop Splunk to do this.

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

Back up the cluster manager node

If you're using **index replication**, you can back up the manager node's static configuration. This is of particular use when configuring a stand-by manager that can take over if the primary manager fails. For details, see "Configure the manager" in the Managing Indexers and Clusters manual.

Check the integrity of your Splunk software files

Most files that Splunk software ships with should not be modified by end users or administrators. However, many users mistakenly modify these files. For example, someone might edit a configuration file in the default directory, or files might be corrupted by hardware flaws, file system problems, a mangled installation, or an errant script.

File validation can identify when the contents of the files of a Splunk software instance have been modified in a way that is not valid. You can run this check manually, and it also runs automatically on startup. If you are an admin, you can view the results in a Monitoring Console health check or in a dashboard from any node.

Run the check manually

You might want to run the integrity check manually under any of the following conditions:

- You have problems after an upgrade.
- You have symptoms that make you suspect that there may have been a storage system problem.

- You suspect or wish to guard against the common error of edits to the default .conf files.
- As part of a regular system check. See Customize the health check in the *Monitoring Splunk Enterprise* manual.

To run the check manually with default settings, from the installation directory, type `./splunk validate files`. You can manually run the integrity check with two controls.

- You can specify the file describing the correct file contents with `-manifest`. You might want to do this to check against an old manifest from a prior installation after a botched upgrade, to validate that the files are simply stale. You can use any valid manifest file. A manifest file ships in the installation directory with a new Splunk Enterprise download.
- You can constrain the test to only files that end with .conf by using `-type conf`. This is the set of messages the startup-time check prints to the terminal.

Options for automatic verification

The check runs at startup in two parts.

First, as part of the pre-flight check before splunkd starts, the check quickly validates only the default conf files and writes a message to your terminal.

Next, after splunkd starts, the check validates all files shipped with Splunk Enterprise (default conf files, libraries, binaries, data files, and so on). This more complete check writes the results to `splunkd.log` as well as to the bulletin message system in Splunk Web. You can configure it in `limits.conf`.

Options for the second part of the check in `limits.conf` include the following:

- run and log
- run, log, and emit a message to Splunk Web
- disable it

See limits.conf.spec.

Reading all the files provided with the installation has a moderate effect on I/O performance. If you need to restart Splunk software several times in a row, you might wish to disable this check temporarily to improve I/O performance.

Files are validated against the manifest file in the installation directory. If this file is removed or altered, the check cannot work correctly.

View results in Splunk Web

If you are an admin, you can view the results in a Monitoring Console health check or in a dashboard from any node. See Access and customize health check for more information about the Monitoring Console health check.

To view the default dashboard from any node:

1. Log in as admin to Splunk Web on any node in your deployment.
2. From Splunk Home, click **Search & Reporting** to enter the Search & Reporting app.
3. In the Apps bar, click **Dashboards**.
4. In the list of dashboards, click **Integrity Check of Installed Files**.

Interpret results of an integrity check

If an integrity check returns an error, such as "File Integrity checks found files that did not match the system-provided manifest", here are some tips to get you started resolving the problem.

- If the integrity check complains about conf files in default directories, determine how these files became changed and avoid this practice in the future. Modified default conf files will be overwritten on upgrade, creating hard-to-identify problems. See How to edit a configuration file for more details on how to edit configuration files in Splunk software.
- If it complains about files in `$SPLUNK_HOME/bin` or `$SPLUNK_HOME/lib`, or on Windows `%SPLUNK_HOME%\Python2.7\`, you probably need to reinstall. First try to find out how Splunk software was installed locally and determine whether this process could have resulted in a mix of files from different versions. AIX can cause this problem by holding library files open even after the Splunk service has been shut down. On most platforms this type of problem can occur when a Splunk product is upgraded while it is still running. If you cannot determine how this situation occurred, or how to resolve it, work with Splunk Support to identify the issue.
- If it cannot read some files, Splunk software may have been run as two or more different users or security contexts. Files created at install time under one user or context might not be readable by the service now running as another context. Alternatively, you might have legitimately modified the access rules to these files, but this is far less common.
- If the integrity check reports that it cannot read or comprehend the manifest, the manifest might be simply missing from `$SPLUNK_HOME`, or you have access problems to it, or the file may be corrupted. You might want to evaluate whether all the files from the installation package made it to the installation directory, and that the manifest contents are the same as the ones from the package. The manifest is not required for Splunk software to function, but the integrity check cannot function without it.
- If the integrity check reports all or nearly all files are incorrect, splunkd and `etc/splunk.version` might be in disagreement with the rest of the installation. Try to determine how this could have happened. It might be that the majority of the files are the ones you intended to be present.
- If the pattern is not described above, you might need to apply local analysis and troubleshooting skills possibly in concert with Splunk Support.

Interaction with monitoring console health check

The monitoring console health check queries the `server/status/installed-file-integrity` endpoint. This endpoint is populated with results when the integrity check runs at startup. See server/status/installed-file-integrity in the *REST API Reference Manual*.

If Splunk Enterprise starts with the integrity check disabled in `limits.conf`, then REST file integrity information is not available. In addition, manual runs do not update the results.

See Access and customize health check in *Monitoring Splunk Enterprise*.

Administer Splunk Enterprise with the command line interface (CLI)

About the CLI

You can use the Splunk Enterprise command line interface (CLI) to monitor, configure, and run searches and other tasks. The CLI help exists in the product and is accessible through a terminal window or command or shell prompt. Read this topic to learn how to access the CLI.

Access the CLI

The Splunk Enterprise CLI is located in the `$SPLUNK_HOME/bin` directory of the Splunk Enterprise installation. On Windows machines, the CLI appears in the `%SPLUNK_HOME%\bin` directory.

You can find the Splunk Enterprise installation path on your instance through Splunk Web by clicking **Settings > Server settings > General settings**.

To access the Splunk Enterprise CLI, you must have:

- A shell prompt, command prompt, or PowerShell session
- Access to a Splunk platform instance or forwarder, or
- Permission to access the correct port on a remote Splunk Enterprise instance.

CLI help documentation

If you have administrator privileges, you can use the CLI not only to search but also to configure and monitor your Splunk Enterprise instance or instances. The CLI commands that configure and monitor Splunk are not search commands. Search commands are arguments to the `search` and `dispatch` CLI commands. Some commands require that you authenticate with a username and password or specify a target Splunk server.

You can look up help information for the CLI using:

UNIX	Windows
`./splunk help`	`.\splunk help`

For more information about how to access help for specific CLI commands or tasks, see "Get help with the CLI" and "Administrative CLI commands" in this manual.

Work with the CLI on *nix

If you have administrator or root privileges, you can simplify CLI access by adding the top level directory of your Splunk Enterprise installation, `$SPLUNK_HOME/bin`, to your shell path. If you installed Splunk Enterprise in a different directory, specify that directory in the following commands.

This example works for Linux/BSD/Solaris users who installed Splunk Enterprise in the default location:

```
# export SPLUNK_HOME=/opt/splunk
# export PATH=$SPLUNK_HOME/bin:$PATH
```

This example works for Mac users who installed Splunk Enterprise in the default location:

```
# export SPLUNK_HOME=/Applications/Splunk
# export PATH=$SPLUNK_HOME/bin:$PATH
```

Now you can invoke CLI commands using:

splunk <command>

To set the `$SPLUNK_HOME` environment variable while working in a CLI session:

- In *nix: `source /opt/splunk/bin/setSplunkEnv`
- In Windows: `splunk.exe envvars > setSplunkEnv.bat & setSplunkEnv.bat`

Splunk CLI skips password prompting for *nix users with access to the /home directory

On a *nix machine, if a *nix user that runs the Splunk CLI has access to the /home directory on that machine, the CLI does not prompt for the Splunk user password.

Mac OS X requires elevated privileges to access system files or directories

Mac OS X requires superuser level access to run any command that accesses system files or directories. Run CLI commands using **sudo** or "su -" for a new shell as root. The recommended method is to use sudo. (By default the user "root" is not enabled but any administrator user can use sudo.)

Work with the CLI on Windows

To run CLI commands in Splunk Enterprise on Windows, use PowerShell or the command prompt as an administrator.

1. Open a PowerShell window or command prompt as an administrator.
2. Change to the Splunk Enterprise `bin` directory.
3. Run a Splunk command by typing in `splunk` followed by the subcommand and any required arguments.

```
C:\Program Files\Splunk\bin> splunk status
splunkd is running.
splunk helpers are running.
```

You can run many commands and perform many tasks from the Splunk Enterprise CLI. For help on using the CLI, see Get help with the CLI.

Set Splunk environment variables on Windows

You do not need to set Splunk environment variables to use the CLI on Windows. If you want to use variables to run CLI commands, you must set variables manually.

Set temporary variables on the command prompt

1. Open a PowerShell window, or a command prompt.
2. Use a Powershell variable or environment variable to set a quick reference path to Splunk Enterprise.

PowerShell	Command prompt
`$splunk_home="C:\Program Files\Splunk"`	`set SPLUNK_HOME="C:\Program Files\Splunk"`

3. Call the variable when running Splunk Enterprise CLI commands.

PowerShell	Command prompt
`& $splunk_home\bin\splunk status`	`%SPLUNK_HOME%\bin\splunk status`

Set permanent environment variables

To set a permanent variable, see Add or change environment variables on MS TechNet.

Answers

Have questions? Visit Splunk Answers and see what questions and answers the Splunk community has about using the CLI.

Get help with the CLI

This topic discusses how to access Splunk's built-in CLI help reference, which contains information about the CLI commands and how to use them. This topic also briefly discusses the universal parameters, which are parameters that you can use with any CLI command.

Access CLI help reference

If you need to find a CLI command or syntax for a CLI command, use Splunk's built-in CLI help reference.

To start, you can access the default help information with the `help` command:

./splunk help

This will return a list of objects to help you access more specific CLI help topics, such as administrative commands, clustering, forwarding, licensing, searching, etc.

Universal parameters

Some commands require that you authenticate with a username and password, or specify a target host or app. For these commands you can include one of the universal parameters: `auth`, `app`, or `uri`.

```
./splunk [command] [object] [-parameter <value> | <value>]... [-app] [-owner] [-uri] [-auth]
```

Parameter	Description
app	Specify the App or namespace to run the command; for search, defaults to the Search App.
auth	Specify login credentials to execute commands that require you to be logged in.
owner	Specify the owner/user context associated with an object; if not specified, defaults to the currently logged in user.
uri	Excute a command on any specified (remote) Splunk server.

app

In the CLI, `app` is an object for many commands, such as `create app` or `enable app`. But, it is also a parameter that you can add to a CLI command if you want to run that command on a specific app.

Syntax:

```
./splunk command object [-parameter value]... -app appname
```

For example, when you run a search in the CLI, it defaults to the Search app. If want to run the search in another app:

./splunk search "eventype=error | stats count by source" -detach f -preview t -app unix

auth

If a CLI command requires authentication, Splunk will prompt you to supply the username and password. You can also use the `-auth` flag to pass this information inline with the command. The `auth` parameter is also useful if you need to run a command that requires different permissions to execute than the currently logged-in user has.

Syntax:

```
./splunk command object [-parameter value]... -auth username:password
```

uri

If you want to run a command on a remote Splunk server, use the `-uri` flag to specify the target host.

Syntax:

```
./splunk command object [-parameter value]... -uri specified-server
```

Specify the target Splunk server with the following format:

```
[http|https]://name_of_server:management_port
```

You can specify an IP address for the `name_of_server`. Both IPv4 and IPv6 formats are supported; for example, the `specified-server` may read as: 127.0.0.1:80 or "[2001:db8::1]:80". By default, splunkd listens on IPv4 only. To enable IPv6 support, see Configure Splunk Enterprise for IPv6.

Example: The following example returns search results from the remote "splunkserver" on port 8089.

./splunk search "host=fflanda error 404 *.gif" -auth admin -uri https://splunkserver:8089

For more information about the CLI commands you can run on a remote server, see the next topic in this chapter.

Useful help topics

When you run the default Splunk CLI help, you will see these objects listed.

Administrative CLI commands

You can use the CLI for administrative functions such as adding or editing inputs, updating configuration settings, and searching. If you want to see the list of administrative CLI commands type in:

./splunk help commands

These commands are discussed in more detail in "Administrative CLI commands", the next topic in this manual.

CLI help for indexer clustering

Indexer clustering is a Splunk feature that consists of clusters of indexers configured to replicate data to achieve several goals: data availability, data fidelity, disaster tolerance, and improved search performance.

You can use the CLI to view and edit clustering configurations on the indexer cluster nodes. For the list of commands and parameters related to clustering, type in:

./splunk help clustering

For more information, read "Configure the cluster with the CLI" in the *Managing Indexers and Clusters* manual.

CLI help for Splunk controls

Use the CLI to start, stop, and restart Splunk server (`splunkd`) and web (`splunkweb`) processes or check to see if the process is running. For the list of controls, type in:

./splunk help controls

For more information, read "Start and stop Splunk" in the Admin Manual.

CLI help for data management

When you add data to Splunk, Splunk processes it and stores it in an **index**. By default, data you feed to Splunk is stored in the **main** index, but you can use the CLI to create and specify other indexes for Splunk to use for different data inputs. To see the list of objects and commands to manage indexes and datastores, type in:

./splunk help datastore
./splunk help index

For more information, read "About managing indexes", "Create custom indexes", and "Remove indexes and data from Splunk" in the *Managing Indexers and Clusters* manual.

CLI help for distributed search deployments

Use the CLI to view and manage your distributed search configurations. For the list of objects and commands, type in:

./splunk help distributed

For information about distributed search, read "About distributed search" in the *Distributed Search* manual.

CLI help for forwarding and receiving

Splunk deployments can include dozens or hundreds of forwarders forwarding data to one or more receivers. Use the CLI to view and manage your data forwarding configuration. For the list of forwarding objects and commands, type in:

./splunk help forwarding

For more information, read "About forwarding and receiving" in the *Forwarding Data* manual.

CLI help for search and real-time search

You can also use the CLI to run both historical and real-time searches. Access the help page about Splunk search and real-time search with:

./splunk help search
./splunk help rtsearch

Also, use objects `search-commands`, `search-fields`, and `search-modifiers` to access the respective help descriptions and syntax:

./splunk help search-commands
./splunk help search-fields
./splunk help search-modifiers

Note: The Splunk CLI interprets spaces as breaks. Use dashes between multiple words for topic names that are more than one word.

To learn more about searching your data with the CLI, refer to "About CLI searches" and "Syntax for CLI searches" in the Search Reference Manual and "Real-time searches and reports in the CLI" in the Search Manual.

Administrative CLI commands

This topic discusses the administrative CLI commands, which are the commands used to manage or configure your Splunk server and distributed deployment.

For information about accessing the CLI and what is covered in the CLI help, see the previous topic, Get help with the CLI. If you're looking for details about how to run searches from the CLI, see About CLI searches in the *Search Reference*.

Your Splunk role configuration dictates what actions (commands) you can execute. Most actions require you to have Splunk admin privileges. Read more about setting up and managing Splunk users and roles in the About users and roles topic in the *Admin Manual*.

Splunk CLI command syntax

The general syntax for a CLI command is this:

```
./splunk <command> [<object>] [[-<parameter>] <value>]...
```

Note the following:

- Some commands don't require an object or parameters.
- Some commands have a default parameter that can be specified by its value alone.
- Some commands can take extra parameters like `-uri` or `-auth`. See the "Universal parameters" section of Get help with the CLI.

Commands, objects, and examples

A **command** is an action that you can perform. An **object** is something you perform an action on.

Most administrative CLI commands are offered as an alternative interface to the Splunk Enterprise REST API without the need for the `curl` command. If you're looking for additional uses or options for a CLI command object, review the REST API Reference Manual and search for the object name.

Command	Objects	Examples
add	exec, forward-server, index, licenser-pools, licenses, manager, monitor, oneshot, saved-search, search-server, tcp, udp, user	**1.** Adds monitor directory and file inputs to source `/var/log`. `./splunk add monitor /var/log/`
		2. Adds another indexer cluster manager node to the list of instances the search head searches across. `./splunk add cluster-manager https://127.0.0.1:8089 -secret testsecret -multisite false`
anonymize	source	**1.** Replaces identifying data, such as usernames and IP addresses, in the file located at `/tmp/messages`. `./splunk anonymize file -source /tmp/messages`
		2. Anonymizes `Mynames.txt` using name-terms, a file containing a list of common English personal names. `./splunk anonymize file -source /tmp/messages -name_terms $SPLUNK_HOME/bin/Mynames.txt`
apply	cluster-bundle, shcluster-bundle	**1.** Makes validated bundle active on peers. `./splunk apply cluster-bundle`
		2. Skip-validation is an optional argument to skip bundle validation on the indexer cluster manager and peers. `./splunk apply cluster-bundle --skip-validation`
		3. For `shcluster-bundle` examples, see Deploy a configuration bundle in the *Distributed Search* manual.
check-integrity	NONE	**1.** Verifies the integrity of an index with the optional parameter `verbose`.

Command	Objects	Examples
		`./splunk check-integrity -index $SPLUNK_HOME/var/lib/splunk/defaultdb/ [-<verbose> ]`
		2. Verifies the integrity of a bucket with the optional parameter `verbose`. `./splunk check-integrity -bucketPath $SPLUNK_HOME/var/lib/splunk/defaultdb/db/ [-<verbose> ]`
clean	all, eventdata, globaldata, inputdata, userdata, kvstore	**1.** Removes data from Splunk installation. `eventdata` refers to exported events indexed as raw log files. `./splunk clean eventdata`
		2. `globaldata` refers to host tags and source type aliases. `./splunk clean globaldata`
cluster-manager-redundancy	NONE	**1.** Shows status of all the cluster managers in redundancy mode. `./splunk cluster-manager-redundancy -show-status`
		2. Switches HA mode of a cluster manager from standby to active. `./splunk cluster-manager-redundancy -switch-mode active`
		3. Switches HA mode of a cluster manager from active to standby. Consequently, another, currently standby cluster manager gets switched to active automatically. `./splunk cluster-manager-redundancy -switch-mode standby`
cmd	btprobe, classify, locktest, locktool, pcregextest, searchtest, signtool, toCsv, toSrs, tsidxprobe, walklex	**1.** Displays the contents in the `$SPLUNK_HOME/bin` directory. `./splunk cmd /bin/ls`
		2. Runs the chosen command from the `$SPLUNK_HOME/bin` directory with the environment variables set. Run `splunk envvars` to see which environment variables are set. `./splunk cmd locktest`
create	app	**1.** Builds myNewApp from a template. `./splunk create app myNewApp -template sample_app`
createssl	NONE	
diag	NONE	
disable		

Command	Objects	Examples
	app, boot-start, deploy-client, deploy-server, dist-search, index, listen, local-index, maintenance-mode, perfmon, webserver, web-ssl, wmi	**1.** Disables the maintenance mode on peers in indexer clustering. Must be invoked on the manager node. `'./splunk disable maintenance-mode'`
		2. Disables the logs1 collection. `./splunk disable eventlog logs1`
display	app, boot-start, deploy-client, deploy-server, dist-search, jobs, listen, local-index	**1.** Displays status information, such as enabled/disabled, for all apps. `./splunk display app`
		2. Displays status information for the unix app. `./splunk display app unix`
edit	app, cluster-config, shcluster-config, exec, index, licenser-localpeer, licenser-groups, monitor, saved-search, search-server, tcp, udp, user	**1.** Edits the current clustering configuration. `./splunk edit cluster-config -mode peer -site site2`
		2. Edits monitored directory inputs in `/var/log` and only reads from the end of this file. `./splunk edit monitor /var/log -follow-only true`
enable	app, boot-start, deploy-client, deploy-server, dist-search, index, listen, local-index, maintenance-mode, perfmon, webserver, web-ssl, wmi	**1.** Sets the maintenance mode on peers in indexer clustering. Must be invoked on the manager node. `'./splunk enable maintenance-mode'`
		2. Enables the `col1` collection. `./splunk enable perfmon col1`
export	eventdata, user data	**1.** Exports data out of your Splunk server into `/tmp/apache_raw_404_logs`. `./splunk export eventdata -index my_apache_data -dir /tmp/apache_raw_404_logs -host localhost -terms "404 html"`
fsck	repair, scan, clear-bloomfilter	
help	NONE	
import	userdata	**1.** Imports user accounts data from directory `/tmp/export.dat`. `./splunk import userdata -dir /tmp/export.dat`
install	app	**1.** Installs the app from foo.tar to the local Splunk server.

Command	Objects	Examples
		`./splunk install app foo.tar`
		2. Installs the app from foo.tgz to the local Splunk server. `./splunk install app foo.tgz`
list	cluster-buckets, cluster-config, cluster-generation, cluster-peers, deploy-clients, excess-buckets, exec, forward-server, index, inputstatus, licenser-groups, licenser-localpeer, licenser-messages, licenser-pools, licenser-peers, licenser-stacks, licenses, jobs, manager-info, monitor, peer-info, peer-buckets, perfmon, saved-search, search-server, tcp, udp, user, wmi	**1.** Lists all active monitored directory and file inputs. This displays files and directories currently or recently monitored by splunkd for change. `./splunk list monitor`
		2. Lists all licenses across all stacks. `./splunk list licenses`
login,logout	NONE	
migrate	kvstore-storage-engine	**1.** Migrates the KV store to the target storage engine. `./splunk migrate kvstore-storage-engine --target-engine wiredTiger`
offline	NONE	**1.** Used to shutdown the peer in a way that does not affect existing searches. The manager node rearranges the primary peers for buckets, and fixes up the cluster state in case the enforce-counts flag is set. `./splunk offline`
		2. Because the `--enforce-counts` flag is used, the cluster is completely fixed up before this peer is taken down. `./splunk offline --enforce-counts`
package	app	**1.** Packages the app "stubby" and returns the package location. `./splunk package app stubby` The `package` command includes local.meta by default. However, if your app package contains local.meta, it will fail AppInspect app validation. To avoid AppInspect failure, use either the `-merge-local-meta` or `-exclude-local-meta` flag.
		2. When packaging the app, merges local.meta to default.meta and packages the resulting default.meta.

Command	Objects	Examples
		`./splunk package app stubby -merge-local-meta true`
		3. When packaging the app, excludes the local.meta from the app package. `./splunk package app stubby -exclude-local-meta true`
rebalance	cluster-data	**1.** Rebalances data for all indexes. `./splunk rebalance cluster-data -action start`
		2. Rebalances data for a single index using the optional `-index` parameter. `./splunk rebalance cluster-data -action start -index _internal`
		3. Rebalances data using the optional `-max_runtime` parameter to limit the rebalancing activity to 5 minutes. `./splunk rebalance cluster-data start -max_runtime 5`
rebuild	NONE	
reload	ad, auth, deploy-server, exec, index, listen, monitor, registry, tcp, udp, perfmon, wmi	**1.** Reloads your deployment server, in entirety or by server class. `./splunk reload deploy-server`
		2. Reloads my_serverclass. `./splunk reload deploy-server -class my_serverclass`
		3. Reloads a specific index configuration. To reload all indexes, do not include an index name. `./splunk reload index [index_name]`
remove	app, cluster-peers, cluster-manager, excess-buckets, exec, forward-server, index, jobs, licenser-pools, licenses, monitor, saved-search, search-server, tcp, udp, user	**1.** Removes the cluster manager node from the list of instances the search head searches across. Uses testsecret as the secret/pass4SymmKey. `'./splunk remove cluster-manager https://127.0.0.1:8089 -secret testsecret'`
		2. Removes the Unix app. `./splunk remove app unix`
rollback	cluster-bundle	Rolls back your Splunk Web configuration bundle to your previous version. From the manager node, run this command: `./splunk rollback cluster-bundle`
rolling-restart	cluster-peers, shcluster-members	

Command	Objects	Examples
rtsearch	app, batch, detach, earliest_time, header, id, index_earliest, index_latest, max_time, maxout, output, preview, rt_id, timeout, uri, wrap	**1.** Runs a real-time search that does not line-wrap for individual lines. `./splunk rtsearch 'error' -wrap false`
		2. Runs a real-time search. Use `rtsearch` exactly as you use the traditional search command. `./splunk rtsearch 'eventtype=webaccess error | top clientip'`
search	app, batch, detach, earliest_time, header, id, index_earliest, index_latest, latest_time, max_time, maxout, output, preview, timeout, uri, wrap	**1.** Uses the wildcard as the search object. Triggers an asynchronous search and displays the job id and ttl for the search. `./splunk search '*' -detach true`
		2. Uses `eventtype=webaccess error` as the search object. Does not line wrap for individual lines that are longer than the terminal width. `./splunk search 'eventtype=webaccess error' -wrap 0`
set	datastore-dir, deploy-poll, default-hostname, default-index, indexing-ready, minfreemb, servername, server-type, splunkd-port, web-port, kvstore-port	**1.** Sets the force indexing ready bit. `./splunk set indexing-ready`
		2. Sets `bologna:1234` as the deployment server to poll updates from. `./splunk set deploy-poll bologna:1234`
show	config, cluster-bundle-status, datastore-dir, deploy-poll, default-hostname, default-index, jobs, minfreemb, servername, splunkd-port, web-port, kvstore-port, kvstore-status, shcluster-kvmigration-status	**1.** Shows current logging levels. `./splunk show log-level`
		2. Shows which deployment server Splunk Enterprise is configured to poll from. `./splunk show deploy-poll`
spool	NONE	
start-shcluster-migration	kvstore	**1.** Migrate the KV store to the target storage engine in a clustered environment. `./splunk start-shcluster-migration kvstore -storageEngine wiredTiger`
		2. Check to see if the KV store is ready to migrate to the target storage engine. `./splunk start-shcluster-migration kvstore -storageEngine wiredTiger -isDryRun`
start,stop,restart	splunkd, splunkweb	
status	splunkd, splunkweb	

Command	Objects	Examples
validate	index, files, cluster-bundle	**1.** Validates the main index and verifies the index paths specified in `indexes.conf`. `./splunk validate index main`
		2. For `files` examples, see Check the integrity of your Splunk software files.
		3. For `cluster-bundle` examples, see Update common peer configurations and apps in the *Managing Indexers and Clusters of Indexers* manual.
version	NONE	

Exporting search results with the CLI

You can use the CLI to export large numbers of search results. For information about how to export search results with the CLI, as well as information about the other export methods offered by Splunk Enterprise, see Export search results in the *Search Manual*.

Troubleshooting with the CLI

The Splunk CLI also includes tools that help with troubleshooting. Invoke these tools using the CLI command `cmd`:

./splunk cmd <tool>

For the list of CLI utilities, see Command line tools for use with Support in the *Troubleshooting Manual*.

Use the CLI to administer a remote Splunk Enterprise instance

You can use the `-uri` argument with any CLI command to send that command to another Splunk Enterprise instance and view the results on your local instance.

Read this topic to learn about the following concepts:

- Syntax for using the `uri` argument.
- How the CLI verifies the host name of the remote server you are connecting to
- CLI commands that you cannot use remotely.

Remote CLI access is disabled by default for the admin user until you have changed its default password.

The CLI verifies the host names of machines to which it connects over TLS

In version 9.0.0 and higher of Splunk Enterprise, the CLI verifies the host name of the machine you connect to using the `-uri` argument. This validation check also happens on the local instance if you use the `-uri` argument to connect locally.

If the machine does not have valid transport layer security (TLS) certificates installed and configured, the remote CLI connection fails. Failed connections do not return any information from the remote instance.

You can temporarily disable TLS host name verification by using the `-no-host-name-check` argument within the CLI command, but this reduces security overall and subjects you to potential cyberattacks between the instance on which you run the CLI and the instance to which the CLI attempts to connect.

For more information on TLS certificates and how to obtain, install, and configure them, see About securing the Splunk platform with TLS in *Securing the Splunk Platform*.

Enable remote access

If you run Splunk Free, which has no login credentials, remote access is disabled by default until you've edited the `[general]` stanza of the `$SPLUNK_HOME/etc/system/local/server.conf` configuration file, and set the following value:

```
allowRemoteLogin=always
```

The add oneshot command works on local instances but cannot be used remotely.

For more information about editing configuration files, see About configuration files in this manual.

Send CLI commands to a remote server

The general syntax for using the `uri` parameter with any CLI command is:

```
./splunk command object [-parameter <value>]... [-no-host-name-check] -uri <specified-server>
```

The `uri` value, `specified-server` is formatted as:

```
[http|https]://name_of_server:management_port
```

Also, the `name_of_server` can be the fully resolved domain name or the IP address of the remote Splunk Enterprise instance.

This `uri` value is the `mgmtHostPort` value that you defined in the `web.conf` configuration file on the remote Splunk Enterprise instance. For more information, see the web.conf reference.

For general information about the CLI, see About the CLI and Get help with the CLI in this manual.

Search a remote instance

The following example returns search results from the remote "splunkserver".

```
./splunk search "host=fflanda error 404 *.gif" -uri https://splunkserver:8089
```

For details on syntax for searching using the CLI, refer to About CLI searches in the *Search Reference Manual*.

View apps installed on a remote instance

The following example returns the list of apps that are installed on the remote "splunkserver".

```
./splunk display app -uri https://splunkserver:8089
```

Change your default URI value

You can set a default URI value using the SPLUNK_URI environment variable. If you change this value to be the URI of the remote server, you do not need to include the `uri` parameter each time you want to access that remote server.

To change the value of SPLUNK_URI, type either:

```
$ export SPLUNK_URI=[http|https]://name_of_server:management_port     # For Unix shells
C:\> set SPLUNK_URI=[http|https]://name_of_server:management_port     # For Windows shell
```

For the examples above, you can change your SPLUNK_URI value by typing:

```
$ export SPLUNK_URI=https://splunkserver:8089
```

CLI commands you cannot run remotely

You can run most CLI commands remotely, with a few exceptions.

You cannot remotely run commands that control the server. These server control commands include:

- start, stop, restart
- status, version

In addition, you cannot run these commands remotely:

- add, edit, list, remove search-server
- add oneshot

You can view all CLI commands by accessing the CLI help reference. See Get help with the CLI in this manual.

Customize the CLI login banner

If you provide CLI access to data, you may need to customize your login banner to notify your users of monitoring, their legal obligations, and penalties for misuse. You can also add additional security (in the form of basic authentication) for your CLI logins.

To create a custom login banner and add basic authentication, add the following stanzas to your local server.conf file:

```
[httpServer]
cliLoginBanner = <string>
allowBasicAuth = true|false
basicAuthRealm = <string>
```

- For `cliLoginBanner = <string>`

Create a message that you want your user to see in the Splunk CLI, such as access policy information, before they are prompted for authentication credentials. The default value is no message.

To create a multi-line banner, place the lines in a comma separated list, putting each line in double-quotes. For example:

```
cliLoginBanner="Line 1","Line 2","Line 3"
```

To include a double quote within the banner text, use two quotes in a row. For example:

```
cliLoginBanner="This is a line that ""contains quote characters""!"
```

- For `allowBasicAuth = true|false`:

Set this value to `true` if you want to require clients to make authenticated requests to the Splunk server using "HTTP Basic" authentication in addition to Splunk's existing (`authtoken`) authentication. This is useful for allowing programmatic access to REST endpoints and for allowing access to the REST API from a web browser. It is not required for the UI or CLI. The default value is `true`.

- For `basicAuthRealm = <string>`:

If you have enabled `allowBasicAuth`, use this attribute to add a text string that can be presented in a Web browser when credentials are prompted. You can display a short message that describes the server and/or access policy. The text: "/splunk" displays by default.

Start Splunk Enterprise and perform initial tasks

Start and stop Splunk Enterprise

This topic provides common methods for starting and stopping Splunk Enterprise.

Start Splunk Enterprise on Windows

Splunk Enterprise installations are placed into the path `C:\Program Files\Splunk` by default. The documentation will refer to this default path as `%SPLUNK_HOME%`. Splunk Enterprise installs one service named `splunkd`. In normal operation, only the `splunkd` service runs and handles all Splunk Enterprise operations, including the Splunk Web interface.

You can start and stop Splunk Enterprise on Windows in one of the following ways:

Use the Windows Services control panel.

1. Click the Start Button and type "services."
2. Select the Services control panel option.
3. In the Services control panel, find the `Splunkd Service` service.
4. Start or stop the service.

Use the `NET START` or `NET STOP` commands.

1. Open an administrative command prompt.
2. Type: `NET START splunkd` or `NET STOP splunkd`.

Use the Splunk Enterprise executable.

1. Open an administrative command prompt.
2. Change the path to `%SPLUNK_HOME%\bin`.
3. Type: `splunk [start|stop|restart]`.

Start Splunk Enterprise on *nix

Splunk Enterprise installations using a package (.rpm or .deb) will install into the path `/opt/splunk` by default. The documentation will refer to this default path as `$SPLUNK_HOME`. Splunk Enterprise installs one process named `splunkd`. In normal operation, only the `splunkd` process runs and handles all Splunk Enterprise operations, including the Splunk Web interface.

You can start and stop Splunk Enterprise on *nix in one of the following ways:

Use the Splunk Enterprise process.

1. Log in as the user account running Splunk Enterprise processes.
2. Open a shell prompt.
3. Change the path to `$SPLUNK_HOME/bin`
4. Type: `splunk [start|stop|restart]`.

Use a service command. If you configured Splunk Enterprise to start at boot time, you will interact with the process using the service command. Using the service command ensures that the user configured in the init.d script starts the process. See Enable boot-start on *nix platforms.

1. Open a shell prompt.
2. Type: `splunkd service [start|stop|restart]`.

Use systemd commands. If you configured Splunk Enterprise to use systemd, you will interact with the process using the systemctl command. See Configure systemd using enable boot-start.

1. Open a shell prompt.
2. Type: `systemctl [start|stop|restart] Splunkd.service`.

Restart Splunk Enterprise from Splunk Web

You can restart Splunk Enterprise from Splunk Web:

1. Log into Splunk Web as an admin role
2. In Splunk Web, go to **Settings > Server controls**
3. Select "Restart Splunk"

Check if Splunk Enterprise is running

To verify that the Splunk Enterprise processes are running:

Use the "status" command on *nix.

1. Log in as the user account running Splunk Enterprise processes.
2. Open a shell prompt.
3. Change the path to `$SPLUNK_HOME/bin`.
4. Type: `splunk status`.

Use the "status" command on Windows.

1. Open an administrative command prompt.
2. Change the path to `%SPLUNK_HOME%\bin`.
3. Type: `splunk status`.

Use the process viewer command on *nix

1. Open a shell prompt.
2. Type: `ps aux | grep splunkd | grep -v grep`.
3. Look for running processes.

Use the process list command on Windows.

1. Open a powershell prompt.
2. Type: `Get-process splunkd`.
3. Look for running processes.

Configure Splunk Enterprise to start at boot time

On most operating systems, you can configure Splunk software to start running automatically after the machine and operating system boots. This reduces interruption of both sending and receiving data. All on-premises versions of Splunk software can be configured this way. On *nix platforms, you must manually configure the software to start at boot time after you install it.

You can configure the software as either the root user, or as a regular user with the `sudo` command. Nearly all distributions include `sudo` but if yours does not have it, you should consult the help for your distribution to download, install, and configure it.

Enable boot-start on the Windows platform

On Windows, the installer configures Splunk software to start at machine startup. To disable this, see Disable boot-start on Windows at the end of this topic.

Enable boot-start on *nix platforms

Splunk provides a utility that updates your system boot configuration so that the software starts when the system boots up. This utility creates an `init` script (or makes a similar configuration change, depending on your OS).

1. Log into the machine that you have installed Splunk software on and that you want to configure to run at boot time.
2. Become the root user if able. Otherwise, you must run the following commands with the `sudo` utility.
3. Run the following command:
   ```
   [sudo] $SPLUNK_HOME/bin/splunk enable boot-start
   ```

The init.d boot-start script is not compatible with RHEL 8 and higher. You can instead configure systemd to manage boot start and run splunkd as a service. For more information, see Enable boot start on machines that run systemd.

Enable boot-start as a non-root user

If you do not run Splunk software as the root user, you can pass in the `-user` parameter to specify the Splunk software user. The user that you want to run Splunk software as must already exist. If it does not, then create the user prior to running this procedure.

The following procedure configures Splunk software to start at boot time as the user 'bob'. You can substitute 'bob' with the user that Splunk software should use to start at boot time on the local machine.

1. Log into the machine.
2. Become the root user.
3. Run the following command:

   ```
   [sudo] $SPLUNK_HOME/bin/splunk enable boot-start -user bob
   ```
4. Change the ownership of the Splunk installation directory and all its files to user bob:
   ```
   [sudo] chown -R bob $SPLUNK_HOME
   ```
5. Using a text editor, open `/etc/init.d/splunk` for editing.
6. Update the service file to add the service user, and enclose the command in single quotes. Note the addition of the user field and {USER} variables `su - ${USER} -c`, and the placement of single quotes to encapsulate only the service command. The init.d service file will have minor differences based upon the *nix distribution and version.

An example is provided in the "After" table.

Before

```
RETVAL=0

. /etc/init.d/functions

splunk_start() {
  echo Starting Splunk...
  "$SPLUNK_HOME/bin/splunk" start --no-prompt --answer-yes
  RETVAL=$?
  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/splunk
}
splunk_stop() {
  echo Stopping Splunk...
 "$SPLUNK_HOME/bin/splunk" stop
  RETVAL=$?
  [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/splunk
}
splunk_restart() {
  echo Restarting Splunk...
  "$SPLUNK_HOME/bin/splunk" restart
  RETVAL=$?
  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/splunk
}
splunk_status() {
  echo Splunk status:
  "$SPLUNK_HOME/bin/splunk" status
  RETVAL=$?
}
case "$1" in
```

After

```
RETVAL=0
USER=bob

. /etc/init.d/functions

splunk_start() {
  echo Starting Splunk...
  su - ${USER} -c '"$SPLUNK_HOME/bin/splunk" start --no-prompt --answer-yes'
  RETVAL=$?
  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/splunk
}
splunk_stop() {
  echo Stopping Splunk...
  su - ${USER} -c '"$SPLUNK_HOME/bin/splunk" stop'
  RETVAL=$?
  [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/splunk
}
splunk_restart() {
  echo Restarting Splunk...
  su - ${USER} -c '"$SPLUNK_HOME/bin/splunk" restart'
  RETVAL=$?
  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/splunk
}
splunk_status() {
  echo Splunk status:
  su - ${USER} -c '"$SPLUNK_HOME/bin/splunk" status'
  RETVAL=$?
}
```

After
`case "$1" in`

Confirm that each splunk command has single quotes around it, and is preceded with the service user substitution.

7. Save the file and close it.

Changes take effect the next time you boot the machine.

Enable boot-start on machines that run systemd

On Linux machines that use the `systemd` system manager, you can configure Splunk Enterprise to let `systemd` control it. By default, Splunk Enterprise configures itself to run as a `init`-managed service, and does not use `systemd`.

1. Log into the machine that you have installed Splunk software on and that you want to configure to run at boot time.
2. Become the root user if able. Otherwise, you must run the following commands with the `sudo` utility.
3. Run the following command:
   `[sudo] $SPLUNK_HOME/bin/splunk enable boot-start -user bob -systemd-managed 1`

See Run Splunk Enterprise as a systemd service for additional information on Splunk Enterprise and `systemd`.

Enable boot-start on machines that run AIX

These instructions work for both Splunk Enterprise and the AIX version of the Splunk universal forwarder. Splunk does not offer a version of Splunk Enterprise for AIX for versions later than 6.3.0.

The AIX version of Splunk does not register itself to auto-start on machine boot. You can configure it to use the System Resource Controller (SRC) to handle boot-time startup.

When you enable boot start on an AIX system, Splunk software interacts with the AIX SRC to enable automatic starting and stopping of Splunk services.

`mkssys -G splunk -s splunkd -p <path to splunkd> -u <splunk user> -a _internal_exec_splunkd -S -n 2 -f 9`
`mkssys -G splunk -s splunkweb -p <path to python> -u <splunk user> -a _internal_exec_splunkweb -S -n 15 -f 9` (on Splunk Enterprise only)

When you enable automatic boot start, the SRC handles the run state of the Splunk Enterprise service. You must use a different command to start and stop Splunk software manually.

- `/usr/bin/startsrc -s splunkd` to start Splunk software manually.
- `/usr/bin/stopsrc -s splunkd` to stop Splunk software manually.

If you try to start and stop the software with the `./splunk [start|stop]` method from the `$SPLUNK_HOME` directory, the SRC catches the attempt and displays the following message:

`Splunk boot-start is enabled. Please use /usr/bin/[startsrc|stopsrc] -s splunkd to [start|stop] Splunk.`
To prevent this message from occurring and restore the ability to start and stop Splunk Enterprise from the `$SPLUNK_HOME` directory, disable boot start:

```
[sudo] ./splunk disable boot-start
```

- For more information on the `mkssys` command line arguments, see Mkssys command on the IBM pSeries and AIX Information Center website.
- For more information on the SRC, see System resource controller on the IBM Knowledge Center website.

Enable boot-start on AIX to run Splunk Software as a root user

1. Log into the AIX machine.
2. Become the root user, if able. Otherwise, you must prepend `sudo` to the following command examples. If you do not have `sudo` on your AIX instance, you must download, install, and configure it.
3. Change to the Splunk bin directory.
4. Enable boot start:
   ```
   [sudo] ./splunk enable boot-start
   ```

Enable boot-start on AIX to run Splunk software as a non-root user

1. Log into the AIX machine.
2. Become the root user, if able. Otherwise, you must prepend `sudo` to the following command examples. If you do not have `sudo` on your AIX instance, you must download, install, and configure it.
3. Create the user account that the Splunk software should run as. For example, if the `splunk` user should run the software:

   ```
   [sudo] mkuser splunk
   [sudo] chown -R splunk <Splunk directory>
   ```
4. Change to the Splunk bin directory.
5. Enable boot start and specify the `-user` flag with the user that the software should run as.
   ```
   [sudo] ./splunk enable boot-start -user <user that Splunk should run as>
   ```

Enable boot-start on MacOS

Splunk software automatically creates a script and configuration file in the directory `/System/Library/StartupItems` on the volume that booted your Mac. This script runs when your Mac starts, and automatically stops Splunk when you shut down your Mac.

If you want, you can still enable boot-start manually. You must either have root level permissions or use `sudo` to run the following command. You must have at least administrator access to your Mac to use `sudo`. If you installed Splunk software in a different directory, replace the example below with your instance location.

1. Log into your machine.
2. Open the Terminal app.
3. Change to the Splunk bin directory:

   ```
   cd /Applications/Splunk/bin
   ```
4. Enable boot start:

   ```
   [sudo] ./splunk enable boot-start
   ```

Enable boot-start on MacOS as a non-root user

1. Log into your machine.
2. Open the Terminal app.

3. Change to the Splunk bin directory:

```
cd /Applications/Splunk/bin
```

4. Enable boot start:

```
[sudo] ./splunk enable boot-start -user <user Splunk Enterprise should run as>
```

5. Open `/Library/LaunchItems/com.splunk.plist` for editing.
6. Locate the line that begins with `<dict>`.
7. Immediately after this line, add the following block of code:

```
    <key>UserName</key>
    <string><user Splunk Enterprise should run as></string>
```

8. Save the file and close it.

Changes take effect the next time you boot the machine.

Disable boot-start

If you want to stop Splunk software from running at machine boot time, run:

```
[sudo] $SPLUNK_HOME/bin/splunk disable boot-start
```

Disable boot-start on Windows

By default, Splunk starts automatically when you start your Windows machine. You can configure the Splunk processes (`splunkd` and `splunkweb`) to start manually from the Windows Services control panel.

Get more help on boot-start

To learn more about boot-start and how to enable it, see the following:

- The file `$SPLUNK_HOME/etc/init.d/README`
- The output from the `$SPLUNK_HOME/bin/splunk help boot-start` command on your Splunk software instance.

Run Splunk Enterprise as a systemd service

Splunk Enterprise 7.2.2 and higher provides support for `systemd` on Linux with an enhanced `enable boot-start` command that lets you automatically configure `systemd` to manage `splunkd` as a service.

What is systemd?

`systemd` is a system startup and service manager that is widely deployed as the default init system on most major Linux distributions. You can configure `systemd` to manage processes, such as `splunkd`, as services, and allocate system resources to those processes under `cgroups`.

systemd advantages

`systemd` offers the following general advantages:

- Enhanced parallel processing.

- Simplified configuration with standardized unit text files. No scripts required.
- Improved mechanism for expressing dependencies. For example, you can specify in the unit file that the network must be up before startup of the `splunkd` service occurs.

`systemd` offers these additional specific advantages for Splunk deployments:

- Start `splunkd` at boot.
- Monitor and manage `splunkd` service during runtime.
- Provides tools to debug and troubleshoot boot-time and service activities.
- Allows more control over plug-in monitoring tools that track the status of Splunk instances.
- Simplifies the set up of `cgroups` required for workload management in Splunk Enterprise. See Set up Linux for workload management in the *Workload Management* manual.

Configure systemd to manage splunkd

You can use either of the following two methods to configure `systemd` to manage `splunkd` as a service:

- Configure systemd using enable boot-start.
- Configure systemd manually.

If you configure systemd using enable boot-start, a Splunk service unit file is created automatically. No additional manual configuration is required.

System requirements

- To run `splunkd` as a `systemd` service requires one of the following supported Linux distributions:
  - RHEL 7, 8, and 9
  - CentOS 7 and 8
  - Ubuntu 16.04 LTS and later
  - Suse 12
- To configure `systemd` using `enable boot-start` requires Splunk Enterprise version 7.2.2 or later.
- To enable workload management in Splunk Enterprise under `systemd` requires `systemd` version 219 or higher. For more information, see Linux operating system requirements in the *Workload Management* manual.

Workload management supports Linux cgroups v1 only. If your Linux system has been upgraded to a version that runs cgroups v2 by default, you must revert your system to cgroups v1 to use Workload Management in Splunk Enterprise.

Permissions requirements

The `enable boot-start` command and `systemd` have the following permissions requirements:

- Non-root users must have super user permissions to configure `systemd` using `enable boot-start`.
- Non-root users must have super user permissions to run `splunk start|stop|restart` operations under `systemd`.

  For instructions on how to create a new user with super user permissions, see your Linux documentation.

Unprivileged users must use `sudo` to run `splunk start|stop|restart`. If you do not use `sudo` and attempt to run `splunk start|stop|restart` when managed by systemd, a prompt appears requesting authentication. For example:

```
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to manage system services or units.
Multiple identities can be used for authentication:
 1.  <username_1>
 2.  <username_2>
Choose identity to authenticate as (1-2): 2
Password:
==== AUTHENTICATION COMPLETE ===
```

Alternately, you can install polkit rules with the `enable boot-start` command to allow unprivileged users to run `start|stop|restart` operations under `systemd` without using `sudo`. For instructions, see Install polkit rules to elevate user permissions.

Unit file naming considerations

The `enable boot-start` command creates a `systemd` unit file named `Splunkd.service`. The unit file name is based on the `SPLUNK_SERVER_NAME` in `splunk-launch.conf`, which is set by default to `Splunkd`.

If for any reason you remove the `SPLUNK_SERVER_NAME` value from `splunk-launch.conf`, `enable boot-start` creates a unit file named `splunkd.service` (lower case "splunkd") and sets `SPLUNK_SERVER_NAME=splunkd` in the `splunk-launch.conf` file.

You can specify a different name of your choice for the unit file when you run `enable boot-start`. See Specify a different unit file name.

Configure systemd using enable boot-start

You can configure `systemd` to manage `splunkd` as a service using the `enable boot-start` command, as follows:

1. Log into the machine on which you want to configure `systemd` to manage `splunkd` as a service.
2. Stop `splunkd`.

   ```
   $SPLUNK_HOME/bin/splunk stop
   ```
3. If you previously enabled Splunk Enterprise to start at boot using the `enable boot-start` command, run `disable boot-start` to remove the `splunk init` script located in `/etc/init.d` and its symbolic links.

   ```
   [sudo] $SPLUNK_HOME/bin/splunk disable boot-start
   ```
   For instructions on how to reinstall the `splunk init` script, see Install splunk init script.
4. Run the `enable boot-start` command, specifying the `-systemd-managed`, `-user`, and `-group` parameters, as follows:

   ```
   [sudo] $SPLUNK_HOME/bin/splunk enable boot-start -systemd-managed 1 -user <username> -group
   <groupname>
   ```

   > Specifying -user and -group is optional but recommended. If you do not specify -user, the SPLUNK_OS_USER in splunk-launch.conf is used. If SPLUNK_OS_USER is not defined, the owner of the splunk binary is used.

   This installs the following `systemd` service unit file, named `Splunkd.service` by default, in `/etc/systemd/system`. To specify a different unit file name, use the `-systemd-unit-file-name` option. See Specify a different unit file name.

   ```
   #This unit file replaces the traditional start-up script for systemd
   #configurations, and is used when enabling boot-start for Splunk on
   #systemd-based Linux distributions.
   ```

```
[Unit]
Description=Systemd service file for Splunk, generated by 'splunk enable boot-start'
After=network.target

[Service]
Type=simple
Restart=always
ExecStart=/opt/splunk/bin/splunk _internal_launch_under_systemd
KillMode=mixed
KillSignal=SIGINT
TimeoutStopSec=360
LimitNOFILE=65536
SuccessExitStatus=51 52
RestartPreventExitStatus=51
RestartForceExitStatus=52
User=splunk
Group=splunk
Delegate=true
CPUShares=1024
MemoryLimit=<value>
PermissionsStartOnly=true
ExecStartPost=/bin/bash -c "chown -R splunk:splunk /sys/fs/cgroup/cpu/system.slice/%n"
ExecStartPost=/bin/bash -c "chown -R splunk:splunk /sys/fs/cgroup/memory/system.slice/%n"

[Install]
WantedBy=multi-user.target
```

The MemoryLimit value is set to the total system memory available in bytes when the service unit file is created. The MemoryLimit value will not update if the total available system memory changes. To update the MemoryLimit value in the unit file, you can manually edit the value or use the boot-start command to disable and re-enable systemd.

The following unit file properties are required. Do not change these values without appropriate guidance. Type=simple Restart=always ExecStart=$SPLUNK_HOME/bin/splunk _internal_launch_under_systemd Delegate=true This property is required for workload management. See Configure workload management.

Do not use the following properties. These properties can cause splunkd to fail on restart. RemainAfterExit=yes ExecStop

For more information, see Systemd unit file properties.

5. Start `splunkd`.

```
[sudo] $SPLUNK_HOME/bin/splunk start
```

This starts `splunkd` as a `systemd` service.

Under systemd, splunk start|stop|restart commands are mapped to systemctl start|stop|restart commands.

6. Verify that `splunkd` is running as a `systemd` service. For example:

```
$SPLUNK_HOME/bin/splunk status
splunkd is running (PID: 24772).
splunk helpers are running (PIDs: 24843 24857 24984 25032).
```

Alternatively, you can use systemctl status to check if the splunkd process is running. However, when using this command, a brief time lag can occur during which systemctl status shows "active" and splunk status shows

"splunkd is not running".

Configuring `systemd` to manage `splunkd` as a service creates CPU and Memory `cgroups` in these locations:

```
CPU: /sys/fs/cgroup/cpu/system.slice/Splunkd.service
Memory: /sys/fs/cgroup/memory/system.slice/Splunkd.service
```

7. For distributed deployments, repeat steps 1-8 on all search heads and indexers.

Additional options for enable boot-start

The `enable boot-start` command supports these additional options:

Install splunk init script

In version 7.2.2 and higher, the `enable boot-start` command adds a `-systemd-managed 0|1` option that controls whether to install the `splunk init` script in `/etc/init.d` or the `Splunkd.service` unit file in `/etc/systemd/system`.

To install the `splunk init` script, specify `-systemd-managed 0`:

```
$SPLUNK_HOME/bin/splunk enable boot-start -systemd-managed 0 -user <username>
Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
```

See Configure Splunk Enterprise to start at boot time.

In version 7.2.2 through 7.2.x, if you do not specify the -systemd-managed option, the enable boot-start command defaults to -systemd-managed 1 and installs the Splunkd.service unit file. In version 7.3.0 and later, this default behavior is reversed, and the enable boot-start command defaults to -systemd-managed 0 and installs the splunkinit file.

The init.d boot-start script is not compatible with RHEL 8 and higher.

Specify a different unit file name

The default `splunkd` unit file name is `Splunkd.service`. You can specify a different name for the unit file and update the `SPLUNK_SERVER_NAME` value in `splunk-launch.conf` using the `-systemd-unit-file-name` option. For example, to create a unit file with the name "splunk.service":

```
$SPLUNK_HOME/bin/splunk enable boot-start -systemd-managed 1 -systemd-unit-file-name splunk
Systemd unit file installed at /etc/systemd/system/splunk.service.
Configured as systemd managed service.
```

For more information, see Unit file naming considerations.

Install polkit rules to elevate user permissions

In version 8.1.1 and higher, the `enable boot-start` command adds an option to install polkit rules that allow non-root users to run `start`, `stop`, and `restart` operations under `systemd` without using `sudo`. Installing the polkit rules can reduce overhead for admins that must otherwise add unprivileged users to the `sudoers` file to run these operations under `systemd`.

To install polkit rules:

Run the `enable boot-start` command, specifying the `-create-polkit-rules` option, as follows:

```
./splunk enable boot-start -systemd-managed 1 -create-polkit-rules 1 -user <username>
```

If you previously ran `enable boot-start` and specified a different user, you must change the owner of `$SPLUNK_HOME` to the new user for whom you create the polkit rules. For example:

```
chown -R <username> $SPLUNK_HOME
```

Before you can install polkit rules using the create-polkit-rules option, you must install the Polkit library on your system if you have not already done so.

Configure systemd on a clean install

To configure `systemd` on a clean installation of Splunk Enterprise:

1. Expand the install package in an appropriate directory. For example:

   ```
   tar xvfz splunk_package_name.tgz -C /opt
   ```
2. Run `enable boot-start` to install the `Splunkd.service` unit file:

   ```
   sudo $SPLUNK_HOME/bin/splunk enable boot-start -systemd-managed 1 -user <username>
   ```

   When running enable boot-start for the first time after a clean install, Splunk Enterprise prompts you to accept the Splunk software license agreement. To automatically accept the license without prompt, specify the --accept-license flag with the command.
3. Start `splunkd`.

   ```
   sudo $SPLUNK_HOME/bin/splunk start
   ```
4. Verify that `splunkd` is running as a `systemd` service.

   ```
   $SPLUNK_HOME/bin/splunk status
   ```

Manage clusters under systemd

When managing an indexer cluster under `systemd`:

- You must use the `sudo` command to start, stop, and restart the cluster manager node or individual peer nodes using `splunk start|stop|restart` commands.
- You do not need `sudo` to perform a rolling restart using the `splunk rolling-restart cluster-peers` command, or to take a peer offline using the `splunk offline` command.

When managing a search head cluster under `systemd`:

- You must use the `sudo` command to start, stop, and restart cluster members using `splunk start|stop|restart` commands.
- You do not need `sudo` to perform a rolling restart using the `splunk rolling-restart shcluster-members` command, or to remove a cluster member using the `splunk remove shcluster-members` command.

Upgrade considerations for systemd

Upgrade from 8.0.x to 8.1

If you configured Splunk Enterprise version 8.0.x to run as a `systemd` service, upon upgrade to version 8.1, Splunk Enterprise adds the following properties to the `Splunkd.service` unit file:

```
User
Group
ExecStartPost
```

When Splunk Enterprise adds these unit file properties, it creates a new unit file that replaces the existing unit file `Splunkd.service`. It also renames the old unit file `Splunkd.service_<timestamp>`, which it saves for backup purposes only.

When upgrading directly from 7.3.x or lower to 8.1, Splunk Enterprise adds the Group property to the unit file.

Upgrade from 7.3.x or lower to 8.0

If you configured Splunk Enterprise version 7.3.x or lower to run as a `systemd` service, upon upgrade to version 8.0.x, on initial start, Splunk Enterprise modifies the existing `systemd` configuration as follows:

- It removes the `ExecStartPost` and `User` properties from the `Splunkd.service` unit file.
- It checks the `systemd` environment, identifies the `cgroup` path, and automatically sets permissions for the correct `cgroup` directories.

You must use `sudo splunk start` to perform the initial start of Splunk Enterprise after installing the version 8.0.0 upgrade tarball.

Using systemctl start to perform the initial start of Splunk Enterprise on upgrade to version 8.0.0 will fail.

For detailed information on upgrading Splunk Enterprise, see How to upgrade Splunk Enterprise in the *Installation Manual*.

Install your license

When you first install an instance of Splunk Enterprise, the instance includes a 60 day Enterprise Trial license that is enabled by default. This license allows you to try all of the features of Splunk Enterprise for 60 days, and to index up to 500 MB of data per day.

If you want to continue using Splunk Enterprise features after the 60 day trial expires, you must purchase an Enterprise license. Contact a Splunk sales rep to learn more.

If you do not install an Enterprise license after the 60 day trial expires, you can switch to Splunk Free. Splunk Free includes a subset of the features of Splunk Enterprise. It allows you to index up to 500 MB of data a day indefinitely. See About Splunk Free

For more information about Splunk licensing, read How Splunk licensing works in this manual.

To install and update your licenses using Splunk Web, see Install a license.

Change default values

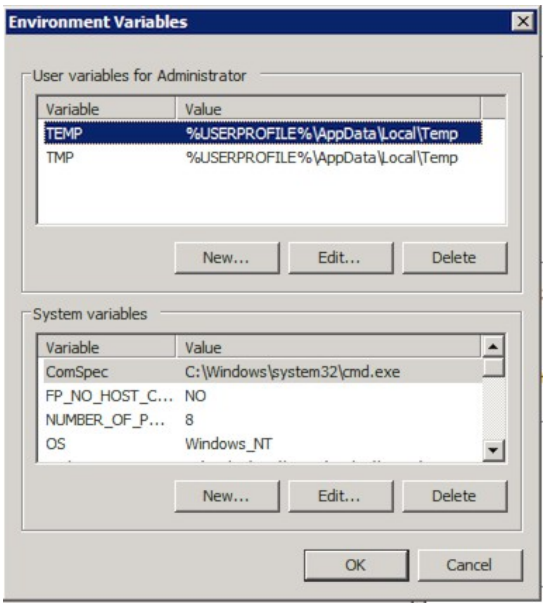
Before you begin configuring Splunk Enterprise for your environment, review the following default settings.

Set or change environment variables

Use operating system environment variables to modify specific default values for the Splunk Enterprise services.

- On *nix, use the `setenv` or `export` commands to set a particular variable. For example:
  ```
  # export SPLUNK_HOME = /opt/splunk02/splunk
  ```
  To modify the environment permanently, edit your shell initialization file, and add entries for the variables you want Splunk Enterprise to use when it starts up.

- On Windows, use the `set` environment variable in either a command prompt or PowerShell window:
  ```
  C:\> set SPLUNK_HOME = "C:\Program Files\Splunk"
  ```
  To set the environment permanently, use the "Environment Variables" window, and add an entry to the "User variables" list.



Several environment variables are available:

Environment variable	Purpose
SPLUNK_HOME	The fully qualified path to the Splunk Enterprise installation directory.
SPLUNK_DB	The fully qualified path to the root directory that contains the Splunk Enterprise indexes.
SPLUNK_BINDIP	The host IP address that Splunk Enterprise should bind to on startup. On hosts with multiple IP addresses, this is used to limit accepted connections to one IP address.
SPLUNK_OS_USER	Tells Splunk Enterprise to assume the credentials of the user you specify, regardless of what user you started it as. For example, if you specify the SPLUNK_OS_USER 'splunk' but start Splunk Enterprise as root, the system adopts the privileges of the 'splunk' user, and any files written by those processes will be owned by the 'splunk' user.
SPLUNK_SERVER_NAME	The name of the splunkd service (on Windows) or process (on *nix). Do not set this variable unless you know what you are doing.
SPLUNK_WEB_NAME	The name of the splunkweb service (on Windows) or process (on *nix). Do not set this variable unless you know what you are doing.

Note: You can set these environment variables in the splunk-launch.conf or web.conf file. This is useful when you run more than one Splunk software instance on a host. See splunk-launch.conf.

Change network ports

Splunk Enterprise configures default TCP ports during installation:

- **The HTTP/HTTPS port.** This port provides the socket for Splunk Web. It defaults to 8000.
- **The appserver port.** 8065 by default.
- **The management port.** This port is used to communicate with the splunkd daemon. Splunk Web talks to splunkd on this port, as does the command line interface, and any distributed connections from other servers. This port defaults to 8089.
- **The KV store port.** 8191 by default.

The default network ports are recommendations, and might not represent what your Splunk Enterprise instance is using. During the Splunk Enterprise installation, if any default port is detected as in-use, you are prompted to provide alternative port assignments.

Splunk instances that are receiving data from forwarders must be configured with a receiver port. The receiver port only listens for incoming data from forwarders. Configuration of the receiver port does not occur during installation. For more information, see Enable a receiver in the Forwarding Data Manual.

Use Splunk Web

To change the ports from their installation settings:

1. Log into Splunk Web as the admin user.
2. Click **Settings**.
3. Click **Server settings**.
4. Click **General settings**.
5. Change the value for either **Management port** or **Web port**, and click **Save**.

Use Splunk CLI

To change the port settings using the Splunk CLI, use the CLI command `set`. For example, this sets the Splunk Web port to 9000:

```
splunk set web-port 9000
```

This command sets the splunkd port to 9089:

```
splunk set splunkd-port 9089
```

Change the default Splunk server name

The Splunk server name setting controls both the name that is displayed within Splunk Web, and the name that is sent to other Splunk Servers in a distributed deployment. The name is chosen from either the DNS or IP address of the Splunk Server host by default.

Use Splunk Web

To change the Splunk server name:

1. Log into Splunk Web as the admin user.
2. Click **Settings**.
3. Click **Server settings**.
4. Click **General settings**.
5. Change the value for **Splunk server name**, and click **Save**.

Use Splunk CLI

To change the server name using the CLI, use the `set servername` command. For example, this sets the server name to *foo*:

```
splunk set servername foo
```

Set minimum free disk space

The minimum free disk space setting controls how low storage space in the datastore location can fall before Splunk software stops indexing. Splunk software resumes indexing when available space exceeds this threshold.

Use Splunk Web

To set minimum free storage space:

1. Log into Splunk Web as the admin user.
2. Click **Settings**.
3. Click **Server settings**.
4. Click **General settings**.
5. Change the value for **Pause indexing if free disk space (in MB) falls below**, and click **Save**.

Use Splunk CLI

To change the minimum free space value using the CLI, use the `set minfreemb` command. For example, this sets the minimum free space to 2000 MB:

```
splunk set minfreemb 2000
```

Set the default time range

The default time range for ad hoc searches in the Search & Reporting App is set to **Last 24 hours**. A Splunk Enterprise administrator can set the default time range globally, across all apps. Splunk Cloud Platform customers cannot configure this setting directly. The setting is stored in $SPLUNK_HOME/etc/apps/user-prefs/local/user-prefs.conf file in the [general_default] stanza.

This setting applies to all Search pages in Splunk Apps, not only the Search & Reporting App. This setting applies to all user roles.

> This setting does not apply to dashboards.

Use Splunk Web

1. Log into Splunk Web as the admin user.
2. Click **Settings**.
3. Click **Server settings**.
4. Click **Search Preferences**.
5. From the **Default search time range** drop-down, select the time that you want to use and click **Save**.

Time range settings in the ui-prefs.conf file

You might already have a time range setting defined in the ui-prefs.conf file for a specific application or user. The settings in the ui-prefs.conf file take precedence over any settings that you make to the global default time range using Splunk Web.

However, if you want to use the global default time range for all users and applications, consider removing the settings that you have in the ui-prefs.conf file. See ui-prefs.conf.

Other default settings

The Settings screen offers additional pages with default settings for you to change. Explore the screen to see the range of options.

Bind Splunk to an IP

By default, the Splunk Enterprise services are bound to IP address 0.0.0.0, meaning all available IP addresses on the host machine. You can force Splunk Enterprise to bind all service ports to a specified IP address.

Changing the IP address applies to the Splunk daemon (splunkd) services:

- TCP port 8089 (by default)

- Splunk Web port 8000 (by default)
- Any port that has been configured as for:
    - SplunkTCP inputs
    - TCP or UDP inputs
    - HEC inputs
- App Server port 8065 (by default)
- KV Store port 8191 (by default)

To bind the Splunk Web process (splunkweb) to a specific IP, use the `server.socket_host` setting in web.conf.

Temporarily change the IP address

To make this a temporary change, use the environment variable `SPLUNK_BINDIP=<ipaddress>` to set an IP address before starting Splunk Enterprise services.

Permanently change the IP address

To permanently change the default IP address for a host machine, update the `$SPLUNK_HOME/etc/splunk-launch.conf` to include the `SPLUNK_BINDIP` attribute and `<ipaddress>` value.

For example, to bind Splunk ports to 127.0.0.1 (for local loopback only), `splunk-launch.conf` should read:

```
# Modify the following line to suit the location of your Splunk install.
# If unset, Splunk will use the parent of the directory this configuration
# file was found in
#
# SPLUNK_HOME=/opt/splunk
SPLUNK_BINDIP=127.0.0.1
```

Important: The `mgmtHostPort` attribute in `web.conf` has a default value of `0.0.0.0:8089`. If you use `SPLUNK_BINDIP` to enforce a different IP address, you must also change `mgmtHostPort` to use the same IP address.

For example, if you change the `splunk-launch.conf`:

```
SPLUNK_BINDIP=10.10.10.1
```

you must also change the `web.conf` to IP address to match:

```
mgmtHostPort=10.10.10.1:8089
```

See web.conf for more information on the `mgmtHostPort` attribute.

IPv6 considerations

The `mgmtHostPort` setting in web.conf accepts IPv6 addresses if they are enclosed in square brackets. If you configure splunkd to only listen on IPv6, you must update the `mgmtHostPort` to use `[::1]:8089` instead of `127.0.0.1:8089`. See "Configure Splunk for IPv6".

Configure Splunk Enterprise for IPv6

Implementing IPv6 support for Splunk Enterprise requires familiarity with configuration files, the ports used by Splunk software, and data input configurations.

IPv6 platform support

Splunk Enterprise IPv6 support depends on the operating system that the Splunk software or a Universal Forwarder is installed on. For a table of supported OS platforms, see Supported Operating Systems in the *Installation Manual*.

Unsupported operating systems

IPv6 support is unavailable on the AIX operating system.

Splunk Enterprise and IPv6 functionality

The IPv6 configuration in Splunk Enterprise is disabled by default. Before enabling IPv6 support, determine what functionality you want to access with an IPv6 address.

Functionality	Details
Allow the Splunk Enterprise software to listen on the Splunk management port and KVStore port over IPv6.	See Configure Splunk Enterprise to listen on an IPv6 network.
Allow access to Splunk Web over IPv6.	See Configure Splunk Web to listen on IPv6.
Configure a single IPv6 listener for inbound network traffic.	See Configure an IPv6 listener on one network input.
Use a Splunk CLI command to access Splunk Enterprise over IPv6.	See Use the Splunk CLI over IPv6.
Configure a Splunk Forwarder to send data to Splunk Enterprise over IPv6.	See Forwarding data over IPv6.
Configure Splunk Enterprise distributed search for outbound communication over IPv6.	See Distributed search configuration for IPv6.
Configure IPv6 support with single sign-on.	See IPv6 support with single sign-on (SSO).
Change how Splunk Enterprise prioritizes IPv4 and IPv6 communication behavior.	See Change the prioritization of IPv4 and IPv6 communications.

Configure Splunk Enterprise to listen on an IPv6 network

Use the steps below to configure Splunk Enterprise to listen on the Splunk management port and KVStore port over IPv6.

1. Using a shell prompt, go to the folder `$SPLUNK_HOME/etc/system/local`.
2. Edit the server.conf file.
3. Under the `[general]` stanza, add the line `listenOnIPv6 = yes`.
4. Save the changes.
5. Restart the Splunk Enterprise instance.
6. Verify that the service is listening on the appropriate port using `netstat` or a similar utility.
7. (Optional) Change the prioritization of IPv4 and IPv6 communications. See Change the prioritization of IPv4 and IPv6 communications.

After IPv6 is enabled on the Splunk management port, any ports previously defined in the inputs.conf will also listen on IPv6.

Configure Splunk Web to listen on IPv6

Use the steps below to configure Splunk Web to accept communications over IPv6.

1. Using a shell prompt, go to the folder `$SPLUNK_HOME/etc/system/local`.
2. Edit the web.conf file.
3. Under the `[settings]` stanza, add the line `listenOnIPv6 = yes`.
4. Save the changes.
5. Restart the Splunk Enterprise instance.
6. Verify that the service is listening on the appropriate port using `netstat` or a similar utility.
7. Use a web browser to connect to Splunk Web. For example, http://[2620:70:8000:c205::129]:8000.

Change the prioritization of IPv4 and IPv6 communications

After you configure Splunk Enterprise to support IPv6, the services will listen on both IPv4 and IPv6 ports for communication. To prioritize or limit ports to one IP protocol, review and change the `connectUsingIpVersion` setting in server.conf.

If you configure both Splunk Enterprise and Splunk Web to listen only on IPv6, you must change the web.conf setting mgmtHostPort from 127.0.0.1:8089 to [::1]:8089.

Configure an IPv6 listener on one network input

The inputs.conf stanzas `[tcp]`, `[udp]`, `[tcp-ssl]`, `[splunktcp]`, `[splunktcp-ssl]` will all accept the `listenOnIPv6` setting. The `listenOnIPv6` setting for a specific input takes precedence over the configuration applied in `server.conf`.

To enable IPv6 on a single input, add the setting `listenOnIPv6 = yes` to the input stanza defined in an inputs.conf file.

1. Using a shell prompt, go to the folder `$SPLUNK_HOME/bin`.
2. Use the `btool` command to identify the location of the inputs.conf you want to modify. For example, to find a splunktcp stanza type:
   ./splunk btool inputs list --debug | grep splunktcp
3. Go to the location of the inputs.conf file found with `btool`.
4. Edit the inputs.conf file.
5. Under the input stanza add the line: `listenOnIPv6 = yes`.
6. Save the changes.
7. Restart the Splunk Enterprise instance.
8. Verify that the service is listening on the appropriate port using `netstat` or a similar utility.

Use the Splunk CLI over IPv6

You can use the Splunk CLI to communicate to a Splunk Enterprise instance over IPv6. The remote instance must be configured to listen for IPv6 on the Splunk management port. See Configure Splunk Enterprise to listen on an IPv6 network.

To access Splunk Enterprise from the CLI, use the `-uri` command with an IPv6 address, for example, `./splunk display app -uri "https://[2620:70:8000:c205::129]:8089"`

You can pre define the destination address, use the `$SPLUNK_URI` environment variable in your shell prompt. See Change your default URI value. For more CLI commands, see Get help with the CLI.

If you use link-local addressing on IPv6 (seen as an IPv6 address beginning with fe80:), some of the CLI commands can fail. This failure is due to the OS-level implementation of IPv6 with link-local addresses, and not Splunk software.

Forwarding data over IPv6

To enable a **forwarder** to send data to another Splunk Enterprise instance over IPv6, edit the outputs.conf and update the `server =` parameter with an IPv6 address formatted as `[host]:port`, for example, `server = [2002:4721:93f0::e956]:9997`. The outputs.conf stanzas `[tcpout], [tcpout-server], [syslog]` accepts IPv6 addresses.

Distributed search configuration for IPv6

The `servers` setting in distsearch.conf can include IPv6 addresses in the standard `[host]:port` format. The remote instance must be configured to listen for IPv6 on the Splunk management port. See Configure Splunk Enterprise to listen on an IPv6 network.

IPv6 support with single sign-on

If you use IPv6 with single sign-on (SSO), don't use the square bracket notation for any IPv6 address referenced in the `trustedIP` setting, as shown in the following example. The square bracket notation exception applies when setting `trustedIP` in web.conf or server.conf.

```
[settings]
mgmtHostPort = [::1]:8089
startwebserver = 1
listenOnIPv6=yes
trustedIP=2620:70:8000:c205:250:56ff:fe92:1c7,::1,2620:70:8000:c205::129
SSOMode = strict
remoteUser = X-Remote-User
tools.proxy.on = true
```

For more information on SSO, see Configure Single Sign-on in the *Securing Splunk Enterprise* manual.

Secure your configuration

If you haven't already, this is a good time to make sure that Splunk and your data are secure. Taking the proper steps to secure Splunk reduces the attack surface and mitigates the risk and impact of most vulnerabilities.

Some key actions you should take after installation:

- Set up users and roles. You can configure users using Splunks native authentication and/or use LDAP to manage users. See About user authentication
- Set up certificate authentication (SSL). Splunk ships with a set of default certificates that should be replaced for secure authentication. We provide guidelines and further instructions for adding SSL encryption and authentication and Configure secure authentication.

The Securing Splunk Enterprise manual provides more information about ways you can secure Splunk. Including a checklist for hardening your configuration. See Securing Splunk Enterprise for more information.

Share performance and usage data in Splunk Enterprise

Splunk Inc. collects critical data so that we can enhance the value of your investment in Splunk software.

We use this data to optimize your deployment, prioritize our features, improve your experience, notify you of patches, and develop high quality product functionality.

Changes in version 9.1.0

There are minor changes to Splunk data collection practices in version 9.1.0, mainly for the Splunk Assist service. Data collection remains on by default. For more information on why Splunk changed its policy to enable the collection of usage data, see the 8.0 version of this topic.

The support usage data that Splunk collects for Splunk Assist and for telemetry are the same. The targets for these data sources, however, are different. You might need to update any firewall settings that you have before you can use Splunk Assist, even though the Splunk platform can send support usage data back to Splunk.

You can still opt out of data sharing at any time, but if you do, you cannot use the Splunk Assist service, which requires that data sharing is active. See How to opt out.

To learn more about Splunk Assist, see About Splunk Assist in the *Monitoring Splunk Enterprise Manual*.

Benefits of sharing data with Splunk

When you share data with Splunk Inc., you receive the following benefits:

- **Improved product quality.** By collecting accurate information about the topology decisions and deployment scale used by our customers, we can replicate those topology configurations and scale in our internal testing, helping us improve your product experience.
- **Timely notification of known bugs, version incompatibilities, and configuration issues.** When you share data about the product versions you have deployed, we can provide accurate messages and support to help you with bugs, upgrade tasks, version compatibility problems, and other configuration issues you might experience.
- **Relevant feature enhancements.** We prioritize what features to develop and enhance first based on the features customers use the most. By sharing your data, you influence these data-driven decisions in favor of the features you use at your organization.
- You can use the Splunk Assist service to monitor your deployment in accordance with Splunk best practices for security, performance, and configuration.

For more information, see How Splunk uses the data it collects.

What data Splunk collects

The following table summarizes the data that your Splunk platform deployment sends to Splunk when you enable data collection. Follow the links to see examples of this data.

Type of data	Description	Examples
Aggregated usage data	Includes features used, deployment topology, and performance metrics in both the platform and apps. This data is not associated with your license ID. You must enable Aggreated usage data to use the Splunk Assist service.	Aggregated usage data examples App usage data examples

Type of data	Description	Examples
Support usage data	Support usage data is the same as the aggregated usage data, but the license ID remains associated with your data when it reaches Splunk Inc. You must enable support usage data to use the Splunk Assist service.	Aggregated usage data examples App usage data examples
License usage data	Includes your license ID, active license group and subgroup, total license stack quota, total license pool consumption, license stack type, license pool quota, license pool consumption.	License usage data examples
Software version data	Includes the version of Splunk Enterprise and of each installed app, along with relevant metadata about deployment architecture.	Software version data examples

Splunk does not collect the contents of your indexed data.

Some cloud and hybrid products modify the kinds of data that Splunk collects. When that happens, a separate agreement or notification states how the data collection differs for that product.

For instructions on how to view the data that your deployment collects and sends to Splunk, see View what data is sent from your deployment.

Examples of data sent to Splunk

Aggregated usage, support usage, and license usage data is sent to Splunk as a JSON packet that includes information like the component name and deployment ID, in addition to the data for the specific data collection component. The deploymentID is unique to a deployment and does not change on upgrade or even after uninstall and reinstall of Splunk Enterprise on the same machine.

Here is an example of a complete JSON packet:

```
{
component: deployment.app
   data: { [-]
     enabled: true
     host: 878e7b21bf98580dbdb4ed3baf6c35d78aa5bc3d3c824eb8714a313c
     name: search
     version: 8.0.0
   }
   date: 2019-09-23
   deploymentID: d6d8e776-a8d3-5467-a03b-375577646cbb
   executionID: 2FC293C59049AC0D44B677D3A9D786
   timestamp: 1569294102
   transactionID: 4E1CFC7E-BE9F-355D-7DDE-D4F8D5E4852D
   version: 3
   splunkVersion: 8.1.2
   visibility: anonymous,support
}
```

The following tables list the component names, descriptions, and an example of what data is collected for that component. For ease of use, the examples for aggregated usage and license data show examples of only the `data` field from the JSON object.

Aggregated usage data examples

The following example demonstrates the data sent to Splunk when sharing of aggregated usage data is enabled.

Component	Description	
app.RapidDiag.cliAccessMetrics	RapidDiag CLI interface usage statistics.	{ [-] app: splunk_rapid_diag component: app.RapidDiag.cliAccessMetrics data: { [-] action: 'run' count: 2 mode: 'templates' result: 0 } deploymentID: 654b5421-eec2-5229-9fc6-5f065 eventID: 8BEB3B43-FC9E-47F3-8FFF-BA6E1D2CF4 executionID: C7212C53-51C7-4CB5-9316-1A3F68 optInRequired: 3 timestamp: 1605611221 type: aggregate visibility: [[-] anonymous support]
app.RapidDiag.uiAccessMetrics	RapidDiag UI interface usage statistics.	{ [-] app: splunk_rapid_diag component: app.RapidDiag.uiAccessMetrics data: { [-] count: 1 status: 200 uri_path: /en-GB/app/splunk_rapid_diag/da user: 8c6976e5b541 } deploymentID: 654b5421-eec2-5229-9fc6-5f065 eventID: 4A5E61B6-C5C8-47F7-A6C9-AA4409E3AE executionID: 07237CFC-6663-44D6-9F12-82D273 optInRequired: 3 timestamp: 1605540721 type: aggregate visibility: [[-] anonymous support]
app.RapidDiag.executionMetrics	RapidDiag task execution statistics.	{ [-] app: splunk_rapid_diag component: app.RapidDiag.executionMetrics data: { [-] count: 10 metricName: dd1cd3d60a28 status: Success type: collector } deploymentID: 654b5421-eec2-5229-9fc6-5f065 eventID: AA2EA083-F71C-473A-B19D-0C0993FCB5 executionID: B0FFB679-2745-4AA6-AF99-71999E optInRequired: 3 timestamp: 1605611641 type: aggregate visibility: [[-] anonymous support] app: splunk_rapid_diag

Component	Description	
		component: app.RapidDiag.executionMetrics data: { [-] count: 10 name: Slow search performance status: Success type: task } deploymentID: 654b5421-eec2-5229-9fc6-5f065 eventID: A6253B1F-7C26-4656-AE8F-848AC12578 executionID: B0FFB679-2745-4AA6-AF99-71999E optInRequired: 3 timestamp: 1605611641 type: aggregate visibility: [[-] anonymous support]
app.session.coreLibrarySettings.save	Tracks if certain core library settings are toggled on or off.	{ [-] component: app.session.coreLibrarySettings. data: { [-] app: search page: core_library_settings setting: enable_jQuery2 value: False } deploymentID: 942a8692-dce5-9b6f-4bd4-f4811 eventID: 899f8692-dce5-9b6f-4bd4-f4811c2032 experienceID: a6c7710b-6822-394e-3292-812ee optInRequired: 3 timestamp: 1617218044 userID: 40babbddf86516c5864e524a6e3b66f38ca version: 4 visibility: anonymous,support }

app.session.createNewDashboardDialog.interact	General telemetry collected when a new dashboard is created.	{ [-]
		"component": "app.session.createNewDashboardDi
		"data": {
		"action": "createNewDashboard",
		"editId": true,
		"hasDescription": false,
		"dashboardType": "udf",
		"layout": "absolute",
		"sharing": "user",
		"status": "success",
		"app": "search",
		"page": "dashboards"
		},
		}

app.session.dashboard.load	Dashboard characteristics, generated as session data when a dashboard loads.	{ [-] app: search dashboard: { [-] autoRun: false hideAppBar: false hideChrome: false hideEdit: false hideExport: false hideFilters: false

Component	Description	
		hideSplunkBar: false hideTitle: false isScheduled: false isVisible: true numCustomCss: 0 numCustomJs: 0 refresh: 0 submitButton: false theme: light version: 1.0 isDeprecatedXMLDashboard: true } elementTypeCounts: { [-] area: 1 column: 1 line: 1 singlevalue: 8 statistics: 10 } formInputTypeCounts: { [-] } layoutType: row-column-layout numElements: 21 numFormInputs: 0 numPanels: 21 numPrebuiltPanels: 0 numSearches: 21 page: network_insights searchTypeCounts: { [-] inline: 21 } }
app.session.dashboard.interact	Whether a user pressed Cancel or Continue for the URL warning modal.	{ [-] "component":"app.session.dashboard.interact", "data":{ "type":"urlWarningModal", "action":"cancel", "app":"search", "page":"giulia_sxml" },
app.session.dashboard.error	If an asynchronous error occurred in a CustomJS script used by a dashboard.	{ [-] data: { [-] app: search errorType: customJSError page: kieran123 }
app.session.dashboard.telemetry	General telemetry collected when adding and configuring dashboard elements.	{ [-] "component": "app.session.dashboard.telemetry" "data": { "pageAction": "scheduledExport "success": true, "enabledInitially": false, "enabledAtSave": true, "cronSchedule": "0 18 * * *", "emailCountTo": 1, "emailCountCC": 0, "emailCountBCC": 0,

Component	Description	
		"emailSubjectLength": 22, "emailMessageLength": 17, "includeLinkInitially": false, "includeLinkAtSave": false, "app": "search", "page": "dashboards" } }
app.session.dataactions.interact	User interactions in the dataactions UI.	{ [-] component:app.session.dataactions.interact data: { [-] action: save app: $SPLUNK_PLATFORM editType: new externalDestinationCount: 0 name: 9dd8c74a33ee89cb4fbe82deee2273ec6b8 page: manager/search/ingest_rulesets ruleCount: 1 ruleCountsByAction: { [-] filter: 1 } sourcetype: 65935aef8944a30f5046ba0159cfa } deploymentID: 825dc0d6-5430-5ef8-9b69-2c54a eventID: 7f49d8ee-5b9c-c401-2cd0-81bc811a25 experienceID: 9b66912a-86df-efa0-e099-ee9cc optInRequired: 3 splunkVersion: 9.0.0 timestamp: 1652730582 userID: d2bb23947441c280c5cf8fee0df81614294 version: 4 visibility: anonymous,support }

app.session.dataactions.load	Number of rulesets and type of deployment.	{ [-]
		component: app.session.dataactions.load
		data: {
		rulesetCount: 2
		deploymentType: cluster-master
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6
		executionID: F0AE995E8653D768A360E73BE3F544
		timestamp: 1540570045
		transactionID: 89F7329E-86AD-BBFD-034F-2090
		version: 3
		visibility: anonymous,support
		}

app.session.datainteractions.load	Apps installed per Splunk instance.	{ [-] data: { [-] rulesetCount: 2 deploymentType: cluster-master } date: 2018-10-26 deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6 executionID: F0AE995E8653D768A360E73BE3F544 timestamp: 1540570045 transactionID: 89F7329E-86AD-BBFD-034F-2090

Component	Description	
		version: 3 visibility: anonymous,support }
app.session.globalBanner.error	Unexpected error responses from GET/POST requests to the global banner endpoint, and the status code.	{ [-] app: $SPLUNK_PLATFORM page: manager/launcher/global_banner responseText: {"messages":[{"type":"ERROR status: 400 }
app.session.globalBanner.interact	Tracks when a user clicks a banner link.	{ [-] action: link click app: $SPLUNK_PLATFORM page: manager/launcher/global_banner }
app.session.html_dashboard	Count the number of HTML dashboards in the Splunk Enterprise instance.	{ [-] component: app.session.html_dashboard data: { [-] app: search page: jquery_staging count: 21 } deploymentID: 942a8692-dce5-9b6f-4bd4-f4811 eventID: 899f8692-dce5-9b6f-4bd4-f4811c2032 experienceID: a6c7710b-6822-394e-3292-812ee optInRequired: 3 timestamp: 1617218044 userID: 40babbddf86516c5864e524a6e3b66f38ca version: 4 visibility: anonymous,support }
app.session.html_dashboard.load	Track the number of times an HTML dashboard is loaded.	{ [-] component: app.session.html_dashboard.load data: { [-] app: search page: network_insights } deploymentID: 942a8692-dce5-9b6f-4bd4-f4811 eventID: 899f8692-dce5-9b6f-4bd4-f4811c2032 experienceID: a6c7710b-6822-394e-3292-812ee optInRequired: 3 timestamp: 1617218044 userID: 40babbddf86516c5864e524a6e3b66f38ca version: 4 visibility: anonymous,support }
app.session.metrics.interact	Track the type of filter the user set on a chart.	{ [-] accessor: METRICS action: SERIES_FILTER_ADD app: search chartType: line context: analysis customInfo: { [-] app: metrics-analysis commitHash: 5b0687f037c02ab76c3adc2391e version: 2.28.0 }

Component	Description	
		numCustomFilters: 1 numFilters: 1 numHostFilters: 0 numIndexFilters: 0 numIndexRefLines: 0 numMeasures: 1 numSeries: 1 numSourceTypeFilters: 0 numStaticRefLines: 0 numTimeRangeRefLines: 0 numTimeShiftRefLines: 0 page: analytics_workspace seriesHasSplit: false seriesId: 264aa232-2d23-47c0-8a0e-9ee6414 type: view/UPDATE_SERIES value: { [+] } viewId: v27f16248-701c-4fe2-b79e-27462e15 }
app.session.metrics.process	De-identified chart configuration data related to the queries sent by workspace charts.	{{ [-] action: EXECUTE_QUERY app: search context: analysis customInfo: { [-] app: metrics-analysis commitHash: 50bd435d736fd97bb0a7125221b splunkVersion: 8.1.0 version: 2.28.0 } elapsed: 232 page: analytics_workspace query: { [-] series: [[-] { [-] accessor: METRICS aggregation: avg axis: left filters: 1 refLines: [[-] { [-] aggregation: max includeValueInLabel: true timeRange: null timeShift: -1d type: indexDataAggregation }] span: 10s split: { [-] limit: 5 type: top } timeshift: -30m }] timeRange: { [-] earliest: 1596751969.139 latest: 1596755569.139 }

Component	Description	
		} requestId: 00961132-3d15-45a2-9d69-0624b1 status: completed viewId: v69289f5f-c33c-4161-9281-53724a9a }
app.session.page.interact	Tracks user interactions with search, reports, alerts, data models, tags, lookups, and search macros.	{ [-] action: Edit Permissions - Save app: search custom: { [+] } page: dataset }
app.session.page.load	Tracks loads and whether web services are supported, generated as session data when a page loads.	{ [-] allowWebService: true app: $SPLUNK_PLATFORM page: manager/search/adddata }
app.session.pageview	Page view session data, generated whenever a user visits a new page.	{ [-] app: launcher page: home }
app.session.pivot.interact	Changes to pivots, generated as session data when a user makes a change to a pivot.	{ [-] app: search context: pivot eventAction: change eventCategory: PivotEditorReportContent eventLabel: Pivot - Report Content eventValue: { [-] transient: true } numAggregations: 1 numColumnSplits: 0 numCustomFilters: 0 numRowSplits: 1 page: pivot reportProps: { [-] display.general.type: visualizations display.statistics.show: 1 display.visualizations.charting.chart: display.visualizations.charting.chart.r display.visualizations.charting.gaugeCo display.visualizations.charting.legend. display.visualizations.show: 1 display.visualizations.singlevalue.rang display.visualizations.singlevalue.tren display.visualizations.type: charting earliest: -24h@h latest: now windowedEarliest: 2019-09-23T03:00:00.0 windowedLatest: 2019-09-24T03:58:52.000 } }
app.session.pivot.load	Pivot characteristics, generated as session data when a pivot loads.	{ [-] app: search context: pivot

Component	Description	
		eventAction: load eventCategory: PivotEditor eventLabel: Pivot - Page numAggregations: 1 numColumnSplits: 0 numCustomFilters: 0 numRowSplits: 1 page: pivot reportProps: { [-] display.general.type: visualizations display.statistics.show: 1 display.visualizations.charting.chart: display.visualizations.charting.chart.r display.visualizations.charting.gaugeCo display.visualizations.charting.legend. display.visualizations.show: 1 display.visualizations.singlevalue.rang display.visualizations.singlevalue.tren display.visualizations.type: charting earliest: -24h@h latest: now windowedEarliest: 2019-09-23T03:00:00.0 windowedLatest: 2019-09-24T03:58:52.000 } }
app.session.roles.srchFilter	Event actions on the authoritzation/roles page of Splunk Web	{ [-] app: $SPLUNK_PLATFORM context: authorization/roles eventAction: CreateEditRole eventCategory: SrchFilterInRoles eventLabel: Search Filter in role - adm eventValue: * page: manager/launcher/authorization/ro }
app.session.rum.mark	Track performance of the first meaningful paint for the global banner settings page and the view itself, when enabled.	{{ [-] app: $SPLUNK_PLATFORM hero: Global Banner Settings - First mean page: manager/launcher/global_banner sourceLocation: Global Banner Settings - timeSinceOrigin: 6917.774999994435 transactionId: 2da6cc30-6880-11ea-a7ac-5f }
app.session.rum.measure	Track performance of the first meaningful paint for the global banner settings page and the view itself, when enabled.	{ [-] app: $SPLUNK_PLATFORM duration: 6917.774999994435 fromSourceDurations: { [+] } fromSourceLocation: origin hero: Global Banner Settings - First mean page: manager/launcher/global_banner timeSinceOrigin: 6917.774999994435 toSourceLocation: Global Banner Settings transactionId: 2da6cc30-6880-11ea-a7ac-5f }
app.session.search.interact	Search page interactions, session data generated by	{ [-] app: search

Component	Description	
	each user interaction with the search page.	context: search eventAction: submit eventCategory: CreateReportDialog eventLabel: Search App - Actions eventValue: success page: search reportProps: { [-] dispatch.sample_ratio: 1 display.events.table.sortDirection: asc display.general.type: statistics display.page.search.mode: smart display.prefs.events.offset: 0 display.prefs.statistics.offset: 0 display.statistics.format.0: display.statistics.format.0.colorPalett display.statistics.format.0.colorPalett display.statistics.format.0.field: display.statistics.format.0.scale: display.statistics.format.0.scale.thres display.statistics.sortColumn: Number c display.statistics.sortDirection: asc display.visualizations.charting.chart: earliest: -24h@h latest: now workload_pool: } }
app.session.session_start	Session data generated when a user is first authenticated. Contains the deploymentID (identifier for deployment), eventID (identifier for this specific event), experienceID (identifier for this session), userID (hashed username), data.guid (GUID for instance serving the page).	{ [-] app: launcher browser: Chrome browserVersion: 68.0.3440.106 device: Linux x86_64 guid: 0C4C7528-375A-4DA5-ABF8-09189051BB5 locale: en-US os: Linux osVersion: not available page: home splunkVersion: 8.0.0 }
app.session.spotlightSearch.redirect	Tracks selections and redirects from settings menu interactive search bar to settings menu pages.	{ [-] component: app.session.spotlightSearch.redi data: { [-] app: $SPLUNK_PLATFORM page: manager/system/saved/searches redirectPageName: Event types redirectURL: /manager/system/saved/eventt searchTextLength: 5 } deploymentID: 0bb68240-cb80-562f-a4d9-3883b eventID: c5980f8c-cbac-f07b-72d0-87c044d1e6 experienceID: c460b442-278b-d554-5c1e-241a8 optInRequired: 3 splunkVersion: 20230402 timestamp: 1680890849 userID: 165b54f7e4e1b3dab838ea6cb45c1729e88 version: 4 visibility: anonymous,support

Component	Description	
		}
app.session.tableUI.interact	Tracks interactions on the Table UI page.	{ [-] action: create_table_view app: search location: datasets listing page page: datasets }
app.session.template.load	Tracks the number of times users access HTML template files that Splunk Enterprise no longer uses.	{ [-] app: asdf page: search template: test-example }

app.session.udf.telemetry	General telemetry collected on visualization usage and settings.	{ [-]
		"component": "app.session.udf.telemetry",
		"data": {
		"pageAction": "dashboard.initialize",
		"metadata": {},
		"udfVersion": "20.3.1",
		"definition": {
		"visualizations": {
		"viz_2aae822a03cb3f7c58a43c046
		"type": "viz.column",
		"options": {},
		"titleLength": 13,
		"descriptionLength": 26
		},
		"viz_3a1a36fecbc0b5b46b5cb8777
		"type": "viz.singlevalueic
		"options": {
		"showValue": false,
		"icon": true
		}
		},
		"viz_cf5bd9532cfe6d8619132f9bb
		"type": "viz.rectangle"
		},
		"viz_36b6e66b1475b0e0677676b94
		"type": "viz.singlevalue",
		"options": {},
		"titleLength": 13,
		"descriptionLength": 24
		},
		"viz_f3479a853843e0e72405cc99f
		"type": "viz.text",
		"options": {
		"content": true
		}
		}
		},
		"inputs": {},
		"layout": {
		"globalInputs": [],
		"type": "absolute",
		"options": {},
		"structure": [
		{
		"item": "viz_2aae822a0

Component	Description	

		"type": "block",
		"position": {
		"x": 0,
		"y": 0,
		"w": 300,
		"h": 300
		}
		},
		{
		"item": "viz_3a1a36fec
		"type": "block",
		"position": {
		"x": 330,
		"y": 0,
		"w": 250,
		"h": 250
		}
		},
		{
		"item": "viz_cf5bd9532
		"type": "block",
		"position": {
		"x": 640,
		"y": 40,
		"w": 150,
		"h": 160
		}
		},
		{
		"item": "viz_36b6e66b1
		"type": "block",
		"position": {
		"x": 10,
		"y": 340,
		"w": 250,
		"h": 250
		}
		},
		{
		"item": "viz_f3479a853
		"type": "block",
		"position": {
		"x": 370,
		"y": 270,
		"w": 310,
		"h": 60
		}
		}
		]
		},
		"descriptionLength": 0,
		"titleLength": 44
		},
		"app": "splunk-dashboard-studio",
		"page": "_do_not_edit_delete_telemetry
		}
		}

app.splunk_monitoring_console	Determines whether splunk_monitoring_console	{ [-] component: app.splunk_monitoring_console.

Component	Description	

	is enabled. If enabled, determines whether the mode is standalone or distributed.	data: {
		disabled: 1
		mode: standalone
		mc_auto_config: disabled
		role_list: ["license_master", "license
		"cluster_master", "cluster_manager",
		"search_head", "kv_store"]
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c
		executionID: F0AE995E8653D768A360E73BE3F
		timestamp: 1540570045
		transactionID: 89F7329E-86AD-BBFD-034F-2
		version: 3
		visibility: anonymous,support
		}

assist-app.appVersion.<appId>	Splunk Assist - App Assist	{ [-]
		"name": "assist-app.appVersion.<appId>",
		"category": "apps",
		"entityID": "<search-head>"
		"entityType": "search-head",
		"status": "critical" | "warning" | "confo
		"updatedAt": "<timestamp>",
		"details": {
		"installedVersion": "<release
		"latestVersion": "<latest vers
		}
		}
		}

assist-certificate.expiry	Splunk Assist - Certificate Assist	{ [-]
		name: assist-certificate.expiry
		displayName: "Certificate expiration"
		category: "availability" | "security" | "
		entityID: "data_034"
		entityType: "indexer"
		status: "critical" | "warning" | "conform
		updatedAt: timestamp
		previousStatus: "critical" | "warning" |
		version: <version>
		details: {
		"expiry" : <timestamp>,
		"subject" : <subject dn>,
		"serial" : <serial number>,
		"fingerprint" : <fingerprint>,
		"issuer" : <issuer dn>
		}
		}

assist-app.appVersion.<appId>	Splunk Assist - Config Assist	{ [-] "name": "assist-config.<file>.<stanza>.<p "entityID": "<splunk_server>", "entityType": "search-head", "status": "critical" | "warning" | "confo "details": { "file": "<file>", "stanza": "<stanza>", "property": "<property>", "valueType": "bool" | "string"

Component	Description	
		"currentValue": "<current_prop "expectedValue": "<expected_pr } } }
scripted_inputc.telemetry	Describes how much data is ingested through scripted input.	app: "scripted_input" version: none bytes: number of bytes ingested { [-] app: component: scripted_inputc.telemetry data: { [-] app: scripted_input bytes: 7645634 version: no version } deploymentID: 18393d55-3552-546c-a5ab-61a96a04 eventID: 367E743C-D629-4B25-B46A-78447116F3A4 executionID: 319FB159-0B47-4CA0-B29D-4CD0EDDF0 optInRequired: 1 timestamp: 1586974636 type: event userID: 574f5debd4e54c49ef018a6e1bde0379df499a visibility: [[-] anonymous support] }
cherrypy.load	How frequently CherryPy routes are used.	{ [-] app: search component: cherrypy.route.load data: { [-] class: ViewController file_path: controllers/view.py route: /:app/:view_id splunkVersion: 20220805 } deploymentID: 06b3d792-4e28-5402-97d4-54bab eventID: 55E95327-6505-423A-B1F8-EA90946977 executionID: 8ED4CAB4-DACF-4F9D-A631-4084DD optInRequired: 3 timestamp: 1663796731 type: event userID: da69a92a70c0997c4db33654b7621445d38 visibility: [[-] anonymous support] }
deployment.app	Apps installed on search head and peers.	{ [-] enabled: true host: 878e7b21bf98580dbdb4ed3baf6c35d78aa name: search

Component	Description	

		<pre>version: 8.0.0
		}</pre>

deployment.clustering.indexer	Host name of an indexer, replication factor, and search factor for indexer cluster.	<pre>{ [-]
		enabled: false
		host: 06d3392e0644587c3c3131833c81bfa6a7b
		timezone: -0700
		}</pre>

deployment.clustering.member	Indexer cluster member status.	<pre>{ [-]
		master: 1b83dc9e131f02b53329dfc1d3700aea9
		member: { [-]
		guid: 14B1E1C3-ABD1-4D02-88D5-3A6964EF8
		host: 942796f349f59b3ae64b47e507299b64b
		status: Up
		}
		site: default
		}</pre>

deployment.clustering.searchhead	Indexer cluster and search head connection status.	<pre>{ [-]
		master: 1b83dc9e131f02b53329dfc1d3700aea9
		searchhead: { [-]
		guid: 141D5E4A-3C5C-4051-B2DB-E679027A0
		host: f7724a2690f17f0fe3ea97418c92fffde
		status: Connected
		}
		site: default
		}</pre>

deployment.distsearch.peer	Distributed search peer status.	<pre>{ [-]
		host: 33b1957bfe1d0f7d3aac34e8655cf49f743
		peer: { [-]
		guid: 676F6738-BA57-44EC-94F0-A6821739D
		host: 76e4ed3636a6f4dc9737d119fde51e000
		status: Up
		}
		}</pre>

deployment.forwarders	Forwarder architecture: Number of hosts, number of forwarder instances, OS/version, CPU architecture, Splunk Enterprise version, distribution of forwarding volume	<pre>{ [-]
		architecture: x86_64
		bytes: { [-]
		avg: 632367800
		max: 689339847
		min: 602231091
		p10: 602891365
		p20: 603551640
		p30: 604211914
		p40: 604872189
		p50: 605532463
		p60: 622293940
		p70: 639055417
		p80: 655816893
		p90: 672578370
		}
		hosts: 3
		instances: 3
		os: Linux
		splunkVersion: 8.0.0
		type: full
		}</pre>

Component	Description	
deployment.httpEventCollector	Describes how much data is ingested through HEC for Splunk apps, add-ons, and connectors.	{ [-] app: component: deployment.httpEventCollector data: { [-] app: stream333 bytes: 50 version: 3.1 } deploymentID: 18393d55-3552-546c-a5ab-61a96a04 eventID: 367E743C-D629-4B25-B46A-78447116F3A4 executionID: 319FB159-0B47-4CA0-B29D-4CD0EDDF0 optInRequired: 1 timestamp: 1586974636 type: event userID: 574f5debd4e54c49ef018a6e1bde0379df499a visibility: [[-] anonymous support] }
deployment.index	Index type and configuration. Includes indicator of whether a metrics index has subsecond search capability.	{ [-] app: search buckets: { [-] cold: { [-] count: 0 events: 0 sizeGB: 0 } coldCapacityGB: unlimited homeCapacityGB: unlimited homeEventCount: 871 hot: { [-] count: 0 max: 3 sizeGB: 0 } thawed: { [-] count: 0 events: 0 sizeGB: 0 } warm: { [-] count: 6 sizeGB: 0 } } host: 6aac2d36b0f11492299b161a6c5a4f79451 name: uba_alarms timeResolution: sec total: { [-] buckets: 6 currentDBSizeGB: 0 events: 871 maxDataSizeGB: 500 maxTime: 1568987048 minTime: 1567603567 rawSizeGB: 0

Component	Description	
		} type: event }
deployment.licensing.slave	License slaves.	{ [-] master: 33b1957bfe1d0f7d3aac34e8655cf49f7 slave: { [-] guid: 1E7D1EA4-9E76-410B-825F-36CDA037F host: 33b1957bfe1d0f7d3aac34e8655cf49f7 pool: auto_generated_pool_enterprise } }
deployment.node	GUID, host, number of virtual and physical cores, CPU architecture, memory size, storage (partition) capacity, OS/version, Splunk Enterprise version	{ [-] cpu: { [+] } guid: 991BECEF-7F25-442D-B388-FF5A5AED16C host: cbefb1beb9ca9908007643320dec0ab0b34 memory: { [-] capacity: 32655630402 utilization: { [-] avg: 0.67 max: 0.74 min: 0.5 p10: 0.6 p20: 0.62 p30: 0.64 p40: 0.66 p50: 0.67 p60: 0.69 p70: 0.7 p80: 0.71 p90: 0.72 } } os: Linux osExt: Linux osVersion: 4.15.0-1031-aws partitions: [[-] { [-] capacity: 208111882207 fileSystem: ext4 utilization: 0.91 }] splunkVersion: 8.0.0 }

deployment.remoteupgrade	Information about remote upgrade of universal forwarder.	component: deployment.remoteupgrade
		data: {
		"total_remote_upgrade":15,
		"total_remote_ugprade_success":7,
		"total_remote_ugprade_failure":8,
		"total_remote_upgrade_pkg_rpm":2,
		"total_remote_upgrade_pkg_tgz":9,
		"total_remote_upgrade_os_darwin_x86_64":9,
		"total_remote_upgrade_os_linux_x86_64":2
		...
		}

Component	Description	
		date: 2018-10-26 deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6 executionID: F0AE995E8653D768A360E73BE3F544 timestamp: 1540570045 transactionID: 89F7329E-86AD-BBFD-034F-209C version: 3 visibility: anonymous,support
deployment.shclustering.member	Search cluster member status.	{ [-] captain: 208999515adad3c46696443afe61049c member: { [-] guid: 45B3EA5E-4868-4243-9BEA-109C2F76F host: 258a814c13167915bedd945acd0f5e16c status: Up } site: default }

htmlcleaner.dashboard	General telemetry collected on CSS tag usage.	{ [-]
		data: {
		app: search
		page: network_insights
		sanitizedTags: [
		"DIV",
		"H1",
		"SPAN"
		],
		inlineStyles: [
		{
		type: "StyleAttribute",
		element: "div",
		properties: [
		"background-color",
		"width"
		]
		},
		{
		type: "StyleElement",
		rulesets: [
		{
		properties: [
		"background-color"
		"content",
		"color"
		]
		},
		{
		properties: [
		"width"
		]
		}
		]
		}
		]
		}
		}

instrumentation.performance	Performance of instrumentation queries.	{ [-] instance_type: Single

Component	Description	
		queries: [[-] { [-] component: deployment.app isFailed: 0 resultCount: 145 runDuration: 0.843 scanCount: 0 searchProviders: 3 sid: 1569294993.84 } { [-] component: deployment.app isFailed: 0 resultCount: 145 runDuration: 1.079 scanCount: 0 searchProviders: 3 sid: 1569294995.85 } { [-] component: deployment.distsearch.peer isFailed: 0 resultCount: 2 runDuration: 0.211 scanCount: 0 searchProviders: 3 sid: 1569294996.86 } { [-] component: deployment.licensing.slave isFailed: 0 resultCount: 1 runDuration: 0.781 scanCount: 0 searchProviders: 3 sid: 1569294997.87 } { [-] component: usage.search.report_accele isFailed: 0 resultCount: 1 runDuration: 0.387 scanCount: 0 searchProviders: 3 sid: 1569294998.88 } { [-] component: usage.search.report_accele isFailed: 0 resultCount: 1 runDuration: 0.36 scanCount: 0 searchProviders: 3 sid: 1569294998.89 } { [-] component: usage.search.searchTelemet isFailed: 0 resultCount: 1

Component	Description	
		runDuration: 1.2650000000000001 scanCount: 14 searchProviders: 3 sid: 1569294999.90 } { [-] component: usage.lookups.lookupDefini isFailed: 0 resultCount: 1 runDuration: 0.28700000000000003 scanCount: 0 searchProviders: 1 sid: 1569295000.91 } { [-] component: performance.bundleReplicat isFailed: 0 resultCount: 3 runDuration: 1.238 scanCount: 2784 searchProviders: 3 sid: 1569295001.92 } { [-] component: performance.indexing isFailed: 0 resultCount: 8 runDuration: 6.098 scanCount: 35273 searchProviders: 3 sid: 1569295010.93 } { [-] component: performance.search isFailed: 0 resultCount: 3 runDuration: 21.253 scanCount: 213234 searchProviders: 3 sid: 1569295016.94 } { [-] component: usage.search.concurrent isFailed: 0 resultCount: 8 runDuration: 8.671 scanCount: 167724 searchProviders: 3 sid: 1569295038.96 } { [-] component: usage.users.active isFailed: 0 resultCount: 3 runDuration: 9.34 scanCount: 56960 searchProviders: 3 sid: 1569295047.97 }

Component	Description	
		{ [-] component: deployment.node isFailed: 0 resultCount: 15 runDuration: 9.965 scanCount: 1166 searchProviders: 3 sid: 1569295056.98 } { [-] component: deployment.index isFailed: 0 resultCount: 113 runDuration: 14.809000000000001 scanCount: 0 searchProviders: 3 sid: 1569295067.99 } { [-] component: usage.search.type isFailed: 0 resultCount: 3 runDuration: 17.365000000000002 scanCount: 167724 searchProviders: 3 sid: 1569295082.100 } { [-] component: licensing.stack isFailed: 0 resultCount: 5 runDuration: 1.772 scanCount: 10 searchProviders: 3 sid: 1569295100.101 } { [-] component: deployment.forwarders isFailed: 0 resultCount: 28 runDuration: 8.309000000000001 scanCount: 268106 searchProviders: 3 sid: 1569295102.102 } { [-] component: usage.indexing.sourcetype isFailed: 0 resultCount: 1373 runDuration: 45.673 scanCount: 735929 searchProviders: 3 sid: 1569295111.103 } { [-] component: deployment.clustering.inde isFailed: 0 resultCount: 1 runDuration: 3.157

Component	Description	
		scanCount: 0 searchProviders: 1 sid: 1569295160.104 } { [-] component: usage.app.page isFailed: 0 resultCount: 9 runDuration: 0.795 scanCount: 65 searchProviders: 3 sid: 1569295163.105 }] roles: { [-] cluster_master: false in_cluster: false indexer: true kv_store: true lead_node: true license_master: true search_head: true } timezone: +0000 }

licensing.stack	Licensing quota and consumption.	{
		consumption: 127025471
		guid: C131C257-98FE-4E8B-9595-CB4D93246F9
		host: Splunk
		name: enterprise
		pools: [
		{
		consumption: 127025471
		quota: 6442450944
		}
		]
		product: enterprise
		quota: 6442450944
		subgroup: Production
		type: enterprise
		}

modinputc.telemetry	Describes how much data is ingested through Splunk apps, add-ons, and connectors.	{ [-] app: component: modinputc.telemetry data: { [-] app: stream333 bytes: 50 version: 3.1 } deploymentID: 18393d55-3552-546c-a5ab-61a96a04 eventID: 367E743C-D629-4B25-B46A-78447116F3A4 executionID: 319FB159-0B47-4CA0-B29D-4CD0EDDF0 optInRequired: 1 timestamp: 1586974636 type: event userID: 574f5debd4e54c49ef018a6e1bde0379df499a visibility: [[-]

Component	Description	
		anonymous support] }
performance.bundleReplicationCycle	Metrics for the bundle replication cycle.	{ [-] avgBundleBytes: 0 avgPeerCount: 1 avgPeerSuccessCount: 1 avgReplicationTimeMsec: 1 cycleCount: 144 replicationPolicy: classic }
performance.indexing	Indexing performance: Core utilization, storage utilization, memory usage, indexing throughput, search latency.	{ [-] host: 3c4681a5be1881de8554c8bab7be78e8d15 thruput: { [-] avg: 1903 max: 7854 min: 4 p10: 1419 p20: 1433 p30: 1452 p40: 1806 p50: 1860 p60: 1865 p70: 1878 p80: 2046 p90: 2326 total: 7138077 } }
performance.search	Search performance: Core utilization, storage utilization, memory usage, indexing throughput, search latency.	{ [-] buckets: { [-] avg: 1.9 max: 27 min: 0 p10: 0 p20: 0 p30: 0 p40: 0 p50: 0 p60: 0.88 p70: 2 p80: 6 p90: 6 } dayRange: { [-] avg: 876.81 max: 18162.29 min: 0 p10: 0 p20: 0 p30: 0 p40: 0 p50: 0 p60: 0.01 p70: 0.01 p80: 0.01

Component	Description	
		p90: 0.03 } latency: { [-] avg: 2.31 max: 19744.69 min: 0.01 p10: 0.02 p20: 0.02 p30: 0.09 p40: 0.47 p50: 1.6 p60: 1.85 p70: 2.05 p80: 2.23 p90: 2.64 } scanCount: { [-] avg: 344030.32 max: 38060408 min: 0 p10: 0 p20: 0 p30: 0 p40: 0 p50: 1.59 p60: 90.32 p70: 1156.18 p80: 25454.25 p90: 308440.56 } searches: 30576 slices: { [-] avg: 5034.33 max: 219740 min: 0 p10: 0 p20: 0 p30: 0 p40: 0 p50: 0 p60: 0 p70: 2246.06 p80: 11491.43 p90: 14170.42 } }
preactivation.activate_button.click	Splunk Assist: Click of the 'Turn on Splunk Assist' button on the preactivation page	{ [-] app: splunk_instrumentation component: otel data: { [-] duration: 100 id: e885d2195fc2da5f name: preactivation.activate_button.click parentId: 384a56ef157fdb04 severity: info source: splunk-assist-telemetry tags: { [-] analyticsSessionId: ac692c428bdd9b311982b2fe4c47ec75-407a7b72cb5c2

Component	Description	
		app: splunk-assist-telemetry browser.name: edge-chromium browser.version: 98.0.1108 environment: play isInternalUser: false location.href: http://f746a79e790051f6d1c546e40fd2392cd155dda /onboarding os.name: Windows 10 preferred.color.scheme: light screen.size: {"width":1024,"height":768 splunk.telemetry: skinny-web-openteleme splunk.telemetryType: manual splunk.telemetryVersion: 1.15.2 tenant: e3b0c44298fc1c149afbf4c8996fb92 user: 407a7b72cb5c2f6caa9bc0a5e8262f6a2 useragent: Mozilla/5.0 (Windows NT 10.0 Chrome/98.0.4758.102 Safari/537.36 Edg/98.0.11 } timestamp: 1681813688910000 traceId: 0c656f44e52f4b033452388f258883b0 } deploymentID: c51647a9-5c49-551c-8c13-1b38b eventID: AF1B33F7-C6AF-495B-AD3F-1D2976E333 executionID: 27B43822-E2F0-4653-B9D0-016AF5 optInRequired: 0 original_timestamp: 1681813703 timestamp: 1681813703 type: event userID: 8c77775e05bdf6989c89d88a47de57bf5df visibility: [[-] anonymous support] }
preactivation.support_button.click	Splunk Assist: Click of the 'Contact Splunk support' button on the preactivation page	{ [-] app: splunk_instrumentation component: otel data: { [-] duration: 100 id: 1cc5688f07374baa name: preactivation.support_button.click parentId: eec194a91c38640f severity: info source: splunk-assist-telemetry tags: { [-] analyticsSessionId: a7960cbdea4fd0e70c587cf2385b166f-407a7b72cb5c2 app: splunk-assist-telemetry browser.name: chrome browser.version: 111.0.0 environment: prod isInternalUser: false location.href: http://f746a79e790051f6d1c546e40fd2392cd155dda /onboarding os.name: Mac OS preferred.color.scheme: dark screen.size: {"width":3440,"height":144

Component	Description	
		splunk.telemetry: skinny-web-opentelem splunk.telemetryType: manual splunk.telemetryVersion: 1.15.2 tenant: e3b0c44298fc1c149afbf4c8996fb92 user: 407a7b72cb5c2f6caa9bc0a5e8262f6a2 useragent: Mozilla/5.0 (Macintosh; Inte Chrome/111.0.0.0 Safari/537.36 } timestamp: 1681844042684000 traceId: 28bc43c6508bf218ee6f5bbb795360af } deploymentID: 09446c1e-b100-526e-923d-c3eab eventID: 392DCE6C-FAA5-491E-A8E3-B485588B6D executionID: 71882FFB-EC35-40F3-B63F-BA98F8 optInRequired: 0 original_timestamp: 1681844042 timestamp: 1681844042 type: event userID: b4d2583bdeef806f1721ca20e001767e215 visibility: [[-] anonymous support] }
onboarding.activate_button.click	Splunk Assist: Click of the 'Turn on Splunk Assist' button on the onboarding page (landing page of Assist)	{ [-] app: splunk_instrumentation component: otel data: { [-] duration: 200 id: e1f05f657184f226 name: onboarding.activate_button.click parentId: cd1ecae930ad4399 severity: info source: splunk-assist-telemetry tags: { [-] analyticsSessionId: ac692c428bdd9b311982b2fe4c47ec75-407a7b72cb5c2 app: splunk-assist-telemetry browser.name: edge-chromium browser.version: 98.0.1108 environment: play isInternalUser: false location.href: http://f746a79e790051f6d1c546e40fd2392cd155dda /onboarding os.name: Windows 10 preferred.color.scheme: light screen.size: {"width":1024,"height":768 splunk.telemetry: skinny-web-opentelem splunk.telemetryType: manual splunk.telemetryVersion: 1.15.2 tenant: e3b0c44298fc1c149afbf4c8996fb92 user: 407a7b72cb5c2f6caa9bc0a5e8262f6a2 useragent: Mozilla/5.0 (Windows NT 10.0 Chrome/98.0.4758.102 Safari/537.36 Edg/98.0.11 } timestamp: 1681813688790000 traceId: 0f0cd0643fc0fa8ac0ec107da09ca4a3 }

Component	Description	

		<pre> deploymentID: c51647a9-5c49-551c-8c13-1b38b
		   eventID: D1A93DAE-EE3E-4428-832E-2445AAD67D
		   executionID: 27B43822-E2F0-4653-B9D0-016AF5
		   optInRequired: 0
		   original_timestamp: 1681813702
		   timestamp: 1681813702
		   type: event
		   userID: 8c77775e05bdf6989c89d88a47de57bf5df
		   visibility: [[-]
		     anonymous
		     support
		   ]
		}</pre>

overview.category_card.click	Splunk Assist: The mouse click event of the category cards at the top of the overview page	<pre>{ [-]
		   app: splunk_instrumentation
		   component: otel
		   data: { [-]
		     duration: 900
		     id: 60c53cbc98c3a193
		     name: overview.category_card.click
		     severity: info
		     source: splunk-assist-telemetry
		     tags: { [-]
		       analyticsSessionId:
		3fc143888a31bfc23a654238a5b4d404-407a7b72cb5c2
		       app: splunk-assist-telemetry
		       browser.name: chrome
		       browser.version: 112.0.0
		       category: availability
		       conforming: 5
		       critical: 1
		       environment: play
		       isInternalUser: false
		       location.href:
		http://a7aa8e2b90a79dd144265ec0e9d9908fd3a12fc
		/overview
		       os.name: Mac OS
		       preferred.color.scheme: light
		       screen.size: {"width":1792,"height":112
		       splunk.telemetry: skinny-web-openteleme
		       splunk.telemetryType: manual
		       splunk.telemetryVersion: 1.18.0
		       tenant: a7aa8e2b90a79dd144265ec0e9d9908
		       user: 407a7b72cb5c2f6caa9bc0a5e8262f6a2
		       useragent: Mozilla/5.0 (Macintosh; Inte
		Chrome/112.0.0.0 Safari/537.36
		       warning: 0
		     }
		     timestamp: 1683157870140000
		     traceId: 31a3c8660a709198f3f1597872e23ee2
		   }
		   deploymentID: 7912c58c-b6bd-5142-9183-bb1aa
		   eventID: 00644935-0ABB-458D-8773-1C9425D7A0
		   executionID: EFDE4AC1-1E12-404E-AA7E-1FB81A
		   optInRequired: 0
		   original_timestamp: 1683157871
		   timestamp: 1683157871
		   type: event
		   userID: 06bf24a855997aa495e698d0e87ea164ddf</pre>

Component	Description	
		visibility: [[+]] }
overview.topology.node.click	Splunk Assist: The mouse click event of the topology cards in the Indicators breakdown panel on the overview page	{ [-] app: splunk_instrumentation component: otel data: { [-] duration: 200 id: 20ab9500a297af6c name: overview.topology.node.click severity: info source: splunk-assist-telemetry tags: { [-] analyticsSessionId: afde1287a145bb32b2da0e473dedbe51-407a7b72cb5c2 app: splunk-assist-telemetry browser.name: chrome browser.version: 113.0.0 count: 11 environment: play isInternalUser: false location.href: http://a7aa8e2b90a79dd144265ec0e9d9908fd3a12fc /overview nodeType: search_head os.name: Mac OS preferred.color.scheme: light screen.size: {"width":1792,"height":112 splunk.telemetry: skinny-web-openteleme splunk.telemetryType: manual splunk.telemetryVersion: 1.18.0 status: warning tenant: a7aa8e2b90a79dd144265ec0e9d9908 user: 407a7b72cb5c2f6caa9bc0a5e8262f6a2 useragent: Mozilla/5.0 (Macintosh; Inte Chrome/113.0.0.0 Safari/537.36 } timestamp: 1683225546025000 traceId: a1865c6b31a78db19a61050915c5c267 } deploymentID: 7912c58c-b6bd-5142-9183-bb1aa eventID: 6CB91A1F-E898-425C-B4B9-76292542F6 executionID: 8DE5F611-B2A5-47A8-BBD4-BBA50A optInRequired: 0 original_timestamp: 1683225547 timestamp: 1683225547 type: event userID: 06bf24a855997aa495e698d0e87ea164ddf visibility: [[+]] }
overview.overview_list.open_assist.click	Splunk Assist: The mouse click event of the action button on Overview table. Clicking on this button will open up the assist page for the indicator indicated by indicatorName	{ [-] app: splunk_instrumentation component: otel data: { [-] duration: 200 id: 22e35157c73c06ea name: overview.overview_list.open_assist.

Component	Description	
		severity: info source: splunk-assist-telemetry tags: { [-] analyticsSessionId: c4f2271527ab0a4497abc1d66d29fcd0-407a7b72cb5c2 app: splunk-assist-telemetry browser.name: chrome browser.version: 112.0.0 conforming: 3 critical: 0 environment: play indicatorName: assist-certificate.expir isInternalUser: false location.href: http://559f8b32939d9748b4e512a6c69050be140e8c3 /overview os.name: Mac OS preferred.color.scheme: light screen.size: {"width":1792,"height":112 splunk.telemetry: skinny-web-openteleme splunk.telemetryType: manual splunk.telemetryVersion: 1.18.0 tenant: 559f8b32939d9748b4e512a6c69050b user: 407a7b72cb5c2f6caa9bc0a5e8262f6a2 useragent: Mozilla/5.0 (Macintosh; Inte Chrome/112.0.0.0 Safari/537.36 warning: 0 } timestamp: 1683227439182000 traceId: c4974a407db142f4dcb30f4ab1f6767e } deploymentID: a3d7ae8d-f11a-56f7-a4d1-fd455 eventID: 6A3486A8-D5BF-4669-AF08-0F48D83BD6 executionID: 5712114F-1486-45E6-A907-5C6CA0 optInRequired: 0 original_timestamp: 1683227446 timestamp: 1683227446 type: event userID: 7503b69d9c6e85f659d04ab0582e4fab2ae visibility: [[+]] }
usage.admissionRules.report	Admission rules: Status, list of rules enabled and rules triggered for filtered searches.	{ [-] app: splunk_instrumentation component: usage.admissionRules.report data: { [-] admissionRulesEnabled: 1 guid: 13E5506A-4C0F-4BB9-B468-B5F977A00FD host: e521fc4eebd5e93b2cadcced3e03f699c86 rules: { [-] allindex_alltime: { [-] predicate: index=df58248c414f342c81e0 } audit: { [-] predicate: index=cb4ed408dd9f3497da0b AND role=d033e22ae348aeb5660fc2140aec35850c4da } internal: { [-] predicate: index=f1b1f1f40216ee2e2b5a

Component	Description	
		AND search_time_range=alltime } totalCount: 3 } rulesTriggered: [[-] { [-] filteredSearchesCount: 1 searchFilterRule: allindex_alltime } { [-] filteredSearchesCount: 3 searchFilterRule: audit } { [-] filteredSearchesCount: 1 searchFilterRule: internal }] serverRoles: indexer, license_master } deploymentID: dc739253-34a9-5b44-afd8-ea73e eventID: DE0063AE-31F5-42FA-AE92-0F62913EF4 executionID: 8B45C62A-0D0B-4689-B1BD-F29BFA optInRequired: 3 timestamp: 1587004320 type: aggregate visibility: [[-] anonymous support] }
usage.app.page	App name, page name, locale, number of users, number of page loads, generated as session data.	{ [-] app: search locale: en-US occurrences: 1 page: users: 1 }
usage.authMethod.config	Authentication method: Hashed host and GUID, authentication method (Splunk, LDAP, or SAML), MFA type (none, Duo, or RSA).	{ [-] authentication method: Splunk guid: C099BFA3-E5B5-4AB1-AB64-471703C5438 host: 8cd44b23a1bd3ae283f21a7d9c543416318 mfa type: none }

usage.bucketmerge.clustered	Usage of cluster bucket merge command, cluster bucket list command, and cluster bucket merge command with -dryrun option.	{ [-]
		component: usage.bucketmerge.clustered
		data: {
		command: merge
		newBucketsCount: 5
		oldBucketsCount: 50
		bucketsFailedToMergeCount: 2
		indexersCount: 10
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6
		executionID: F0AE995E8653D768A360E73BE3F544
		timestamp: 1540570045

Component	Description	
		transactionID: 89F7329E-86AD-BBFD-034F-209C version: 3 visibility: anonymous,support }

usage.bucketmerge.standalone	Usage of bucket merge command, bucket list command, and bucket merge command with --dryrun option.	{ [-]
		component: usage.bucketmerge.standalone
		data: {
		command: merge
		newBucketsCount: 5
		oldBucketsCount: 50
		durationSec: 7.5
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6
		executionID: F0AE995E8653D768A360E73BE3F544
		timestamp: 1540570045
		transactionID: 89F7329E-86AD-BBFD-034F-209C
		version: 3
		visibility: anonymous,support
		}

usage.configtracker.config	Whether or not the feature is enabled or disabled. What "mode" the feature is in (e.g. - diff, track_only, auto.) And what kinds of file paths, and/or fields are added to the denylist.	{ [-]
		component: usage.configtracker.config
		data: {
		disabled: false
		mode: auto
		denylist: someregexfilterhere
		uses_inotify: true
		exclude_fields: server.conf:general:pass4
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6
		executionID: F0AE995E8653D768A360E73BE3F544
		timestamp: 1540570045
		transactionID: 89F7329E-86AD-BBFD-034F-209C
		version: 3
		visibility: anonymous,support
		}

usage.configtracker.introspection	Configuration file change logs made on a Splunk instance.	{ [-]
		component: usage.configtracker.introspectio
		data: {
		count: 102
		path: $SPLUNK_HOME/etc/system/local/trans
		stanza: hostoverride
		prop: DEST_KEY, REGEX, FORMAT
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6
		executionID: F0AE995E8653D768A360E73BE3F544
		timestamp: 1540570045
		transactionID: 89F7329E-86AD-BBFD-034F-209C
		version: 3
		visibility: anonymous,support
		}

usage.configtracker.searches	Configuration file change SPL queries that were run on an environment, and	{ [-] component: usage.configtracker.searches data: {

Component	Description	
	their corresponding results.	user_count: 20 total_search_count: 754 } date: 2018-10-26 deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6 executionID: F0AE995E8653D768A360E73BE3F544 timestamp: 1540570045 transactionID: 89F7329E-86AD-BBFD-034F-209C version: 3 visibility: anonymous,support }
usage.durableSearch	Number of users of the durable search feature, how durable search is being used (for scheduled searches? for summary indexing?), and commonly-used durable search setting values.	{ [-] durableBackfillType: auto durableLagTime: 60 durableMaxBackfillIntervals: 100 durableTrackTimeType: _indextime enableSummaryIndex: Yes name: 8a4d0e8816a25ed813c5f40dbfc34d0bd46 } date: 2020-06-02 deploymentID: 87402ea1-6505-59d5-b04a-c12dc executionID: ED6EF443C5FC863A9AABA6B89A1839 timestamp: 1591117572 transactionID: 0B2234FD-2D78-7939-75B1-B5BE version: 4 visibility: anonymous,support
usage.healthMonitor.currentState	Distributed health report: Enabled status, number of clicks, node status (node path, current color, worst color in last 24 hours), Splunk version.	{ [-] enabled: 1 } healthReportClicks: 10 nodeStatus: [[-] { [-] color: green nodePath: splunkd worstColorInLast24Hours: green } { [-] color: green nodePath: splunkd.file_monitor_input worstColorInLast24Hours: green } { [-] color: green nodePath: splunkd.file_monitor_input. worstColorInLast24Hours: green } { [-] color: green nodePath: splunkd.file_monitor_input. worstColorInLast24Hours: green } { [-] color: green nodePath: splunkd.index_processor worstColorInLast24Hours: green } { [+] }

Component	Description	
		{ [+] } { [+] } { [+] } { [+] } { [+] } { [+] }] splunkVersion: 8.1.0 }
usage.healthMonitor.report	Health report manager: Alert actions and enabled status, feature thresholds and enabled status.	{ [-] alert: { [-] alert_action:email: { [-] action/ action.to/ action.url/ action disabled: 0 } alert_action:webhook: { [-] action/ action.to/ action.url/ action disabled: 0 } health_reporter: { [-] action/ action.to/ action.url/ action disabled: 0 } } feature:batchreader: { [-] enabled: 1 threshold: { [-] indicator:data_out_rate:red: 2 indicator:data_out_rate:yellow: 1 } } feature:buckets: { [-] enabled: 1 threshold: { [-] indicator:buckets_created_last_60m:re indicator:buckets_created_last_60m:ye indicator:percent_small_buckets_creat indicator:percent_small_buckets_creat } } feature:cluster_bundles: { [-] enabled: 1 threshold: { [-] indicator:cluster_bundles:yellow: 1 } } feature:data_durability: { [-] enabled: 1 threshold: { [-] indicator:cluster_replication_factor: indicator:cluster_search_factor:red: } }

Component	Description	
		feature:data_searchable: { [-] enabled: 1 threshold: { [-] indicator:data_searchable:red: 1 } } feature:ddaa_archived_buckets: { [-] enabled: 1 threshold: { [-] indicator:archived_buckets_failed_las indicator:archived_buckets_failed_las } } feature:disk_space: { [-] enabled: 1 threshold: { [-] indicator:disk_space_remaining_multip indicator:disk_space_remaining_multip } } feature:indexers: { [-] enabled: 1 threshold: { [-] indicator:detention:red: 1 indicator:detention:yellow: 1 indicator:missing_peers:red: 1 indicator:missing_peers:yellow: 1 } } feature:indexing_ready: { [-] enabled: 1 threshold: { [-] indicator:indexing_ready:red: 1 } } feature:master_connectivity: { [-] enabled: 1 threshold: { [-] indicator:master_connectivity:red: 1 } } feature:replication_failures: { [-] enabled: 1 threshold: { [-] indicator:replication_failures:red: 1 indicator:replication_failures:yellow } } feature:s2s_autolb: { [-] enabled: 1 threshold: { [-] indicator:s2s_connections:red: 70 indicator:s2s_connections:yellow: 20 } } feature:search_lag: { [-] enabled: 1 threshold: { [-] indicator:count_extremely_lagged_sear

Component	Description	

		indicator:count_extremely_lagged_sear
		indicator:percent_searches_lagged_hig
		indicator:percent_searches_lagged_nor
		}
		}
		feature:searches_delayed: { [-]
		enabled: 1
		threshold: { [-]
		indicator:percent_searches_delayed_hi
		indicator:percent_searches_delayed_hi
		indicator:percent_searches_delayed_no
		indicator:percent_searches_delayed_no
		}
		}
		feature:searches_skipped: { [-]
		enabled: 1
		threshold: { [-]
		indicator:percent_searches_skipped_hi
		indicator:percent_searches_skipped_hi
		indicator:percent_searches_skipped_no
		indicator:percent_searches_skipped_no
		}
		}
		feature:searchheadconnectivity: { [-]
		enabled: 1
		threshold: { [-]
		indicator:master_connectivity:red: 1
		indicator:master_version_compatibilit
		}
		}
		feature:shc_captain_common_baseline: { [-
		enabled: 1
		threshold: { [-]
		indicator:common_baseline:red: 1
		}
		}
		feature:shc_captain_connection: { [-]
		enabled: 1
		threshold: { [-]
		indicator:captain_connection:red: 1
		indicator:captain_existence:red: 1
		}
		}
		feature:shc_captain_election_overview: {
		enabled: 1
		threshold: { [-]
		indicator:dynamic_captain_quorum:yell
		}
		}
		feature:shc_members_overview: { [-]
		enabled: 1
		threshold: { [-]
		indicator:detention:red: 1
		indicator:detention:yellow: 1
		indicator:replication_factor:yellow:
		indicator:status:red: 1
		indicator:status:yellow: 1
		}
		}

Component	Description	
		feature:shc_snapshot_creation: { [-] enabled: 1 threshold: { [-] indicator:snapshot_creation:red: 20 indicator:snapshot_creation:yellow: 1 } } feature:slave_state: { [-] enabled: 1 threshold: { [-] indicator:slave_state:red: 1 indicator:slave_state:yellow: 1 } } feature:slave_version: { [-] enabled: 1 threshold: { [-] indicator:slave_version:red: 1 } } feature:splunkoptimize_processes: { [-] enabled: 1 threshold: { [-] indicator:concurrent_optimize_process } } feature:tailreader: { [-] enabled: 1 threshold: { [-] indicator:data_out_rate:red: 2 indicator:data_out_rate:yellow: 1 } } feature:wlm_configuration_check: { [-] enabled: 1 threshold: { [-] indicator:configuration_check:red: 0 } } feature:wlm_system_check: { [-] enabled: 1 threshold: { [-] indicator:system_check:red: 0 } } }
usage.indexing.sourcetype	Indexing volume, number of events, number of hosts, source type name.	{ [-] bytes: 90962 events: 354 hosts: 1 name: splunk_telemetry }

usage.ingestactions.deletions	Count of destination and ruleset deletions	{ [-]
		data: {
		destinationDeletions: 3
		rulesetDeletions: 1
		}
		date: 2018-10-26

Component	Description	
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6 executionID: F0AE995E8653D768A360E73BE3F544 timestamp: 1540570045 transactionID: 89F7329E-86AD-BBFD-034F-2090 version: 3 visibility: anonymous,support }

usage.ingestactions.destinations	Characteristics of routing destinations	{ [-]
		data: {
		destinations: { [
		{
		batchSizeThresholdKB: 131072
		batchTimeout: 5
		compression : none
		dropEventsOnUploadError: false
		encryption: none
		signatureVersion: v1
		supportsVersioning: true
		urlVersion: v1
		destinationType: s3
		authMethodAccesskey: true
		authMethodIAM": false
		}
		}]
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6
		executionID: F0AE995E8653D768A360E73BE3F544
		timestamp: 1540570045
		transactionID: 89F7329E-86AD-BBFD-034F-2090
		version: 3
		visibility: anonymous,support
		}

usage.ingestactions.rulesets	Count of routing destinations, ruletset types, and ruleset conditions	{ [-]
		data: {
		s3: 2
		filter: 3
		mask: 3
		route: 5
		set_index: 2
		clone: 2
		maskRegexCount: 3
		filterRegexCount: 2
		filterEvalExprCount: 1
		routeRegexCount: 3
		routeEvalExprCount: 2
		uniqueIndexCount: 2
		}
		date: 2018-10-26
		deploymentID: 99b6ffd8-2e80-5e3b-905c-8c6f6
		executionID: F0AE995E8653D768A360E73BE3F544
		timestamp: 1540570045
		transactionID: 89F7329E-86AD-BBFD-034F-2090
		version: 3
		visibility: anonymous,support
		}

usage.kvstore		

Component	Description	
	Metrics and performance data about KV store.	{ [-] usage.flushAverageMs: 5.3538461538461535 usage.instanceType: primary usage.memRamMb: 0 usage.memVirtualMb: 0 usage.oplogEndTime: 1569301264 usage.oplogStartTime: 1569222045 usage.oplogTimeRange: 79219 usage.readLatencyToUpTime: 0.000153653421 usage.readLatencyUsPerOp: 0.0215805328061 usage.storageEngine: mmapv1 usage.upTime: 3956 usage.version: 3.6.12-splunk usage.writeLatencyToUpTime: 0.00015365342 usage.writeLatencyUsPerOp: 0.000480090369 }
usage.lookups.lookupDefinitions	Lookup definition metadata with hashed lookup names.	{ [-] lookups: [[-] { [-] _timediff: is_temporal: 0 name: 96117ed21e74f16d452027ed8e16c5c sharing: system size: type: external } { [-] _timediff: is_temporal: 0 name: 256d0fae9448acc55cd2e5cbabe7dbe sharing: global size: 18053 type: file } { [-] _timediff: is_temporal: 0 name: 88767984d9dc6308309ffde5dc3591f sharing: global size: 832 type: file } { [-] _timediff: is_temporal: 0 name: 1b0131dbc851786586e269a2ba8b2f0 sharing: global size: type: geo } { [-] _timediff: is_temporal: 0 name: 6d47b91d0c0753e9332ec2c0f8c9561 sharing: global size: type: geo }]

Component	Description	
		}
usage.passwordPolicy.config	Password policy management: hashed host and GUID, attribute configurations.	{ [-] constant login time: 0.000 days until password expires: 90 enable lockout users: false enable password expiration: false enable password history: false enable verbose login fail message: true expiration alert in days: 15 failed login attempts: 5 force existing users to change weak passw guid: 32BEE8DE-E64D-4B02-B2FE-4F13F18A0CA host: b8758da2f94fd58e648bce573fa3d9dc579 lockout duration in minutes: 30 lockout threshold in minutes: 5 minimum number of characters: 1 minimum number of digits: 0 minimum number of lowercase letters: 0 minimum number of special characters: 0 minimum number of uppercase letters: 0 password history count: 24 }
usage.python	Default setting for Python version in the app, path of the script with its name hashed, version of Python used in the script.	{ [-] pythonDefault: python2 scriptPath: /usr/local/bamboo/splunk-inst /D7A80DE23601F645B8A06995DF910A3D08AB9EAA scriptPythonVersion: python2 }
usage.rest	Usage of an endpoint, HTTP method, status code, and user agent in a REST request made from a Splunk Enterprise SDK. The data that is collected includes the partial endpoint URL of the target feature. Any user-identifiable data or resource names in the URL are discarded.	{ [-] endpointUri: search/jobs method: POST status: 200 userAgent: splunk-sdk-python/1.6.3 }
usage.savedSearches.alert	Usage of the saved search alerting functionality: triggering conditions and modes, alert actions, alert suppression, schedules, and so on.	{ [-] actionList: script alertConditionType: number of hosts alertSeverity: 3 alertSuppress: No alertSuppressGroup: 58e7079db82d48abfcdda alertTrackable: Yes cronSchedule: 0 0 * * * name: 831ee1f249cf286c2065e7ba7e38b0b5228 triggerMode: Once }
usage.search.concurrent	Distribution of concurrent searches.	{ [-] host: 3c4681a5be1881de8554c8bab7be78e8d15 searches: { [-] avg: 2 max: 2 min: 1

Component	Description	
		p10: 1 p20: 1 p30: 1 p40: 1 p50: 2 p60: 2 p70: 2 p80: 2 p90: 2 } }
usage.search.report_acceleration	Report acceleration metrics.	{ [-] existing_report_accelerations: 0 }
usage.search.searchTelemetry	List of commands and corresponding counts for all searches run on the system in the span of one day.	{ [-] commands: [[-] { [-] count: 1 name: addinfo } { [-] count: 5 name: eval } { [-] count: 6 name: external_command } { [-] count: 9 name: fields } { [-] count: 1 name: inputlookup } { [-] count: 1 name: join } { [-] count: 1 name: litsearch } { [-] count: 2 name: makemv } { [-] count: 1 name: mvcombine } { [-] count: 2 name: mvexpand } { [-] count: 2

Component	Description	
		name: noop } { [-] count: 4 name: prerest } { [-] count: 1 name: prestats } { [-] count: 4 name: presummarize } { [-] count: 2 name: rename } { [-] count: 4 name: rest } { [-] count: 1 name: search } { [-] count: 3 name: stats } { [-] count: 4 name: summarize } { [-] count: 6 name: timeliner } { [-] count: 1 name: where }] }
usage.search.searchtelemetry.type	Search type, count, average bytes read, max bytes read, duration.	{ [-] searchTypeInformation: [[-] { [-] avg(bytes_read): 90531.02683363149 count: 559 duration: 1488.45949719 max(bytes_read): 46382154 type: adhoc } { [-] avg(bytes_read): 0 count: 3224 duration: 199.042348043 max(bytes_read): 0 type: scheduled

Component	Description	
		}] }
usage.search.searchtelemetry.sourcetypeUsage	Sourcetype usage.	{ [-] sourcetypeUsage: [[-] { [-] http_event_collector_metrics: 1 kvstore: 1 mongod: 3 search_telemetry: 1 splunk_disk_objects: 1 splunk_resource_usage: 1 splunk_web_service: 3 splunkd: 11 splunkd_remote_searches: 3 splunkd_ui_access: 2 }] }
usage.search.type	Number of searches of each type.	{ [-] ad-hoc: 3619 datamodel acceleration: 1 other: 2 report acceleration: 1 scheduled: 34412 summary index: 506 }
usage.smartStore.Config	SmartStore global configuration, per index configuration, hashed internal and external index names.	{ [-] global config: { [-] cachemanager: { [-] eviction_padding: 5120 hotlist_bloom_filter_recency_hours: 3 hotlist_recency_secs: 86400 max_cache_size: 0 } clustering: { [-] mode: disabled } diskUsage: { [-] minFreeSpace: 5000 } } list of indexes: { [-] non-SmartStore enabled: ea9f4255e269599dd961c3efd8775ab5ac1d3948,f1b1f 7a7a0fa8d74d,568b2f85dcc1c8608d713a66a0eabd5b8 06619007f6659c41827885700,66f79d8a6327c82c9033 f77578164d1b03fb4c931f727a3e2966e541d4,0d176ba c2d248f862,05535ecff78ef61038725b6ed3016b8c9a0 } per index config: { [-] external_05535ecff78ef61038725b6ed3016b frozenTimePeriodInSecs: 188697600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000

Component	Description	
		} external_0d176ba3aa7be325bcaeaf13ea2da4 frozenTimePeriodInSecs: 188697600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } external_66f79d8a6327c82c9033e6d65ff033 frozenTimePeriodInSecs: 604800 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } external_87da723b9f33eb0f1bcad8ea3405d8 frozenTimePeriodInSecs: 188697600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } external_b28b7af69320201d1cf206ebf28373 frozenTimePeriodInSecs: 188697600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } external_f397214775e4f8191c17e838b4d518 frozenTimePeriodInSecs: 188697600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } external_f4f77578164d1b03fb4c931f727a3e frozenTimePeriodInSecs: 188697600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } internal_302a11446cd560395417c9e2d2177a frozenTimePeriodInSecs: 1209600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } internal_568b2f85dcc1c8608d713a66a0eabc frozenTimePeriodInSecs: 1209600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } internal_5a74588fcf73bdd06619007f6659c4 frozenTimePeriodInSecs: 2419200

Component	Description	
		hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } internal_d140ef99de26b2f8b6f54081084d0b frozenTimePeriodInSecs: 63072000 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } internal_ea9f4255e269599dd961c3efd8775a frozenTimePeriodInSecs: 188697600 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 7776000 } internal_f1b1f1f40216ee2e2b5a526eec43c8 frozenTimePeriodInSecs: 2592000 hotlist_bloom_filter_recency_hours: n hotlist_recency_secs: none maxGlobalDataSizeMB: 0 maxHotSpanSecs: 432000 } } total storage capacity: { [-] 0: { [-] available: 130459.672 capacity: 476802.039 free: 142405.105 fs_type: apfs } } }
usage.streamingMetricAlerts	Usage of the streaming metric alerting functionality: group by alerts, triggering evaluations and thresholds, alert suppression, result enrichment or filtering, and alert actions.	{ [-] actionList: email,rss alertSeverity: 2 alertTrackable: No hasComplexCondition: Yes hasDescription: Yes hasFilter: No hasGroupby: Yes hasLabels: Yes hasMultipleMetricIndexes: Yes name: 227a3ad2631f5a7fe8709f7cac3308580f5 triggerActionPerGroup: Yes triggerEvaluationPerGroup: Yes triggerExpires: 48h triggerMaxTracked: 10 triggerPrepare: No triggerSuppress: No triggerThreshold: once after 5m }
usage.users.active	The number of active users per day.	{ [-] active: 1 }

Component	Description	
usage.workloadManagement.report	Workload management: Hashed host and GUID, OS/version, server roles, WLM support and enable status, pool configurations, rule configurations.	{ [-] categories: { [-] ingest: { [-] allocated cpu percent: 20.00 allocated mem limit: 100.00 } misc: { [-] allocated cpu percent: 10.00 allocated mem limit: 10.00 } search: { [-] allocated cpu percent: 70.00 allocated mem limit: 70.00 } } guid: F3DC7C6B-DF89-4585-A7A6-B4A3510D957 host: eadc124359ea492c6b04c079dcf3bec3be2 os: Linux osVersion: 4.9.184-linuxkit pools: { [-] total count: 0 } rules: { [-] total count: 0 } server roles: indexer, license_master, kv wlm enabled: 0 wlm supported: 1 }

Support usage data examples

Support usage data is the same data as the aggregated usage data, but if you opt to send support usage data, Splunk can use the license GUID to identify usage data from a specific customer account to help troubleshoot support cases.

See Aggregated usage data examples.

Support usage data is distinct from diagnostic file data. Diagnostic files are never automatically generated and can only be sent to Splunk Support manually by a user with the appropriate permissions. For more about diagnostic files, see Generate a diag in the *Troubleshooting Manual*.

License usage data examples

The following example demonstrates the type of data sent to Splunk when sharing of license usage data is enabled.

Component	Description	Example
licensing.stack	Licensing quota and consumption	{ [-] consumption: 14462827 guid: 47798245-85D7-4DCA-A303-D49910F40ED1 host: fecaab81b0934386719a161bfe3656ca782ec6d14806ae15d4ec4dc5 name: enterprise pools: [[-] { [-] consumption: 14462827 quota: 53687091200

Component	Description	Example

		<pre> }
		  ]
		  product: enterprise
		  quota: 53687091200
		  subgroup: Production
		  type: enterprise
		}</pre>

Software version data examples

The following example demonstrates the software version data sent to Splunk for Splunk Enterprise when sharing of software version data is enabled.

Description	Example
CPU architecture	x86_64
Operating system	Linux
Product	enterprise
Splunk roles	admin
License group, subgroup, and hashed GUID	Enterprise, Production, <GUID>
Splunk software version	7.0.0

The following example demonstrates the software version data sent to Splunk for each app when sharing of software version data is enabled for that app.

Description	Example
App ID, name, and version	gettingstarted, Getting Started, 1.0
Splunk version	7.0
Platform, architecture	Darwin, x86_64

App usage data examples

In addition to the data enumerated in this topic, certain apps collect usage data. See the documentation for each app for details and examples.

- Splunk Add-on Builder: Share data in Splunk Add-on Builder
- Splunk App for AWS: Share data in the Splunk App for AWS
- Splunk Business Flow: Share data in Splunk Business Flow
- Splunk DB Connect: Share data in Splunk DB Connect
- Splunk Enterprise Security: Share data in Splunk Enterprise Security
- Splunk Industrial Asset Intelligence: Share data in Splunk Industrial Asset Intelligence
- Splunk IT Service Intelligence: Share data in Splunk IT Service Intelligence
- Splunk Machine Learning Toolkit: Share data in the Splunk Machine Learning Toolkit
- Splunk Security Essentials: Splunk Security Essentials Telemetry

How Splunk collects the data

If aggregated, support, or license usage data collection is enabled, a few instances in your Splunk Enterprise deployment collect data through scheduled searches. Most of the searches run in sequence, starting at 3:05 AM on the node that runs

the searches, unless you change the schedule. All searches are triggered with a scripted input.

In addition, when aggregated or support data collection is enabled, session data about user activity transmits from the browser directly to the Splunk telemetry API.

Which instance runs the searches and sends data to Splunk

One primary instance in your deployment runs distributed searches that collect most of the usage data. This primary instance is also responsible for sending the data to Splunk. The instance that acts as the primary instance depends on the details of your deployment:

- If indexer clustering is enabled, the cluster manager is the primary instance. If you have more than one indexer cluster, each cluster manager is a primary instance.
- If search head clustering is enabled but not indexer clustering, each search head captain is a primary instance.
- If your deployment does not use clustering, the searches run on a search head.

If you opt out of instrumentation, the searches from the primary instance do not run.

Additional instances in your deployment run a smaller number of searches, depending on colocation details. If data collection is enabled, the data from these searches is collected by the primary node and sent to Splunk. If you opt out, these searches still run, but no data is sent.

For your deployment to send data to Splunk, the primary instance responsible for the searches must be connected to the internet with no firewall rules or proxy server configurations that prevent outbound traffic to https://quickdraw.splunk.com/telemetry/destination or https://*.api.splkmobile.com. If necessary, add these URLs for outbound traffic to your firewall allow list.

Instrumentation in the Splunk Enterprise file system

After the searches run, the primary instance packages the searched data and sends it to Splunk. It also indexes the data to the `_telemetry` index. Session data is transmitted directly to the telemetry API from the browser. It does not go to the `_telemetry` index. The `_telemetry` index retains the data for two years by default and is limited in size to 256 MB.

The instrumentation app resides in the file system at `$SPLUNK_HOME/etc/apps/splunk_instrumentation`.

How Splunk uses the data it collects

If you share aggregated usage data, Splunk collects data about your Splunk software usage and aggregates it together with similar data from other deployments so Splunk can understand what features and workflows are most important to users and improve its products and services over time. Collected license IDs are used only to verify that data is received from a valid Splunk product and persisted only for deployments opting into license or support usage reporting. These license IDs help Splunk analyze how different Splunk products are being deployed across the population of customers and are not attached to any aggregated usage data.

If you share support usage data, Splunk links the data about your software usage to your installed license ID so that Splunk can provide improved support and services for your deployment. The Splunk Assist service uses support usage data to identify and provide insights to let you align your Splunk Enterprise deployment with Splunk best practices for security, performance, and configuration. The Support and Customer Success teams use this data to identify and troubleshoot support issues that you file and improve your Splunk software implementation.

If you share license usage data, Splunk uses the data to ensure compliance with your purchased offering.

If you share Splunk product version data, Splunk uses the data to track how many deployments use particular versions of Splunk software offerings and to provide in-product notifications when updates are available. For apps, version data is correlated with information about app downloads to populate app analytics views on Splunkbase provided to the app's developer, and to compute the number of installs on the app details page.

How Splunk transmits and stores the data it collects

When you enable aggregated, support, and license usage data sharing, Splunk Enterprise runs searches to collect this data and sends the search summaries to a collection endpoint. Session data and Splunk software version data is not included in the searches. Session data is sent from your browser as the events are generated. Version data about Splunk Enterprise is sent to Splunk by your browser after you log into Splunk Web. Version data about your Splunk apps is sent to Splunk daily through a REST call from splunkd to splunkbase.splunk.com. Data is transmitted to Splunk from a single primary instance in your deployment. See Which instance runs the searches and sends data to Splunk.

The Splunk platform encrypts telemetry data with transport layer security (TLS) before it leaves your deployment, and verifies authentication before it stores the data securely on Splunk cloud infrastructure. The infrastructure that customer telemetry uses has strict access controls that are subject to regular audit. For more information about how Splunk collects, uses, and discloses information about the data collected, see the Splunk Privacy Policy. For more information about Splunk's data privacy, security, and compliance practices, see Splunk Protects.

View the data your Splunk Enterprise deployment sends to Splunk

You can view aggregated usage, support usage, and license usage data that your deployment has recently sent in Splunk Web.

1. Navigate to **Settings > Instrumentation**.
2. Click the category of data you wish to view in Search.

This log is available only after the first run of the collection. To inspect the type of data that gets sent before you opt in on your production environment, you can opt in on your sandbox environment.

To view the browser session data, use JavaScript logging in your browser. Look for network events sent to a URL containing `splkmobile`. Events are triggered by user actions such as navigating to a new page in Splunk Web.

To view version data that is sent for Splunk Enterprise, watch JavaScript network traffic as you log into Splunk Web. The data is sent inside a call to quickdraw.splunk.com.

How to opt out

Splunk collects support usage, aggregated usage, license data, and software version data by default. You can opt in or out at any time.

Prerequisite
To enable or disable collection of usage data, the user that you use to log into Splunk Enterprise must hold a role that includes the `edit_telemetry_settings` capability.

Opt out of sharing aggregated or support usage data

To change your aggregated or support usage data sharing settings, follow these steps:

1. Click **Settings > Instrumentation** in Splunk Web.
2. Click the gear icon next to **Usage Data**.
3. Adjust the sliders to enable or disable sharing aggregated or support usage data.

Opt out of sharing license data automatically

By default, Splunk collects license usage data based on your installed license to ensure compliance with your purchased offering. To disable sharing license data automatically, edit your local copy of the `telemetry.conf` configuration file and set `sendLicenseUsage = false`.

Certain license programs require that you report your license usage. The easiest way to do this is to automatically send this information to Splunk. If you disable automatic license data sharing, you can send license data manually. Follow these steps each time you want to send data manually:

1. On a search head, log into Splunk Web.
2. Select **Settings > Instrumentation**.
3. Click **Export**.
4. Select a date range and data type.
5. Click **Send** to send data to Splunk directly or click **Export** to export the data to your local machine and send the data to Splunk using another mechanism.

Opt out of sharing software version data

To stop sending Splunk data about the version of Splunk Enterprise you have installed, edit the web.conf configuration file and set the value of the `updateCheckerBaseURL` setting to `0`.

In addition, you can turn off version data sharing for each Splunk app. To disable notifications of new versions and stop sending Splunk data about the app version, set `check_for_updates` to `false` in the local copy of the `app.conf` file for each app.

Opt out of sharing data and prevent future admins from opting in

To opt out from all collection of usage, support, and license data and prevent other admins from enabling it in the future, do the following on one search head in each cluster and on each non-clustered search head:

1. Click **Settings > Instrumentation** in Splunk Web.
2. Click the gear icon next to **Usage Data**.
3. Disable all options.
4. Click **Settings > Roles**.
5. Remove the `edit_telemetry_settings` capability from the `admin` role. Users with this role no longer receive notifications about data collection, nor can they access **Settings > Instrumentation** in Splunk Web.

If you want to disable collection of usage information across multiple deployments of the Splunk platform that are not centrally managed, block DNS resolution of `e1345286.api.splkmobile.com`.

How to adjust your data collection schedule

If you share data, the collection process begins daily at 3:00 AM by default. You can change the frequency and timing of this collection.

If all instances in your deployment are running Splunk Enterprise version 7.1.0 or later, you can schedule instrumentation to run starting at any hour of the day on a daily or a weekly schedule. The collection process runs a few searches in sequence on several instances in your deployment. Depending on the size of your deployment and whether you run instrumentation daily or weekly, it can take a few minutes before the final searches run on the primary instance to package and send the data to Splunk. See Which instance runs the searches.

Changing the instrumentation collection schedule has trade-offs. Scheduling the collection to run weekly instead of daily might decrease the total search load for the week. A weekly collection takes longer than a daily collection, because it gathers data from all seven days. If you choose weekly collection, set it for a day and time when you expect the search load to be low.

Change the collection schedule using Splunk Web

1. On a search head, in Splunk Web, navigate to **Settings > Instrumentation**.
2. Next to **Usage Data**, click the gear icon.
3. Click **Edit usage data schedule**.
4. Select a frequency, day, and time.
5. Click **Save**.

You do not need to restart the search head.

Change the collection schedule using configuration files

You can change the collection schedule by editing the `telemetry.conf` file. For guidelines on editing this file, see telemetry.conf.

1. At the command line on any search head, navigate to `$SPLUNK_HOME/etc/apps/splunk_instrumentation/local/`.
2. Create or edit `telemetry.conf`.
3. Edit the values for any of `scheduledHour`, `scheduledDay`, and `reportStartDate` according to the guidelines in `telemetry.conf.spec`.

Impacts on performance during collection of shared data

Aggregated usage, support usage, and license usage data is summarized and sent once per day at around 03:00 (3 am) by default. Splunk tested the performance impact on a deployment of one search head and three indexers and found the following performance impacts during the time that the searches were running:

- 4.5% increase in CPU overhead
- Negligible effects on memory, disk, and network overhead
- Up to 5% increase on the search time of regular search workloads

Session data and update checker data is sent from your browser as the events are generated. The performance implications are negligible.

How to enable data sharing for Splunk Assist

If you want to use the Splunk Assist service to monitor your Splunk Enterprise deployment according to Splunk best practices, or need to turn data sharing back on after you have opted out, use this procedure to confirm that data sharing is active.

1. Log into your Splunk Enterprise instance.
2. From the system bar, click **Settings > Instrumentation**.
3. On the "Instrumentation" page, click the gear icon next to **Usage Data**.
4. In the pop-up window that appears, review the **Aggregated Usage Data** and **Support Usage Data** toggle switches. Ensure that both toggle switches are set to "Enabled".
5. Click the gear icon again to close the Usage Data settings popup.

Data sharing is now on.

Configure Splunk licenses

How Splunk Enterprise licensing works

When data is sent to the Splunk platform, that data is **indexed** and stored on disk. Part of the indexing process is to measure the volume of data being ingested, and report that volume to the license manager for license volume tracking.

How data is measured

When ingesting **event data**, the measured data volume is based on the raw data that is placed into the indexing pipeline. It is not based on the amount of compressed data that is written to disk. Because the data is measured at the indexing pipeline, data that is filtered and dropped prior to indexing does not count against the license volume quota.

When ingesting **metrics data**, each metric event is measured by volume like event data. However, the per-event size measurement is capped at 150 bytes. Metric events that exceed 150 bytes are recorded as only 150 bytes. Metric events less than 150 bytes are recorded as event size in bytes plus 18 bytes, up to a maximum of 150 bytes. Metrics data draws from the same license quota as event data.

Data that is not measured

The Splunk software troubleshooting and internal communications logs that are indexed into the internal indexes such as `_internal` and `_introspection` do not count against your license volume quota.

The use of **summary indexing** and metric rollup summaries do not count against your license volume quota.

What happens if I exceed my license volume?

License warnings occur when you exceed the indexing volume allowed for your license. The indexing volume is measured daily from midnight to midnight using the system clock on the license manager. See About license violations.

How vCPU is calculated for infrastructure licensing

For Splunk software, a vCPU is any logical CPU core as reported by the host operating system. A vCPU can represent a physical core, a logical core created through the use of hyper-threading or simultaneous multithreading, or a shared logical CPU provided through virtualization. The term vCPU is commonly used when provisioning resources in virtualized environments and in cloud infrastructure allocations; but each implementation of vCPU is unique.

Splunk software uses the CPU's reported by the OS as the total vCPU's for each measured node.

Which nodes are measured for vCPU use?

The total vCPU count across all Splunk Enterprise search heads and indexers count towards the vCPU licensed capacity.

To check the vCPU count in your deployment, use the Resource Usage: CPU Usage dashboards in the Monitoring Console, and filter the report for the search head and indexer roles. See Resource Usage: CPU Usage in the *Monitoring Splunk Enterprise* manual.

License types and license management

There are multiple types of Splunk software licenses available, see Types of Splunk licenses.

To learn about Splunk software license management, see Allocate license volume.

Types of Splunk Enterprise licenses

Each Splunk software instance requires a license. Splunk software licenses specify the features you have access to and how much data can be indexed. As a customer, you'll work with licenses for a Splunk platform instance like Splunk Enterprise. This topic briefly describes different types of licenses you can obtain for Splunk Enterprise software.

This web page provides a brief summary for convenience. It does not create or dictate the terms of any legal contract or license. To fully understand your rights and obligations under any Splunk license (including the licenses listed on this page), consult the license text itself. In the event of any conflict between this summary and the license text, the license text controls.

Splunk Enterprise commercial end-user licenses

Customers can purchase a commercial end-user license to Splunk Enterprise based on either data volume or infrastructure. These Splunk Enterprise licenses are the most common license types. They provide access to the full set of Splunk Enterprise features within a defined limit of indexed data per day (volume-based license), or vCPU count (infrastructure license). Pricing and purchasing information are available on the Splunk website.

The Splunk Enterprise volume-based license

The following important points apply to the volume-based license:

- This volume-based license gives you access to all Splunk Enterprise features.
- This volume-based license allows for both single-instance and distributed installations of Splunk Enterprise.
- The volume-based license can be stacked and assigned to license pools. For information on allocating portions of a license, see Allocate license volume.
- This license cannot stack with an infrastructure-based license (explained in the next section).
- The license prevents searching if your license stack is less than 100 GB of data per day and there are a set number of license warnings. * To learn about license warnings and violation, see What happens during a license violation?

Contact Splunk for information about purchasing this license for Splunk Enterprise.

The Splunk Enterprise infrastructure license

The following important points apply to the infrastructure-based license:

- The infrastructure license gives you access to all Splunk Enterprise features.
- The infrastructure license is for both single-instance and distributed installations.
- The infrastructure licenses can be stacked with one another and assigned to pools. For information on allocating portions of a license, see Allocate license volume.
- The infrastructure license cannot stack with a volume-based license.

Contact Splunk for information about purchasing this license for Splunk Enterprise.

Compare Splunk Enterprise licenses

Consult this table for a comparison of Splunk Enterprise license types:

License conditions	Enterprise: with less than 100 GB of data per day license stack	Enterprise: with 100 GB of data per day or larger license stack	Enterprise: Infrastructure (vCPU)
Currently blocks search while in violation	Yes	No	No
Logs internally and displays message in Splunk Web when in warning or violation	Yes	Yes	No
Stacks with other licenses	Yes	Yes	No
Enables full Splunk Enterprise feature set	Yes	Yes	Yes

Splunk developer licenses

There are two different Splunk developer licenses, the Dev/Test license and the Developer license.

These Splunk developer licenses are for development and testing of content for use with Splunk and cannot be used for production use cases.

Dev/Test license

The Test and Development license ("Dev/Test license") is only available to customers that have acquired a commercial, paid license to Splunk Enterprise and is subject to Splunk General Terms. It's used by customers with pre-production environments to test upgrades and evaluate customized app configuration changes before moving the changes into production. The following important points apply to the Dev/Test license:

- The Dev/Test license cannot be stacked with other licenses. (For example, if you install a Dev/Test license after installing another non-Dev/Test Splunk Enterprise license, the Dev/Test license removes and replaces other Splunk Enterprise license files.)
- The Dev/Test license allows you to index up to the amount of data licensed per the relevant customer order. If you exceed that, you will receive a license warning.
- The Dev/Test license prevents searching if there are a set number of license warnings. To learn about license warnings and violation, see What happens during a license violation?
- Instances of Splunk Enterprise with a Dev/Test license are indicated by a "DEVTEST" stanza in the License XML file.
- The personalized Dev/Test license gives limited access to Splunk Enterprise features.
    - If you are employed by a Splunk customer with an active paid entitlement and wish to obtain a personal license to test Splunk Enterprise for your individual (non-commercial, separate from your organization's license) testing purposes, you may request a personalized Dev/Test license online: Personalized Dev/Test Licenses for Splunk Customers. This personalized license expires after 6 months (unless renewed) and allows you to index up to 50 GB of data per day in single-instance use only. Note that this license is currently not available for other products than Splunk Enterprise and only a limited set of features may be available.

Developer license

The Developer license is available to developers under the Splunk Developer Agreement. The Developer license is for developing content in connection with Splunk Enterprise alongside various development tools so that developers can test that content prior to its release on Splunkbase. This license is different from the license provided to partners participating in our Build Motion Partner program, described in the following section. The following important points apply to the Developer license:

- The Developer license gives you access to various Splunk developer tools on dev.splunk.com and a full set of Splunk Enterprise features.
- The Developer license is for single-instance and distributed installations of Splunk Enterprise.
- The Developer license cannot be stacked with other licenses.
- The Developer license expires after 6 months. You can submit a request to renew your license one week before it expires using the request form at Splunk Developer License Signup.
- The Developer license allows you to index 10 GB of data per day. If you exceed that you will receive a license warning.
- The Developer license prevents searching after a set number of license warnings. To learn about license warnings and violation, see What happens during a license violation?

To request a Developer license, see Splunk Developer License Signup.

Build Partner license

The Splunk Build Program formerly known as the Splunk Technology Alliance Program provides developers an opportunity to become Splunk Partners by signing the Splunk Partnerverse Partner General Terms (PGT) and the Splunk Build Addendum. Build Partners receive all the benefits mentioned in the Partnerverse Program Guide including a continuously renewable (if in good standing) 50 GB NFR license to Splunk Enterprise (and access to other products) and Partnership support via the Splunk Partner Portal & the Splunk Build/TAP Team. If you want to build Splunk-interoperable solutions to commercially market and distribute with Splunk's assistance, this license is the right one for you.

Other types of licenses

Splunk provides other licenses for specific uses:

The Splunk Enterprise Trial license

When you download and install Splunk Enterprise, a Splunk Enterprise Trial license is automatically generated for that instance. The following important points apply to the Enterprise Trial license:

- The Enterprise Trial license gives access to all Splunk Enterprise features.
- The Enterprise Trial license is for standalone, single-instance installations of Splunk Enterprise only.
- The Enterprise Trial license cannot be stacked with other licenses.
- The Enterprise Trial license expires 60 days after you install the Splunk Enterprise instance, unless otherwise specified to customers.
- The Enterprise Trial license allows you to index 500 MB of data per day to Splunk Enterprise. If you exceed that limit you receive a license warning.
- The Enterprise Trial license prevents searching if there are a set number of license warnings. To learn about license warnings and violation, see What happens during a license violation?

If you want to set up a trial Splunk Enterprise distributed deployment consisting of multiple Splunk Enterprise instances

communicating with each other, each instance must use its own self-generated Enterprise Trial license. You cannot use centralized license management with the Enterprise Trial license.

If you need a more customized version of a trial license to prepare a pilot or proof of concept for a large deployment or with a longer duration or indexing volume, Contact Splunk or your sales representative with your request.

Free license

The Free license allows a completely free Splunk Enterprise instance with limited functionality and license usage. The following important points apply to the Free license:

- The Free license gives access to a limited set of Splunk Enterprise features.
- The Free license is for a standalone, single-instance installation of Splunk Enterprise only.
- The Free license cannot be stacked with other licenses.
- The Free license does not expire.
- The Free license allows you to index 500 MB of data per day. If you exceed that you will receive a license warning.
- The Free license prevents searching if there are a number of license warnings. To learn about license warnings and violation, see What happens during a license violation?

For a list of features that are disabled in Splunk Free, see About Splunk Free.

Pre-release license

Splunk invites certain customers to participate in its pre-release programs from time to time. These pre-release licenses are subject to certain beta terms similar to those in the Splunk Pre-release Software License Agreement, where users are limited to using a newer version of Splunk Enterprise for testing and evaluation purposes only. These licenses are not compatible with other Splunk software releases. Pre-release licenses typically enable specific Splunk Enterprise features for a specified testing duration.

Licenses and distributed deployments

Distributed Splunk Enterprise deployments consist of multiple Splunk Enterprise instances. Separate instances perform various functions such as indexing and search management. Each instance is categorized as one or more **component** types, based on the functions that it performs. See Scale your deployment with Splunk Enterprise components and Components that help to manage your deployment in *Distributed Deployment*. In most cases, an instance serves as just a single component, but it is possible for an instance sometimes to combine the functionality of several components.

This topic does not pertain to standalone Splunk Enterprise deployments, which consist of a single Splunk Enterprise instance plus forwarders. For a standalone deployment, simply install the appropriate license directly on the instance. See Install a license.

License requirements

All Splunk software instances must have a license.

- Splunk Enterprise instances need access to an Enterprise license unless they are functioning only as forwarders. The license access is required even when they do not index external data. Access to specific features of a distributed deployment, such as **distributed search** and **deployment server** are only available with Enterprise

licenses. The recommended way to connect instances to an Enterprise license is to associate the instance with a license manager. See Configure a license peer.

- Universal forwarders only need a Forwarder license. If a heavy forwarder is performing additional functions such as indexing data or managing searches, it requires access to an Enterprise license.

This table provides a summary of the license needs for the various Splunk Enterprise component types.

Component type	License type	Notes
Indexer	Enterprise	
Search head	Enterprise	
Deployment server	Enterprise	
Indexer cluster manager node	Enterprise	
Search head cluster deployer	Enterprise	
Monitoring console	Enterprise	
Universal forwarder	Forwarder	
Light forwarder	Forwarder	
Heavy forwarder	Enterprise or Forwarder	Heavy forwarders that index data or use other Splunk Enterprise features need access to an Enterprise license.

Components and licensing issues

Indexers

The **Indexers** index, store, and search external data.

To participate in a distributed deployment, indexers need access to an Enterprise license. The data that indexers ingest is metered against the license.

Search heads

A **search head** is a Splunk Enterprise instance that manages searches.

Search heads need access to an Enterprise license.

Forwarders

Forwarders ingest data and forward that data to another forwarder or an indexer. Because data is not metered until it is indexed, forwarders do not incur license usage.

In most distributed deployments, forwarders only need a Forwarder license.

There are several types of forwarders:

- The **universal forwarder** has the Forwarder license applied automatically.
- The **light forwarder** must be changed manually to another license type. You can use the Forwarder license, but you must manually enable it by changing to the Forwarder license group.

- The **heavy forwarder** must be changed manually to another license type. If the heavy forwarder will be performing indexing or using other Enterprise features, it must be connected to a **license manager** node.

A forwarder can use the Free license instead of a Forwarder license, but some critical functionality is unavailable with a Free license. For example, a forwarder using a Free license cannot be a deployment client and it does not offer any authentication.

Management components

All Splunk Enterprise instances functioning as management components need access to an Enterprise license.

Management components include the **deployment server**, the **indexer cluster manager node**, the **search head cluster deployer**, and the **monitoring console**. For information on management components, see Components that help to manage your deployment.

Clustered deployments and licensing issues

Indexer cluster nodes

Each indexer cluster node requires an Enterprise license. There are a few license issues that are specific to indexer clusters:

- Cluster nodes must all share the same licensing configuration.
- Only incoming data counts against the license; replicated data does not.

Search head cluster members

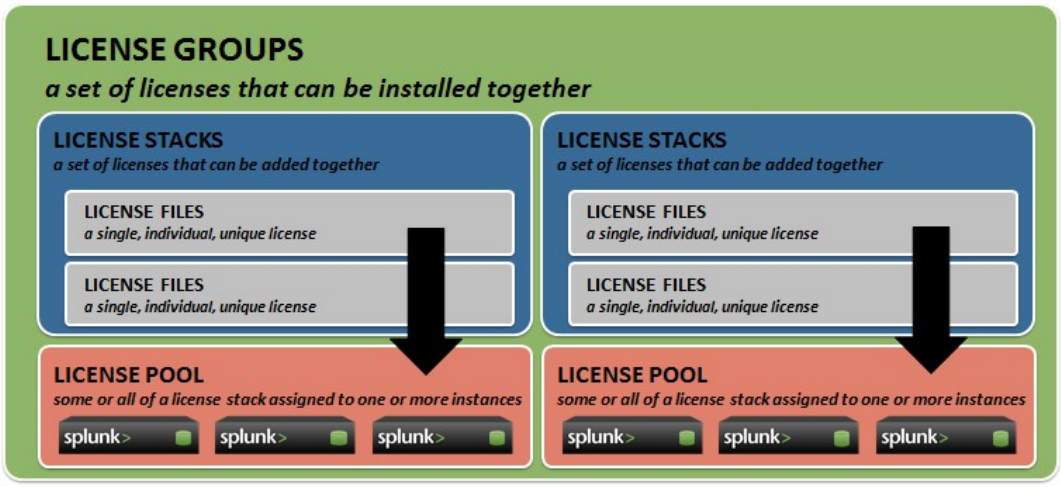
Each search head cluster member needs access to an Enterprise license. The search head cluster **deployer**, which distributes apps to the members, also needs access to an Enterprise license.

Allocate license volume

Splunk Enterprise licensing management uses logical license groupings to allow for multiple licenses and license assignments, and to monitor the license usage.

The **license manager** is a Splunk Enterprise **component** used to manage **licenses** and assign license volume.

Use the license manager to **group** licenses, and assign them to **stacks**. You can create license **pools** from the stacks, and assign the **license peers** to a pool so they can use Splunk Enterprise features and have their license usage levied against a pool.



Groups

A **license group** represents a set of license stacks:

- Only one license group is active at a time.
- A license group can contain zero to many license stacks.
- A license manager can only administer one license group at a time.

The license groups are:

- Enterprise/Sales Trial group -- This group contains Enterprise licenses and Sales Trial licenses. You can stack these licenses.
- Enterprise Trial group -- This is the default group when you first install a new Splunk Enterprise instance. If you switch an instance to a different group, you cannot switch back to the Enterprise trial group. You cannot stack Enterprise trial licenses.
- Free group -- This group accommodates Splunk Free installations. When an Enterprise Trial license expires after 60 days, that Splunk instance is converted to the Free group. You cannot stack Splunk Free licenses.
- Forwarder group -- This group is for forwarders that function solely as forwarders and do not perform other roles, such as indexing. You cannot stack Forwarder licenses.

Subgroups

The license subgroup is used to further categorize license types, and is set inside the license. There are several subgroups, including DevTest and Production. A license belongs to a single subgroup.

Stacks

A **stack** is one or more licenses that allow their assigned license volume to be added together. Enterprise licenses and Sales Trial licenses can be stacked together, and with each other. This allows you to increase indexing volume capacity without the need to swap out licenses. As you purchase additional capacity, just add the license to the appropriate stack.

The daily license volume is tracked at the stack and pool level. If your daily data ingest exceeds the assigned license volume, you will receive warnings at the stack or pool level depending upon how the license volume was allocated. See

About license violations.

A stack contains one or more license pools, with each pool having a portion of the stack's total licensing volume. Stacks and pools are not available with these license types:

- Enterprise Trial
- Free
- Dev/Test. If you install a Dev/Test license over an Enterprise license, the Enterprise license will be deleted.
- Forwarder

Pools

A **pool** contains some or all of a stack's license volume. If you have a volume-based license, you can manage license volume usage by creating multiple pools and assigning Splunk Enterprise components to specific pools. The components must be configured as license peers to the license manager, and assigned to a pool.

For example, if you create a license pool used for production indexers, and use a separate license pool for the test indexers' you will ensure that testing activity does not impact production license needs. Each indexer is made a license peers to the license manager, and the indexers are assigned to the appropriate pool; some to production and some to test.

Other components must be assigned to a license pool so that they are permitted access to Splunk Enterprise features, such as distributed search. As a general rule, assign all of your Splunk Enterprise instances to a license pool, with the exception of universal forwarders. See Licenses and distributed deployments.

License manager

A **license manager** is a Splunk Enterprise component that hosts licenses and allows you to configure license volume assignments to license peers. You will use the license manager to define pools, add licensing capacity, and manage license peers by adding them to pools. In a distributed infrastructure, there is typically one designated license manager.

License peers

A **license peer** is a Splunk Enterprise instance that connects to the license manager to receive license validation and a license volume assignment. A license peer is assigned to a single license pool. For example, indexers, search heads, and heavy forwarders all use features that require an Enterprise license. By configuring those components as license peers to the license manager, they have full access to the Splunk Enterprise features and license volume as needed.

Configure a license manager

A license manager is a central license repository for Splunk Enterprise licenses. Once the license manager is configured, you can direct other Splunk Enterprise instances to communicate with the license manager to provide them access to the license features, and allocate license volume. The remote instances become **license peers**

If you only have a single Splunk Enterprise instance, you do not need to configure it as a license manager.

Choose the instance to serve as the license manager

The license manager role is not typically run on a dedicated Splunk Enterprise instance. Instead, you can colocate it on an instance that is also performing other tasks:

- A **monitoring console**. See Which instance should host the console? in *Monitoring Splunk Enterprise* for a description of the circumstances under which a monitoring console and a license manager can colocate.
- A **deployment server**. See Deployment server and other roles in *Updating Splunk Enterprise Instances* for a description of the circumstances under which a deployment server and a license manager can colocate.
- An **indexer cluster manager node**. See Additional roles for the manager node in *Managing Indexers and Clusters of Indexers* for a description of the circumstances under which an indexer cluster manager node and a license manager can colocate.
- A **search head cluster deployer**. See Deployer requirements in *Distributed Search*.
- A **search head**.
- An **indexer**. If a license manager is located on an indexer, it will become the indexer's license manager.

For a general discussion of management component colocation, see Components that help to manage your deployment in the *Distributed Deployment Manual*.

Configure the license manager

To configure the license manager:

1. Install your Enterprise licenses onto the license manager. See Install a license.
2. Configure the license peers to communicate with the license manager. See Configure a license peer.
3. Review the license allocation on the license manager, and create pools to allocate license volume. See Create or edit a license pool.
4. Verify the license capacity usage and indexing volume the next day. See About the Splunk Enterprise license usage report view.

License manager and peer version compatibility

A license manager must be on the same or later version than its license peers. For example, an 8.2.x license manager is compatible with 7.3, 8.0, and 8.1 peers.

Compatibility is significant at the major/minor release level, but not at the maintenance level. For example, a 7.2 license manager is not compatible with a 7.3 license peer, because the 7.2 is a lower minor release level than 7.3. However, a 7.3.1 license manager is compatible with a 7.3.3 license peer, despite the lower maintenance release level.

Install a license

This topic describes how to install new Enterprise licenses.

If you install a Dev/Test license over an Enterprise license, it replaces the Enterprise license.

Install a license for a distributed deployment

To install a license for a distributed deployment of Splunk Enterprise:

1. Choose an instance to function as the license manager, if you have not already done so. See Configure a license manager.
2. On the license manager, navigate to **Settings > Licensing**.
3. Click **Add license**.
4. Do one of the following:
    1. Click **Choose file** and browse for your license file and select it, or
    2. Click **copy & paste the license XML directly...** and paste the text of your license file into the provided field.
5. Click **Install**.
6. If this is the first Enterprise license that you are installing on the license manager, you must restart Splunk Enterprise.

Install a license for a standalone instance

To install a license for a standalone instance of Splunk Enterprise:

1. On the instance, navigate to **Settings > Licensing**.
2. Click **Add license**.
3. Do one of the following:
    1. Click **Choose file** and browse for your license file and select it, or
    2. Click **copy & paste the license XML directly...** and paste the text of your license file into the provided field.
4. Click **Install**.
5. If this is the first Enterprise license that you are installing on the instance, you must restart Splunk Enterprise.

Add a note to a license file

Once an Enterprise license is installed, you can add a note or other text to your license file:

1. Navigate to **Settings > Licensing**.
2. Under **Licenses**, click **Notes**.
3. In the Notes field, add a note or other text.
4. Click **Save**.

The Notes field is only available for licenses installed in an Enterprise license group.

Configure a license peer

This topic discusses configuring a Splunk Enterprise instance as a **license peer**. Before you proceed, review these topics:

- Read Allocate license volume for general information about allocating license volume across Splunk Enterprise instances.
- Read Configure a license manager for instructions on configuring a license manager.

Configure the instance as a license peer

1. Log into Splunk Web and navigate to **Settings > Licensing**.
2. Click **Change to Peer**.
3. Switch the radio button from **Designate this Splunk instance as the license server** to **Designate a different Splunk instance as the license server**.
4. Specify the license manager. You must provide an IP address or a hostname, and include the management port. The default management port is 8089.
5. Click **Save**.
6. Restart Splunk Enterprise services.

Use the command line to manage licenses

You can also use the command line to configure a license peer. For examples, see Manage license peers.

Revert a license peer to a standalone license

A standalone instance uses a locally installed license. You will need a Splunk Enterprise license to revert a license peer to an independent instance.

1. Log into Splunk Web, and navigate to **Settings > Licensing**.
2. Click **Switch to local manager**.

If this instance does not already have an Enterprise license installed, you must restart Splunk for this change to take effect.

Create or edit a license pool

This topic describes how to create or edit a **license pool**. Before you proceed, read Allocate license volume for general information about allocating license volume across Splunk Enterprise instances.

Note: You can also perform these tasks through the CLI. See Manage licenses from the CLI.

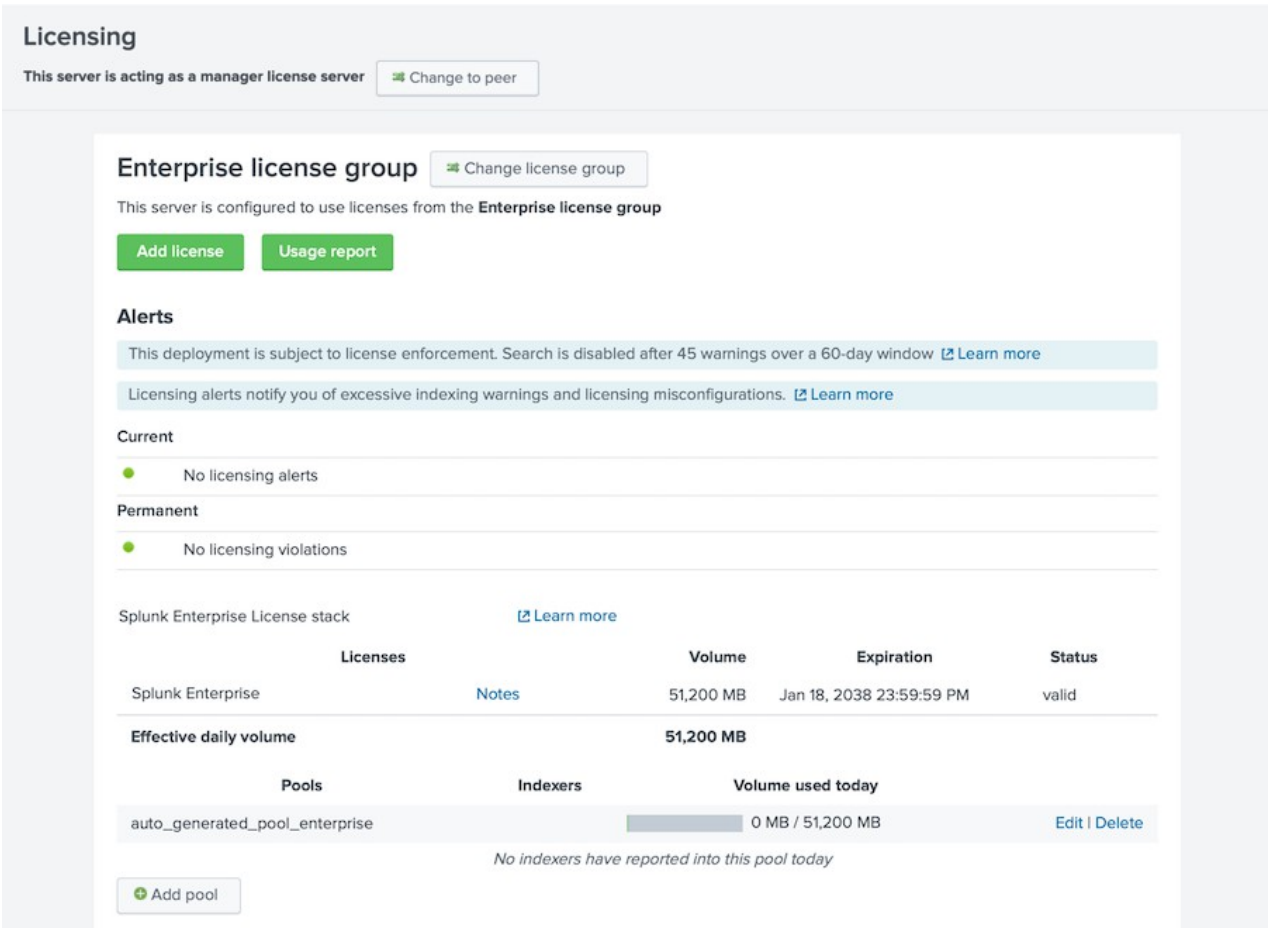
The default license pool

When you first install an Enterprise license on a Splunk Enterprise instance, the instance becomes the license manager for that license. Several default configurations result:

- The license resides in a **license stack** called Splunk Enterprise Stack
- The stack has a default license pool called `auto_generated_pool_enterprise`.
- Any license peer that connects to this license manager has access to the default pool.

You can change the set of pools. You can also configure access of license peers to stacks.

The following example shows the **Settings > Licensing** screen for an Enterprise license.


Licensing
This server is acting as a manager license server
Change to peer
Enterprise license group
Change license group
This server is configured to use licenses from the Enterprise license group
Add license
Usage report
Alerts
This deployment is subject to license enforcement. Search is disabled after 45 warnings over a 60-day window Learn more
Licensing alerts notify you of excessive indexing warnings and licensing misconfigurations. Learn more
Current
No licensing alerts
Permanent
No licensing violations
Splunk Enterprise License stack
Learn more
Licenses
Volume
Expiration
Status
Splunk Enterprise
Notes
51,200 MB
Jan 18, 2038 23:59:59 PM
valid
Effective daily volume
51,200 MB
Pools
Indexers
Volume used today
auto_generated_pool_enterprise
0 MB / 51,200 MB
Edit | Delete
No indexers have reported into this pool today
Add pool

Edit an existing license pool

You can edit a license pool to change the pool's allocation or to change the set of indexers that have access to the pool.

1. Next to the license pool that you want to edit, click **Edit**. The Edit license pool page is displayed.
2. (Optional) Change the allocation for the pool. The allocation is how much of the stack's overall licensing volume is available for use by the indexers that access this pool. The allocation can be a specific value, or it can be the entire amount of indexing volume available in the stack, as long as it is not already allocated to any other pool.
3. (Optional) Change the indexers that have access to the pool. The options are:
    1. Any indexer configured as a license peer can access the pool and use the license allocation within it.
    2. Only specific indexers can access the pool and use the license allocation within it. To allow a specific indexer to draw from the pool, click the plus sign next to the name of the indexer in the list of available indexers to move it into the list of associated indexers.
4. Click **Submit**.

Once a license pool is created, there's no option to rename the pool using License Management in Splunk Web. To modify the license pool name, you can delete the old pool and create a new pool with the chosen name. Or you can edit the server.conf file, change the pool name in the [lmpool:] stanza, and restart Splunk Enterprise services.

Create a new license pool

Before you can create a new license pool from the default Enterprise stack, you must make some indexing volume available by either editing an existing pool and reducing its allocation, or by deleting an existing pool entirely. Click **Delete** next to the pool's name to delete it.

To create a new license pool:

1. Click **Add pool** toward the bottom of the page. The Create new license pool page is displayed.
2. Specify a name for the pool, and optionally add a description for the pool.
3. Set the allocation for the pool. The allocation is how much of the stack's overall licensing volume is available for use by the indexers that access this pool. The allocation can be a specific value, or it can be the entire amount of indexing volume available in the stack, as long as it is not already allocated to any other pool.
4. Specify the indexers that have access to the pool. The options are:
    1. Any indexer configured as a license peer can access the pool and use the license allocation within it.
    2. Only specific indexers can access the pool and use the license allocation within it. To allow a specific indexer to draw from the pool, click the plus sign next to the name of the indexer in the list of available indexers to move it into the list of associated indexers.

About Splunk Free

If you want to run Splunk Enterprise to practice searches, data ingestion, and other tasks without worrying about a license, Splunk Free is the tool for you.

- The Free license gives very limited access to Splunk Enterprise features.
- The Free license is for a standalone, single-instance use only installation.
- The Free license does not expire.
- The Free license allows you to index 500 MB per day. If you exceed that you will receive a license violation warning.
- The Free license will prevent searching if there are a number of license violation warnings.

Is Splunk Free for you?

The major limitations of Splunk Free are the license volume restriction and removed features.

- Will you ingest less than or up to 500 MB per day of data? At that volume of data per day, you will use around 7GB of storage space per month.
- Are you planning to ingest a large (over 500 MB per day) data set only once, and then analyze it? The Splunk Free license lets you bulk load a much larger data sets up to 2 times within a 30 day period. This can be useful for forensic review of large data sets.
- The Free license will prevent searching if there are 3 license warnings in a rolling 30 day window. If that happens, Splunk Free continues to index your data but disables search functionality. You will regain search when you are below 3 license violation warnings in a 30 day period. See About license violations.

What features are disabled on Splunk Free?

Splunk Free is for standalone, single-instance use only installations. Most Splunk Enterprise features are available on the Free license, with the following exceptions:

- Ingest actions is not available.
- Alerting (monitoring) is not available.
- There are no users or roles. This means:
  - There is no login. You are passed straight into Splunk Web as an administrator-level user.
  - The command line or browser can access and control all aspects of Splunk Free with no user and password prompt.
  - There is only the admin role, and it is not configurable. You cannot add roles or create user accounts.
  - Restrictions on search, such as user quotas, maximum per-search time ranges, and search filters are not supported.
  - Features that rely on user authentication do not work.
- Distributed search configurations including search head clustering are not available.
- Deployment management capabilities are not available.
- Indexer clustering is not available.
- Forwarding in TCP/HTTP formats is not available. This means you can forward data from a Free license instance to other Splunk platform instances, but not to non-Splunk software.
- Report acceleration summaries are not available.

How do I get Splunk Enterprise with the Free license?

1. Create your user account on splunk.com.
2. Review the list of supported operating systems for the "Free" license in Supported Operating Systems.
3. Download the latest version of Splunk Enterprise for your operating system from Free Trials and Downloads on splunk.com. Login required.
4. Use the installation instructions for your operating system. See Installation instructions.
    1. After installation, you'll have an Enterprise Trial license for 60 days. You can change to the Free license at any point before the Enterprise Trial is complete. See Switching to Free from an Enterprise Trial license.
5. If this is the first time you have installed Splunk Enterprise, see the *Search Tutorial* to learn how to index data into Splunk software and search that data using the Splunk Enterprise search language.

Switching to Free from an Enterprise Trial license

When you first download and install Splunk Enterprise, an Enterprise Trial license is created and enabled by default. You can continue to use the Enterprise Trial license until it expires, or switch to the Free license right away depending on your requirements.

What you should know about switching to Free

Splunk Enterprise Trial gives you access to a number of features that are not available in Splunk Free. When you switch, **be aware of the following**:

- Any alerts you defined no longer trigger. You **no longer receive alerts** from Splunk software. You can still schedule searches to run for dashboards and summary indexing purposes.
- Configurations in outputs.conf to forward to third-party applications in TCP or HTTP formats do not work.
- User accounts or roles that you created no longer work.
    - Anyone connecting to the instance will automatically be logged on as admin. You will no longer see a login screen.
- Any knowledge objects created by any user other than admin (such as event type, transaction, or source type definitions) and not already globally shared will not be available. If you need these knowledge objects to continue to be available after you switch to Splunk Free, you can do one of the following:
    - Use Splunk Web to promote them to be globally available before you switch. See Manage app and add-on objects.
    - Hand edit the configuration files they are in to promote them. See App architecture and object ownership.

When you attempt to make any of the above configurations in Splunk Web while using an Enterprise Trial license, you will be warned about the limitations in Splunk Free.

How do I switch to the Splunk Free license?

You can change from the Enterprise Trial license to a Free license at any time. To switch licenses:

1. Log in to Splunk Web as a user in the admin role
2. Select **Settings > Licensing**
3. Click **Change License Group**
4. Select **Free license**
5. Click **Save**
6. You are prompted to restart

If your Enterprise Trial license has expired, use the above procedure except that you can only log into Splunk Web as the admin user. No other credentials will work.

If you need to reset your administrator account, see Unlock a user account in the *Securing the Splunk Platform* manual.

Switching to the Free license removes all authentication and the ability to create or define users. Once the services are restarted, there's no Splunk Web login page displayed. You are passed straight into Splunk Web as an administrator-level user.

Manage Splunk licenses

Delete a license

If a license expires, you can delete it. To delete a license:

1. On the license manager, navigate to **System > Licensing**.
2. Click **Delete** next to the license you want to delete.
3. Click **Delete** again to confirm.

You cannot delete the last license in a list of licenses.

Swap the license manager

You can change the license manager by promoting a license peer to be the manager, and demoting the old manager to a license peer.

Review the license configuration on the license manager:

1. Verify the number of active licenses, and license capacity.
2. Verify the pool allocation and the pool assignments for all license peers.
3. Get a copy of your licenses from the support portal, if needed. See Working with Support and the Support Portal.

Promote a license peer to be the manager:

1. On the peer, navigate to **Settings > Licensing**.
2. Click **Switch to local manager**.
3. On the Change manager association page, choose **Designate this Splunk instance as the manager license server**.
4. Click **Save**.
5. Restart the Splunk Enterprise services.
6. On the new license manager, install your licenses. See Install a license.

Configure the license peers to use the new license manager:

1. On the peer, navigate to **Settings > Licensing**.
2. Click **Switch to local manager**.
3. Update the **Manager license server URI** to point at the new license manager.
4. Click **Save**.
5. Restart the Splunk Enterprise services.

Demote the old license manager to be a peer:

1. On the old license manager, navigate to **Settings > Licensing**.
2. Click **Change to peer**.
3. Click **Designate a different Splunk instance as the manager license server**.
4. Update the **Manager license server URI** to point at the new license manager.

5. Click **Save**.
6. Stop the Splunk Enterprise services.
7. Using the CLI, delete any license files under `$SPLUNK_HOME/etc/licenses/enterprise/`.
8. Start the Splunk Enterprise services.

Manage licenses from the CLI

This topic describes how to use Splunk Enterprise command line (CLI) to monitor and manage your licenses. It covers some of the common uses and options available for managing licenses. The definitive reference to any CLI command is the command's online help.

For general information on the Splunk CLI, see "About the CLI".

For information on managing licenses through Splunk's REST API, refer to "Licenses" in the REST API Reference Manual.

CLI license commands and objects

You can use the CLI to add, edit, list, and remove licenses and license-related objects. The available commands are:

Command	Object(s)	Description
add	licenses, licenser-pools	Add a license or a pool of licenses to a license stack. This command is only available if you have an Enterprise license.
edit	licenser-groups, licenser-localpeer, licenser-pools	Edit the attributes of a license group, peers, or pools of licenses. Some license commands are only available if you have an Enterprise license.
list	licenser-groups, licenser-localpeer, licenser-messages, licenser-pools, licenser-peers, licenser-stacks, licenses	Depending on the object specified, lists either the attributes of that object or members of that object.
remove	licenser-pools, licenses	Remove licenses or license pools from a license stack.

License-related objects are:

Object	Description
licenser-groups	The set of available license groups. This includes Enterprise, Forwarder, and Free.
licenser-localpeer	A local license peer's configuration.
licenser-messages	Any alerts or warnings about the state of your licenses.
licenser-pools	The set of license pools.
licenser-peers	All the peers that have contacted the license manager.
licenser-stacks	A stack of licenses.
licenses	The set of licenses for this Splunk instance.

Common license-related tasks

The following are examples of common license-related tasks that you can perform with the CLI.

Manage licenses

Splunk Enterprise stores applied licenses in the `$SPLUNK_HOME/etc/licenses/enterprise/` directory.

To add a new license to the license stack, specify the path to the license file:

```
splunk add licenses <path_to_license>/<license_name>.xml
```

To list all the licenses in a license stack:

```
splunk list licenses
```

The `splunk list` command also displays the properties of each license, including the features it enables (`features`), the license group and stack it belongs to (`group_id, stack_id`), the indexing quote it allows (`quota`), and the license key that is unique for each license (`license_hash`).

If a license expires, you can remove it from the license stack. To remove a license from the license stack, specify the license's hash:

```
splunk remove licenses BM+S8VetLnQEb1F+5Gwx9rR4M4Y91AkIE=781882C56833F36D
```

Manage license pools

You can create a license pool from licenses in a license stack (if you have an Enterprise license). A license stack can be divided into multiple license pools. Multiple license peers can share the quota of the pool.

To see all the license pools in all the license stacks:

```
splunk list licenser-pools
```

To add a license pool to the stack, you need to: name the pool, specify the stack that you want to add it to, and specify the indexing volume allocated to that pool:

```
splunk add licenser-pools pool01 -quota 10mb -peers guid1,guid2 -stack_id enterprise
```

You can also specify an optional description for the pool, and any peers that are members of the pool.

You can edit the license pool's description, indexing quota, and peers. For example, assuming you created pool01 in the previous example:

```
splunk edit licenser-pools pool01 -description "Test" -quota 15mb -peers guid3,guid4 -append_peers true
```

This adds a description for the pool, "Test", changes the quota from 10mb to 15mb, and adds peers guid3 and guid4 to the pool. The peers with guid1 and guid2, which you added in the previous example, continue to have access to the pool.

To remove a license pool from a stack:

```
splunk remove licenser-pools pool01
```

Manage license peers

A license peer uses license quota from one or more license pools. The license manager controls access .

To list all the license peers that have contacted the license manager:

```
splunk list licenser-peers
```

To list all the properties of the local license peer:

```
splunk list licenser-localpeer
```

To add a license peer, edit the attributes of that local license peer node (specify the uri of the license manager or 'self'):

```
splunk edit licenser-localpeer -manager_uri 'https://<license_manager_host>:<port>'
```

Monitor license status

You can use the `splunk list` command to view messages (alerts or warnings) about the state of your licenses.

```
splunk list licenser-messages
```

Select a different license group

You can change the license group assigned to a Splunk Enterprise instance. For example:

License group	Example
Free	`splunk edit licenser-groups -name "Free" -is_active "1"`
Forwarder	`splunk edit licenser-groups -name "Forwarder" -is_active "1"`
Enterprise	`splunk edit licenser-groups -name "Enterprise" -is_active "1"`

Changing the license group requires a restart of the Splunk Enterprise services.

Choosing the Free or Forwarder license group automatically applies the associated license to the Splunk Enterprise instance. Using a Free or a Forwarder license changes the behavior of your Splunk Enterprise instance, and limits the functionality based upon the restrictions for those license types. To review the license limitations, see Types of Splunk Enterprise licenses.

> Authentication is required to switch license groups, except when moving from the Free group.

If you need to reset your administrator account, see Unlock a user account in the *Securing the Splunk Platform* manual.

About license violations

A license violation occurs after a series of license warnings. License warnings occur when you exceed the maximum daily indexing volume allowed for your license. If you have multiple license warnings, and have exceeded the license warning limit for your license, you will receive a license violation.

What is a license warning?

License warnings occur when you exceed the maximum daily indexing volume allowed for your license. Here are the conditions:

- Your daily indexing volume is measured from midnight to midnight using the system clock on the **license manager**.
- If you exceed your licensed daily volume in any single calendar day, you generate a license warning.
- If you generate a license warning, you have until midnight on the license manager to resolve the warning before it counts against the total number of warnings allowed by your license. For guidance on what to do when a warning appears, see Correcting license warnings.

What do license warnings look like?

A license warning appears as an administrative message in Splunk Web. Clicking the link in the message takes you to the **Licensing** page, where the warning appears under **Alerts**.

These are some of the conditions that generate a license warning:

- When a license pool has reached its daily license volume limit.
- When a license stack has reached its daily license volume limit.
- When a **license peer** is unable to communicate with the license manager. To troubleshoot a communications issue, see Violations due to broken connections between license manager and peers.

What happens during a license violation?

A license violation happens when you exceed the number of warnings allowed on your license. The license violation conditions are based upon the license type.

Here is what happens to indexing and search capability during a license violation:

- For license stacks with a licensed volume of less than 100 GB per day, using search is blocked while you are in violation. This restriction includes scheduled reports and alerts.
- Splunk Enterprise continues to index your data.
- Searching the internal indexes is not blocked. You can use the monitoring console or run searches against the `_internal` index to diagnose the licensing problem.
- If you're using a license manager, a message will appear in the Search app, and in Global Messages navigation bar on the search heads notifying all users that their license is invalid, or has expired.

Here is a table of license violation conditions by Splunk Enterprise license type:

License	Violation conditions

License	Violation conditions
Splunk Enterprise license	For Splunk Enterprise license stacks with a licensed volume of 100 GB per day or higher, warnings are issued when the system exceeds its daily licensed capacity. Search is not disabled. If you have a license stack with less than 100 GB of data per day of license volume, and you generate 45 license warnings in a rolling 60 day period, you are in violation of your license. If that license stack is split into multiple pools, search is disabled for a pool and its license pool member(s) after 45 warnings over a rolling 60-day window. Other pools and their members will remain searchable if the usage across the remaining license pools does not exceed their allocated license. To reenable search, request a reset license from Splunk Sales.
Splunk Enterprise infrastructure license	An Enterprise license based on vCPU usage does not currently violate.
Splunk Enterprise Trial license	If you generate five or more warnings in a rolling 30-day period, you are in violation of your license. Splunk Enterprise continues to index your data, but you cannot search it. The warnings persist for 14 days. No reset license is available.
Dev/Test license	If you generate five or more warnings in a rolling 30-day period, you are in violation of your license. Splunk Enterprise continues to index your data, but you cannot search it. The warnings persist for 14 days. To enable searching, request a reset license using the request form at Personalized Dev/Test Licenses for Splunk Customers.
Developer license	If you generate five or more warnings in a rolling 30-day period, you are in violation of your license. Splunk Enterprise continues to index your data, but you cannot search it. The warnings persist for 14 days. To enable searching, request a reset license by emailing `devinfo` at splunk.com.
Free license	If you generate three or more warnings in a rolling 30-day period, you are in violation of your license. Splunk Enterprise continues to index your data, but you cannot search it. The warnings persist for 14 days. No reset license is available.

Violations due to broken connections between license manager and peers

A license peer transmits its license volume usage to the license manager every minute. If a license peer cannot communicate with the license manager for 72 hours or more, the peer is placed in violation, and search is blocked. A violation still allows indexing to continue. You cannot search a peer in violation until it is reconnected with the license manager.

To find out if a license peer is unable to reach the license manager, search for an error event in the `_internal` index or the license peer's splunkd.log:

```
index=_internal LMTracker error "failed to send rows" OR "unable to connect"
```

Avoiding license warnings

To avoid license warnings, monitor the license usage over time and ensure that you have sufficient license volume to support your daily license use:

- Use the license usage report view on the license manager to troubleshoot index volume. See About the Splunk Enterprise license usage report view.
- Enable an alert on the monitoring console to monitor daily license usage. See Platform alerts in *Monitoring Splunk Enterprise*.

Correcting license warnings

If you receive a message to correct a license warning before midnight, you have already exceeded your license quota for the day. This warning is issued to make you aware of the license use and to provide you time to change or update your license configuration. The daily license volume quota resets at midnight on the license manager, and at that point the warning is recorded as a license warning. Most licenses allow for a limited number of warnings before a violation occurs.

Once data is indexed, you cannot change the volume recorded against your license. You can't un-index data. Instead, you need to gain additional license volume using one of these options:

- If you have another license pool with extra license volume, reconfigure your pools and move license capacity where you need it.
- Purchase more licenses and add them to the license stack and pool.

If you cannot use either of those options, you can analyze your indexing volume and make a change to reduce the data sources that are using more license than usual. To learn which data sources are contributing the most to your license quota, see the license usage report view.

Once you identify a data source that is using a lot of the licensed volume, you can determine how to manage the data to correct the license warnings:

- Determine if this was a one-time data ingestion issue. For example, debug logging was enabled on the application logs to troubleshoot an issue, but the logging-level will be reset tomorrow.
- Determine if this is a new average license usage based upon changes in the infrastructure. For example, a new application or server cluster came online, and the team didn't update you before ingesting their data.
- Determine if you can filter and drop some of the incoming data. For examples of drop filters, see Route and filter data in the *Forwarding Data* manual.

License usage report view

About the Splunk Enterprise license usage report view

When you want to view and monitor your license capacity usage and indexing volume over time, use the license usage reports. These reports are available on both the license manager and the monitoring console roles. To learn about license allocation, and license stacks and pools, see Allocate license volume.

Access the license usage report view

On the license manager:

1. Navigate to **Settings > Licensing**.
2. Select **Usage report**.

On the monitoring console:

1. Navigate to **Settings > Monitoring Console**.
2. Navigate to **Indexing > License Usage**.
3. Select **License Usage**.

If you use infrastructure licensing, use the Resource Usage: CPU Usage dashboards in the Monitoring Console to check your vCPU counts for the search head and indexer roles. See Resource Usage: CPU Usage in the *Monitoring Splunk Enterprise* manual.

License Usage - Today

The panels in this report show the status of your license usage, and any warnings for the current day. The panels include:

Panel name	Description
Today's license usage (GB)	Today's license usage and the total daily license quota across all pools.
Today's license usage per pool	Today's license usage and the daily license quota for each pool.
Today's percentage of daily license quota used per pool	The percentage of today's license quota used by each pool. The percentage is displayed on a logarithmic scale.
Pool usage warnings	Displays any warnings that a pool has received in the past 30 days, or since the last license reset key was applied. See "About license violations".
Peer usage warnings	The pool membership, the number of warnings, and any violations recorded for each license peer.

License Usage - Previous 30 Days

The panels in this report show the historical license usage, and any warnings. The report uses data collected from the `license_usage.log`, message `type=RolloverSummary`. These represent the daily totals recorded for all peer nodes.

If the license manager is down or inaccessible during the time period that represents midnight using the system clock, the license manager will not generate a RolloverSummary event for that day, and you will not see that day's data in

these panels.

The License Usage report will change to "Previous 60 Days" if your Splunk Enterprise license stack is less than 100GB, and is subject to conditional license enforcement.

The panels include:

Panel name	Split by	Description
Daily License Usage	Yes: pool, indexer, source type, host, source, index.	The total daily license usage over time. Use the split-by option to sort.
Percentage of Daily License Quota Used	Yes: pool, indexer, source type, host, source, index.	The percentage of the daily license quota used over time. Use the split-by option to sort.
Average and Peak Daily Volume	Yes: pool, indexer, source type, host, source, index.	The average and peak license usage over time. Use the split-by option to sort.

The visualizations in these panels limit the number of values plotted for each field that you can split by host, source, source type, index, indexer, or pool. If you have more than 10 distinct values for any of these fields, the values after the 10th are labeled "Other."

Improve performance by accelerating reports

By default, generating a historical report using a split-by field with many values will take time to run. You can accelerate the report If you plan to run it regularly.

Enable report acceleration only on the instance where you plan to view the licensing report, such as the license manager or the monitoring console.

When you use the split by option for source type, host, source, or index; you'll be prompted to turn on report acceleration. You can view the options and schedule for accelerating licensing searches in **Settings > Searches, Reports, and Alerts > License Usage Data Cube**. Report acceleration can take up to 10 minutes to start after you select it for the first time. After the historical data has been summarized, the data is kept current using a scheduled report. See Accelerate reports in the *Reporting Manual*.

Squashing fields

Each license peer periodically reports the stats for data indexed by source, source type, host, and index to the license manager. If the number of distinct tuples (host, source, sourcetype, index) grows beyond a configurable threshold, the host and source values are automatically squashed. This is done to lower memory usage, and prevent a flood of log events. The license usage report emits a warning message when squashing occurs. Because of this squashing of the host and source fields, only the split by source type and index choices offer full reporting.

The squashing threshold is configurable. Increasing the value also increases memory usage. See the `squash_threshold` setting in server.conf.

To view more granular information without squashing, search metrics.log for `per_host_thruput`.

Identify metrics data in your license usage report

You can identify metrics data by selecting License Usage - Previous 30 Days, and split by index.

Set up an alert

You can turn any of the license usage report view panels into an alert. For example, if you want to set up an alert for when license usage reaches 80% of the quota:

1. Go to the **Today's percentage of daily license usage quota used** panel.
2. Click "Open in search" at the bottom left of a panel.
3. Append a new percentile value `| where '% used' > 80`
4. Select **Save as > Alert** and follow the alerting wizard.

Splunk Enterprise comes with several preconfigured alerts that you can enable. See Enable and configure platform alerts in *Monitoring Splunk Enterprise*.

Troubleshoot the license usage report view

No results in Previous 30 Days tab

If the panel is empty, the Splunk Enterprise instance acting as the license manager can not find any licensing events. These events are recorded in the `license_usage.log` file, and are ingested and stored in the `internal` index. Here are some scenarios that might cause the issue:

- The license manager instance is not configured to search the indexers or cluster peers. For instructions on configuring the license manager to search indexers or peer nodes, see Add search peers to the search head.
- The license manager instance stopped ingesting its local Splunk Enterprise log files. Use the `btool` command to check the default Splunk Enterprise log monitor `[monitor://$SPLUNK_HOME/var/log/splunk]` and verify it is enabled. For examples of `btool` use, see Use btool to troubleshoot configurations.

A gap might appear in the data if the license manager was unavailable at midnight, when license reconciliation occurs.

Single-source type license limitations

An instance that has both a single-source type license and an Enterprise license does not always show accurate information.

Administer the app key value store

About the app key value store

The app key value store (or KV store) provides a way to save and retrieve data within your Splunk apps, thereby letting you manage and maintain the state of the application.

Here are some ways that Splunk apps might use the KV Store:

- Tracking workflow in an incident-review system that moves an issue from one user to another.
- Keeping a list of environment assets provided by users.
- Controlling a job queue.
- Managing a UI session by storing the user or application state as the user interacts with the app.
- Storing user metadata.
- Caching results from search queries by Splunk or an external data store.
- Storing checkpoint data for modular inputs.

For information on using the KV store, see app key value store documentation for Splunk app developers.

How KV store works with your deployment

The KV store stores your data as key-value pairs in collections. Here are the main concepts:

- **Collections** are the containers for your data, similar to a database table. Collections exist within the context of a given app.
- **Records** contain each entry of your data, similar to a row in a database table.
- **Fields** correspond to key names, similar to the columns in a database table. Fields contain the values of your data as a JSON file. Although it is not required, you can enforce data types (number, boolean, time, and string) for field values.
- **_key** is a reserved field that contains the unique ID for each record. If you don't explicitly specify the _key value, the app auto-generates one.
- **_user** is a reserved field that contains the user ID for each record. This field cannot be overridden.
- **Accelerations** improve search performance by making searches that contain accelerated fields return faster. Accelerations store a small portion of the collection's data set in an easy-to-traverse form.

The KV store files reside on search heads.

In a search head cluster, if any node receives a write, the KV store delegates the write to the **KV store captain**. The KV store keeps the reads local, however.

System requirements

KV store is available and supported on all Splunk Enterprise 64-bit builds. It is not available on 32-bit Splunk Enterprise builds. KV store is also not available on universal forwarders. See the Splunk Enterprise system requirements.

KV store uses port 8191 by default. You can change the port number in server.conf's `[kvstore]` stanza. For information about other ports that Splunk Enterprise uses, see "System requirements and other deployment considerations for search head clusters" in the *Distributed Search Manual*.

For information about other configurations that you can change in KV store, see the "KV store configuration" section in server.conf.spec.

About Splunk FIPS

To use FIPS with KV store, see the "KV store configuration" section in server.conf.spec.

If Splunk FIPS is not enabled, those settings will be ignored.

If you enable FIPS but do not provide the required settings (`caCertFile`, `sslKeysPath`, and `sslKeysPassword`), KV store does not run. Look for error messages in splunkd.log and on the console that executes `splunk start`.

Determine whether your apps use KV store

KV store is enabled by default on Splunk Enterprise 6.2+.

Apps that use the KV store typically have collections.conf defined in `$SPLUNK_HOME/etc/apps/<app name>/default`. In addition, transforms.conf will have references to the collections with external_type = kvstore

Use the KV store

To use the KV store:

1. Create a collection and optionally define a list of fields with data types using configuration files or the REST API.
2. Perform create-read-update-delete (CRUD) operations using search lookup commands and the Splunk REST API.
3. Manage collections using the REST API.

Monitor the KV store on your Splunk Enterprise deployment

You can monitor your KV store performance through two views in the monitoring console. One view provides insight across your entire deployment. The other provides detailed information about KV store operations on each search head. See KV store dashboards in *Monitoring Splunk Enterprise*.

Disable the KV store

KV store is enabled by default. You can disable the KV store on indexers and forwarders, and on any installation that does not have any local apps or local lookups that use the KV store.

To disable the KV store, open the local server.conf file and edit the following stanza:

```
[kvstore]
disabled=true
```

You can disable the KV store on an instance while it is running if you don't have any additional collections.conf files beyond the following list of default files:

- $SPLUNK_HOME/etc/system/default/collections.conf
- $SPLUNK_HOME/etc/apps/splunk_secure_gateway/default/collections.conf
- $SPLUNK_HOME/etc/apps/splunk_instrumentation/default/collections.conf
- $SPLUNK_HOME/etc/apps/python_upgrade_readiness_app/default/collections.conf

- $SPLUNK_HOME/etc/apps/splunk-dashboard-studio/default/collections.conf
- $SPLUNK_HOME/etc/apps/splunk-rolling-upgrade/default/collections.conf

Resync the KV store

When a KV store member fails to transform its data with all of the write operations, then the KV store member might be stale. To resolve this issue, you must resynchronize the member.

Before downgrading Splunk Enterprise to version 7.1 or earlier, you must use the REST API to resynchronize the KV store.

Identify a stale KV store member

You can check the status of the KV store using the command line.

1. Log into the shell of any KV store member.
2. Navigate to the `bin` subdirectory in the Splunk Enterprise installation directory.
3. Type `./splunk show kvstore-status`. The command line returns a summary of the KV store member you are logged into, as well as information about every other member in the KV store cluster.
4. Look at the `replicationStatus` field and identify any members that have neither "KV store captain" nor "Non-captain KV store member" as values.

Resync stale KV store members

If more than half of the members are stale, you can either recreate the cluster or resync it from one of the members. See Back up KV store for details about restoring from backup.

To resync the cluster from one of the members, use the following procedure. This procedure triggers the recreation of the KV store cluster, when all of the members of current existing KV store cluster resynchronize all data from the current member (or from the member specified in -source sourceId). The command to resync the KV store cluster can be invoked only from the node that is operating as search head cluster captain.

1. Determine which node is currently the search head cluster captain. Use the CLI command `splunk show shcluster-status`.
2. Log into the shell on the search head cluster captain node.
3. Run the command `splunk resync kvstore [-source sourceId]`. The source is an optional parameter, if you want to use a member other than the search head cluster captain as the source. `SourceId` refers to the GUID of the search head member that you want to use.
4. Enter your admin login credentials.
5. Wait for a confirmation message on the command line.
6. Use the `splunk show kvstore-status` command to verify that the cluster is resynced.

If fewer than half of the members are stale, resync each member individually.

1. Stop the search head that has the stale KV store member.
2. Run the command `splunk clean kvstore --local`.
3. Restart the search head. This triggers the initial synchronization from other KV store members.
4. Run the command `splunk show kvstore-status` to verify synchronization.

Prevent stale members by increasing operations log size

If you find yourself resyncing KV store frequently because KV store members are transitioning to stale mode frequently (daily or maybe even hourly), this means that apps or users are writing a lot of data to the KV store and the operations log is too small. Increasing the size of the operations log (or oplog) might help.

After initial synchronization, noncaptain KV store members no longer access the captain collection. Instead, new entries in the KV store collection are inserted in the operations log. The members replicate the newly inserted data from there. When the operations log reaches its allocation (1 GB by default), it overwrites the beginning of the oplog. Consider a lookup that is close to the size of the allocation. The KV store rolls the data (and overwrites starting from the beginning of the oplog) only after the majority of the members have accessed it, for example, three out of five members in a KV store cluster. But once that happens, it rolls, so a minority member (one of the two remaining members in this example) cannot access the beginning of the oplog. Then that minority member becomes stale and need to be resynced, which means reading from the entire collection (which is likely much larger than the operations log).

To decide whether to increase the operations log size, visit the Monitoring Console **KV store: Instance** dashboard or use the command line as follows:

1. Determine which search head cluster member is currently the KV store captain by running `splunk show kvstore-status` from any cluster member.
2. On the KV store captain, run `splunk show kvstore-status`.
3. Compare the oplog start and end timestamps. The start is the oldest change, and the end is the newest one. If the difference is on the order of a minute, you should probably increase the operations log size.

While keeping your operations log too small has obvious negative effects (like members becoming stale), setting an oplog size much larger than your needs might not be ideal either. The KV store takes the full log size that you allocate right away, regardless of how much data is actually being written to the log. Reading the oplog can take a fair bit of RAM, too, although it is loosely bound. Work with Splunk Support to determine an appropriate operations log size for your KV store use. The operations log is 1 GB by default.

To increase the log size:

1. Determine which search head cluster member is currently the KV store captain by running `splunk show kvstore-status` from any cluster member.
2. On the KV store captain, edit server.conf file, located in `$SPLUNK_HOME/etc/system/local/`. Increase the `oplogSize` setting in the `[kvstore]` stanza. The default value is 1000 (in units of MB).
3. Restart the KV store captain.
4. For each of the other cluster members:
    1. Stop the member.
    2. Run `splunk clean kvstore --local`.
    3. Edit server.conf file, located in `$SPLUNK_HOME/etc/system/local/`. Increase the `oplogSize` setting in the `[kvstore]` stanza. The default value is 1000 (in units of MB).
    4. Restart the member.
    5. Run `splunk show kvstore-status` to verify synchronization.

Back up and restore KV store

Back up the KV store and restore it from backup. Taking regular backups from a healthy environment enables you to restore from a backup in the event of a disaster, or if you add a search head to a cluster. You can also take a backup before migrating to a different machine. See Migrate a Splunk Enterprise instance from one physical machine to another

in the *Installation Manual* for more information.

Make sure to be familiar with the standard backup and restore tools and procedures used by your organization.

You can perform different tasks with the KV store, including checking the status, taking a backup, and restoring the KV store to an existing or a new search head or search head cluster. Use the following table to decide which methods to use.

Task	Description	Limitations
Check the KV store status	Before taking a backup or restoring the KV store, you need to check that that KV store is ready. You can also check on backups and restores that are in progress.	n/a
Backup and restore with point in time consistency	Choose this method to guarantee consistency in the backup and restore process. This method captures all changes during the backup process, and blocks all changes during the restore process.	You must ensure that all searches, particularly real-time searches, are complete before restoring the KV store. You cannot backup specific apps or collection, only the entire KV store.
Backup and restore without guaranteed consistency	Choose this method to backup and restore specific apps or collections, or the entire KV store.	This method doesn't guarantee consistency in backup and restore. Changes made during backup aren't always captured.

Check the KV store status

To check the status of the KV store, use the `show kvstore-status` command:

```
./splunk show kvstore-status
```

The `backupRestoreStatus` field and the `status` field indicate the statuses of the KV store. The `backupRestoreStatus` field indicates the readiness of the node to perform a backup. The `status` field indicates the status of the storage engine. Both must be in a ready state for you to take a backup.

In a deployment that uses a search head cluster, use the `./splunk show shcluster-status --verbose` command at any time to see if any or all cluster members are in maintenance mode. The `kvstore_maintenance_status` field indicates the captain's status, and the `kvstore_status` field indicates the status for individual members.

Back up and restore the KV store with point in time consistency

Use the following steps to back up the KV store, prepare to restore the KV store data, and then restore the KV store data.

Back up the KV store

Complete the following steps to back up the KV store with point in time consistency.

1. In the CLI, run the `splunk show kvstore-status` command.
2. Ensure that the `backupRestoreStatus` field and the `status` field are both in the ready state.
3. If you are running any searches that use `outputlookup` with the default `append=f` parameter, end them or allow them to complete before taking a backup, or the backup fails.
4. (Optional) Create a separate partition for your backup directory, so that the backup is preserved if the $SPLUNK_DB/kvstore directory fails.
5. Use the `splunk backup kvstore -pointInTime true` command from any search head. This creates an archive file in the $SPLUNK_DB/kvstorebackup directory. You must use the `-pointInTime true` portion of the command to back up with consistency.

In a search head cluster deployment, only one backup operation can take place at a time. If you initiate a backup on more than one search head at the same time, only one backup succeeds.

To customize your backup, check the full list of arguments for the backup command:

```
./splunk backup kvstore [-pointInTime <true|false>] [-cancel <true|false>] [-parallelCollections <num>]
[-archiveName <archive>]
```

Argument	Description
-pointInTime	Defaults to `false`. To take a consistent backup, set it to `true`.
-cancel	Defaults to `false`. Set the argument to `true` to cancel an in-progress backup.
-parallelCollections	Defaults to `1`. Raise the number to increase the number of collections to back up in parallel.
-archiveName	Defaults to `kvdump_<epoch>.tar.gz`. Set to change the name of the backup file. Do not include the extension `.tar.gz`. It is appended automatically.

Prepare to restore the KV store data

Next, complete the following steps to prepare to restore the KV store data:

1. Check to see if a backup file was taken with consistency by using the `./splunk show kvstore -archiveName <archive file>` command. You can only restore the KV store with consistency with a backup file that was taken with consistency. Backups taken with consistency used the `-pointInTime true` argument in the backup command.
2. Make sure the KV store collection collections.conf file exists on the Splunk Enterprise instance in the same application name that the KV store is going to be restored to.

   If you create the collection collections.conf after restoring the KV store data, the KV store data will be lost.
3. Ensure that your backup archive file is in the $SPLUNK_DB/kvstorebackup directory. If your deployment uses a search head cluster, make sure that the archive file is in that directory on the captain node.
4. Check that you created the backup archive file from the same collection that you are restoring. You cannot restore a backup to a different collection.

Restore the KV store data to an existing deployment

Now complete the following steps to restore the KV store data.

Restoring KV store data overwrites any KV store data in your Splunk Enterprise instance with the data from the backup.

1. Ensure all searches are complete, especially real-time searches.
2. (Optional) To ensure that no searches that use the KV store are started by the scheduler, temporarily disable the scheduler.
3. If your deployment uses a search head cluster, switch to static captain mode.
4. Use the `splunk enable kvstore-maintenance-mode` command to enable maintenance mode. Once you enable maintenance mode, you cannot make any changes to the KV store, and searches that attempt to modify the KV store contents fail. Maintenance mode ensures that the restore completes with consistency.
5. From the search head cluster captain, restore the KV store data with the `splunk restore kvstore -pointInTime true -archiveName <archive>` command. Include the `.tar.gz` extension in the archive name. Even in a clustered deployment, only one restore operation can take place at a time.
6. Verify that the restore process is complete with the `splunk show kvstore-status` command.
7. Disable maintenance mode with the `splunk disable kvstore-maintenance-mode` command.

8. If you disabled the scheduler, enable it now.
9. If your deployment uses a search head cluster, switch back to dynamic captain mode.

To customize your restore, check the full list of arguments for the restore command:

```
./splunk restore kvstore [-pointInTime <true|false>] -archiveName <archive> [-parallelCollection <num>]
[-insertionsWorkersPerCollection <num>] [-cancel]
```

Argument	Description
-pointInTime	Defaults to `false`. To restore from a backup taken with consistency, set the argument to `true`.
-cancel	Defaults to `false`. Set the argument to `true` to cancel an in-progress restore.
-parallelCollections	Defaults to `1`. Raise the number to increase the number of collections to restore in parallel, which speeds up the store.
-archiveName	**Required**. Specify the name of the backup file to use. Include the `.tar.gz` extension in the archive name.
-insertionsWorkersPerCollection	Defaults to `1`. Raise the number to increase the number of insertion workers per collection, which speeds up the restore.

Back up and restore the KV store without guaranteed consistency

Use the following steps to back up the KV store, prepare to restore the KV store data, and then restore the KV store data, either to an existing deployment or to a new one.

Back up the KV store

Complete the following steps to back up the KV store:

1. In the CLI, run the `splunk show kvstore-status` command.
2. Ensure that the `backupRestoreStatus` field and the `status` field are both in the `ready` state before taking a backup.
3. (Optional) Create a separate partition for your backup directory, so that the backup is preserved if the $SPLUNK_DB/kvstore directory fails.
4. To create archive file in the $SPLUNK_DB/kvstorebackup directory, run the `splunk backup kvstore` command according to the following conditions:
    - If you're backing up a single search head deployment, run the command from the search head.
    - If you're backing up a search head cluster, run the command from the node with the most recent data.
5. (Optional) Add the following arguments to specify the name of the backup archive file, or define specific collections or apps to back up instead of the entire KV store:
   ```
   ./splunk backup kvstore [-archiveName <archive>] [-collectionName <collection>]  [-appName <app>]
   ```

Prepare to restore the KV store data

Complete the following steps to prepare to restore the KV store data.

1. Make sure the KV store collection collections.conf file exists on the Splunk Enterprise instance in the same application name that the KV store is going to be restored to.

   If you create the collection collections.conf after restoring the KV store data, the KV store data will be lost.
2. Ensure that your backup archive file is in the $SPLUNK_DB/kvstorebackup directory. In a search head cluster deployment, ensure the file is in this directory on the node from which you are restoring. You only need to restore from one node. The restore replicates across all of the other nodes automatically.
3. Check that you created the backup archive file from the same collection that you are restoring. You cannot restore a backup to a different collection.

Restore the KV store data to an existing deployment

Complete the following steps to restore the KV store data to an existing search head cluster:

> Restoring KV store data overwrites any KV store data in your Splunk Enterprise instance with the data from the backup.

1. Restore the KV store data with the `splunk restore kvstore` command.
2. (Optional) Add the following arguments to specify the name of the backup archive file, or specific collections or apps to restore instead of the entire KV store:
   `./splunk restore kvstore [-archiveName <archive>] [-collectionName <collection>]  [-appName <app>]`
3. Verify that the restore process is complete by running the `splunk show kvstore-status` command.

Restore the KV store data to a new search head cluster

Complete the following steps to create a new search head cluster with new Splunk Enterprise instances. This procedure only works if you took the backup from a search head cluster deployment without using the `-pointInTime true` argument.

1. Back up the KV store data from the same search head in the current search head cluster from which you took the backup.
2. On that search head that will be in the new search head cluster environment, create the KV store collection using the same collection name as the KV store data you are restoring.
3. Initialize the search head cluster with `replication_factor=1`
4. Restore the KV store data to the new search head by using the `splunk restore kvstore` command.
5. Run the following command from the CLI: `splunk clean kvstore --cluster`
6. Start the Splunk instance and bootstrap with the new search head.
7. After the KV store has been restored onto the new search head, add the other new search head cluster members.
8. After complete, change the `replication_factor` on each search head to the desired replication factor number.
9. Perform a rolling restart of your deployment.

Migrate the KV store storage engine

Splunk Enterprise versions 9.0 and higher require the WiredTiger storage engine and server version 4.2, which significantly reduces the amount of storage you need and improves performance. Migrate to WiredTiger either before or during upgrade to Splunk Enterprise 9.0, and then upgrade to server version 4.2. Migrating your storage engine before or during upgrade to Splunk Enterprise 9.0 or higher is a best practice, but migrating immediately after upgrade is required.

All Splunk Enterprise versions 8.1 and higher support WiredTiger, so you can consider migrating to WiredTiger before your upgrade to reduce downtime during the upgrade. If you prefer to perform the WiredTiger migration and the upgrade to Splunk Enterprise at separate times, check the documentation for your current version of Splunk Enterprise to complete your migration before initiating your upgrade to Splunk Enterprise 9.0 or higher.

To migrate your KV store storage engine during your upgrade to Splunk Enterprise 9.0 or higher, first determine your deployment type. If your single instance of the KV store is located on a search head, the cluster manager, or any indexer node, you have a single-instance KV store deployment. If you have multiple KV store nodes across a search head cluster, then you have a clustered KV store deployment. Complete the steps associated with your deployment type.

After completing your migration to WiredTiger and server version 4.2, you can optionally remove the unsupported binary files for previous versions of MongoDB.

Migrate the KV store in a single-instance deployment

Single-instance deployments of Splunk Enterprise 9.0 and higher are automatically migrated to the WiredTiger storage engine and the latest version of MongoDB, server version 4.2, during the upgrade.

1. Complete any prompts during your Splunk Enterprise upgrade.
2. Verify that you have the latest version of the storage engine after upgrade with the following command:

   ```
   splunk show kvstore-status --verbose
   ```
3. Verify that the `serverVersion` and `storageEngine` fields indicate the latest versions. See the following example:

   ```
   serverVersion : 4.2.17
   [...]
   storageEngine : wiredTiger
   ```

Set your library path

If your instance fails to automatically migrate to WiredTiger and the latest version of Mongo DB during upgrade, you might need to set or correct your library path. If you receive the following error message, consider setting your library path:

```
/opt/splunk/bin/mongodump: error while loading shared libraries: libssl.so.1.0.0: cannot open shared object file: No such file or directory
```

To learn how to set your library path, see About using SSL tools on Windows and Linux in the *Securing Splunk Enterprise* manual.

Migrate manually to the latest version of Mongo DB

If your instance failed at migrating to WiredTiger, complete the steps for a manual migration in Migrate the KV store after an upgrade to Splunk Enterprise 8.1.* or 8.2.* in a single-instance deployment.

If you have successfully migrated to WiredTiger, but not to the latest version of Mongo DB, manually upgrade to the latest version of Mongo DB with the following CLI command:

```
splunk migrate migrate-kvstore
```

Migrate the KV store in a clustered deployment

You must manually migrate your KV store storage engine during your upgrade to Splunk Enterprise 9.0 and higher if you have not done so prior to beginning the upgrade. First prepare your deployment, then migrate to WiredTiger. After you verify that the migration is complete, then upgrade to MongoDB version 4.2.

Prepare your deployment to migrate your storage engine

Avoid any of the following actions right before or during migration:

- If you are running any searches on a KV store node when you begin migrating, that search might fail. Searches that start running after you begin migration are not impacted.
- Do not do any heavy writes to the KV store while the migration is in progress.
- Do not add new search heads while the migration is in progress.

Complete the following steps to prepare your deployment before you migrate your storage engine:

1. Plan sufficient time for your upgrade and migration. The time it takes to migrate the KV store storage engine is proportional to the total data in your KV store.
2. (Optional) Back up your KV store data before you begin the migration process. The KV store non-captain nodes are synced from the captain on a rolling basis, one node at a time, and the migration process does not automatically back up KV store data to a separate location.
3. Upgrade to Splunk Enterprise 9.0 or higher. For more information, see How to upgrade Splunk Enterprise in the *Installation Manual*.
4. After your upgrade completes, Splunk Enterprise prompts you to upgrade your storage engine immediately to WiredTiger.

Use the `curl -k -u admin:changeme -X POST https://localhost:8089/services/shcluster/captain/kvmigrate/stop` command to stop the migration process at any time.

Initiate your KV store storage engine migration

After you prepare your deployment, initiate your migration.

1. Check that your instance is ready to migrate by using one of the following commands. You can perform this check with either the REST API or with the Splunk Enterprise command-line interface (CLI).
   REST API:
   ```
   curl -k -u admin:changeme https://localhost:8089/services/shcluster/captain/kvmigrate/start -d
   storageEngine=wiredTiger -d isDryRun=true
   ```
   CLI:
   ```
   splunk start-shcluster-migration kvstore -storageEngine wiredTiger -isDryRun true
   ```
2. Resolve any issues blocking migration. Perform the migration only if all checks pass.
3. Initiate the migration from any node with the following command. To select the options for your command, choose if you want to migrate based on a percentage of nodes or based on specific URIs. If you want to migrate specific peers, specify their names and the management port number. If you specify neither option, then all nodes are migrated on a rolling basis one at a time. Initiate the migration only once from any one node. All nodes are automatically migrated after that.

Option	REST API sample	
By percentage	`curl -k -u admin:changeme https://localhost:8089/services/shcluster/captain/kvmigrate/start -X POST -d storageEngine=wiredTiger -d clusterPerc=50`	`splunk start-shcluste -storageEngine wired -clusterPerc 50`
By URIs	`curl -k -u admin:changeme https://localhost:8089/services/shcluster/captain/kvmigrate/start -X POST -d storageEngine=wiredTiger -d peersList="https://server1:8089,https://server2:8089,https://server3:8089"`	`splunk start-shcluste -storageEngine wired -peersList "https://server1:808`

Monitor and verify your KV store storage engine migration

After your migration is in progress, you can use any of several methods to monitor your migration and verify that it is complete.

- To check which nodes are currently migrating, use the following commands. You can perform this check with either the REST API or with the Splunk Enterprise command-line interface (CLI).
  REST API:
  ```
  curl -k -u admin:changeme https://localhost:8089/services/shcluster/captain/kvmigrate/status
  ```
  CLI:
  ```
  splunk show shcluster-kvmigration-status
  ```

- For more information about the status of the upgrade, use the following command:
  `splunk show kvstore-status`
- To check the progress of the migration of a cluster member, see the `KVStoreReplicaSetStats` entry in the `$SPLUNK_HOME/var/log/introspection/kvstore.log` file on *nix, or the `%SPLUNK_HOME\var\log\introspection\kvstore.log` file on Windows, on that member. This status updates every 30 seconds.

If you backed up your KV store, verify that the migration is successful and then delete the KV store backup data.

Upgrade KV store server to version 4.2

If you have a single-instance deployment, your server version updates to MongoDB version 4.2 automatically. If you have a clustered deployment, however, choose a maintenance window in which to upgrade to MongoDB version 4.2, and then complete the following steps:

1. Check that your instance is ready to migrate with one of the following commands:
   CLI:
   ```
   splunk start-shcluster-upgrade kvstore -version 4.2 -isDryRun true
   ```
   REST:
   ```
   curl -ku admin:changeme -X POST
   https://localhost:8089/services/shcluster/captain/kvstore-upgrade/start -d version=4.2 -d
   isDryRun=true
   ```
2. Resolve any issues blocking migration, and then perform the migration only if all checks pass. Initiate the migration only once from any one node. All nodes are automatically migrated after that.
3. Use one of the following commands to initiate this upgrade:
   CLI:
   ```
   splunk start-shcluster-upgrade kvstore -version 4.2
   ```
   REST:
   ```
   curl -ku admin:changeme -X POST
   https://localhost:8089/services/shcluster/captain/kvstore-upgrade/start -d version=4.2
   ```
4. Verify that you have the latest version of the storage engine after upgrade with one of the following commands:
   CLI:
   ```
   splunk show kvstore-status --verbose
   ```
   REST:
   ```
   curl -k -u admin:changeme https://localhost:8089/services/shcluster/captain/kvmigrate/status
   ```
5. Check the output to see that the `serverVersion` and `storageEngine` fields indicate the latest versions:
   ```
   serverVersion : 4.2.17
   [...]
   storageEngine : wiredTiger
   ```
   In Unix operating systems, the latest server version is 4.2.17. In Windows operating systems, the latest server version is 4.2.19.

Remove unsupported binary files for lower server versions

After you complete your migration to the WiredTiger storage engine and server version 4.2, you can choose to remove the unsupported binary files for MongoDB version 3.6. Removing these files is optional. Complete the following steps to remove the files:

1. Verify that you have the latest version of the storage engine with the following command:

   ```
   splunk show kvstore-status --verbose
   ```
2. Verify that the serverVersion and storageEngine fields indicate the latest versions:

   ```
   serverVersion : 4.2.17
   [...]
   storageEngine : wiredTiger
   ```

   In Unix operating systems, the latest server version is 4.2.17. In Windows operating systems, the latest server version is 4.2.22.
3. Delete the following files from the $SPLUNK_HOME/bin directory:
   - mongod-3.6
   - mongod-4.0
   - mongodump-3.6
   - mongorestore-3.6

After you remove these files and restart your instance, you can ignore the following message. You don't need to take any action.

```
03-25-2022 12:34:18.203 -0700 WARN  InstalledFilesHashChecker [3769773 LazyGlobalManifestCheck] - An
installed file="/opt/splunk/bin/mongod-3.6" did not pass hash-checking due to reason="file missing"
```

KV store troubleshooting tools

This topic discusses tools for viewing KV store status and its log files. It also discusses some monitoring tools that you can use in Splunk Enterprise.

Check KV store status

You can check the status of the KV store in the following ways:

- Use the command line.
- Make a REST API GET request.
- Run the KV store health check in the monitoring console. See Access and customize health check in *Monitoring Splunk Enterprise*.

KV store status CLI command

On the command line from any KV store member, in `$SPLUNK_HOME/bin` type the following command:

```
./splunk show kvstore-status
```

See About the CLI for information about using the CLI in Splunk software.

KV store status REST endpoint

Use cURL to make a GET request with the REST API:

```
curl -k -u user:pass https://<host>:<mPort>/services/kvstore/status
```

See Basic Concepts in the *REST API User Manual* for more information about the REST API.

KV store status definitions

The following is a list of possible values for `status` and `replicationStatus` and their definitions. For more information about abnormal statuses for your KV store members, check mongod.log and splunkd.log for errors and warnings.

KV store status	Definition
starting	• In the case of a standalone search head, this status switches to `ready` after synchronization of a list of defined collections, accelerated fields, and so on. • In the case of a search head cluster, this status switches to `ready` when the search head cluster is bootstrapped (after the search head cluster captain is elected) and the search head cluster captain propagates status to all search head cluster members.
disabled	KV store is disabled in server.conf on this instance. If this member is a search head cluster member, its status remains disabled only if all other members of the search head cluster have KV store disabled.
ready	KV store is ready for use.
failed	Failed to bootstrap and join the search head cluster.
shuttingdown	Splunk software has notified KV store about the shutting down procedure.

KV store replication status	Definition
Startup	Member is starting.
KV store captain	Member is the elected KV store captain.
Non-captain KV store member	Healthy noncaptain member of KV store cluster.
Initial sync	This member is resynchronizing data from one of the other KV store cluster members. If this happens often, or if this member remains in this state, check mongod.log and splunkd.log on this member, and verify connection to this member and connection speed.
Down	Member is stopped.
Removed	Member is removed from the KV store cluster, or is in the process of being removed.
Rollback / Recovering / Unknown status	Member might have a problem. Check mongod.log and splunkd.log on this member.

Sample command-line response:

```
This member:
                        date : Tue Jul 21 16:42:24 2016
                    dateSec : 1466541744.143000
                    disabled : 0
                        guid : 6244DF36-D883-4D59-AHD3-5276FCB4BL91
          oplogEndTimestamp : Tue Jul 21 16:41:12 2016
```

```
                    oplogEndTimestampSec : 1466541672.000000
                     oplogStartTimestamp : Tue Jul 21 16:34:55 2016
                   oplogStartTimestampSec : 1466541295.000000
                                     port : 8191
                               replicaSet : splunkrs
                         replicationStatus : KV store captain
                                standalone : 0
                                    status : ready

Enabled KV store members:
       10.140.137.128:8191
                                     guid : 6244DF36-D883-4D59-AHD3-5276FCB4BL91
                               hostAndPort : 10.140.137.128:8191
       10.140.137.119:8191
                                     guid : 8756FA39-F207-4870-BC5D-C57BABE0ED18
                               hostAndPort : 10.140.137.119:8191
       10.140.136.112:8191
                                     guid : D6190F30-C59A-423Q-AB48-80B0012317V5
                               hostAndPort : 10.140.136.112:8191

KV store members:
       10.140.137.128:8191
                               configVersion : 1
                                electionDate : Tue Jul 21 16:42:02 2016
                             electionDateSec : 1466541722.000000
                                hostAndPort : 10.140.134.161:8191
                                 optimeDate : Tue Jul 21 16:41:12 2016
                               optimeDateSec : 1466541672.000000
                             replicationStatus : KV store captain
                                      uptime : 108
       10.140.137.119:8191
                               configVersion : 1
                                 hostAndPort : 10.140.134.159:8191
                               lastHeartbeat : Tue Jul 21 16:42:22 2016
                            lastHeartbeatRecv : Tue Jul 21 16:42:22 2016
                          lastHeartbeatRecvSec : 1466541742.490000
                              lastHeartbeatSec : 1466541742.937000
                                    optimeDate : Tue Jul 21 16:41:12 2016
                               optimeDateSec : 1466541672.000000
                                      pingMs : 0
                             replicationStatus : Non-captain KV store member
                                      uptime : 107
       10.140.136.112:8191
                               configVersion : -1
                                 hostAndPort : 10.140.133.82:8191
                               lastHeartbeat : Tue Jul 21 16:42:22 2016
                            lastHeartbeatRecv : Tue Jul 21 16:42:00 2016
                          lastHeartbeatRecvSec : 1466541720.503000
                              lastHeartbeatSec : 1466541742.959000
                                    optimeDate : ZERO_TIME
                               optimeDateSec : 0.000000
                                      pingMs : 0
                             replicationStatus : Down
                                      uptime : 0
```

KV store messages

The KV store logs error and warning messages in internal logs, including splunkd.log and mongod.log. These error messages post to the bulletin board in Splunk Web. See What Splunk software logs about itself for an overview of internal log files.

Recent KV store error messages also appear in the REST `/services/messages` endpoint. You can use cURL to make a GET request for the endpoint, as follows:

```
curl -k -u user:pass https://<host>:<mPort>/services/messages
```

For more information about introspection endpoints, see System endpoint descriptions in the *REST API Reference Manual.*

Updating the IP address of a KV store server can require a resync

If you update the IP address of a KV store server, you might receive the following error in mongod.log:

```
Did not find local replica set configuration document at startup; NoMatchingDocument
Did not find replica set configuration document in local.system.replset
```

To reconfigure the cluster to pick up the new IP address, resync to force the cluster configuration to refresh:

```
splunk resync shcluster-replicated-config
```

A manual resync with this command overwrites any local changes on that KV store server. For more information about manually resyncing a cluster member, see Why a recovering member might need to resync manually in the *Distributed Search* manual.

For more information about resyncing the KV store, see Resync the KV store.

Monitor KV store performance

You can monitor your KV store performance through two views in the monitoring console. The KV store: Deployment dashboard provides information aggregated across all KV stores in your Splunk Enterprise deployment. The KV store: Instance dashboard shows performance information about a single Splunk Enterprise instance running the KV store. See KV store dashboards in *Monitoring Splunk Enterprise*.

Meet Splunk apps

Apps and add-ons

Apps and add-ons allow you to extend the functionality of the Splunk platform.

App

An **app** is an application that runs on the Splunk platform. Apps are designed to analyze and display knowledge around a specific data source or data set.

An app might include any or all of the following configurations:

- Dashboards and supporting searches that integrate knowledge of the data source and structure.
- Authentication management and other data source management interfaces.
- An app might require the use of one or more add-ons to facilitate the collection or configuration of data.

Some apps are free and a few are paid. Examples of free apps include: Splunk App for Microsoft Exchange, Splunk App for AWS, and Splunk DB Connect.

> Store your apps on a fast, local disk, not on network file system (NFS). Loading apps on NFS can become a performance bottleneck.

Add-on

An **add-on** provide specific capabilities to assist in gathering, normalizing, and enriching data sources.

An add-on might include any or all of the following configurations:

- Data source input configurations.
- Data parsing and transformation configurations to structure the data for Splunk Enterprise.
- Lookup files for data enrichment.
- Supporting knowledge objects.

Examples include: Splunk Add-on for Checkpoint OPSEC LEA, Splunk Add-on for Box, and Splunk Add-on for McAfee.

App and add-on support

Anyone can develop an app or add-on for Splunk software. Splunk and members of our community create apps and add-ons and share them with other users of Splunk software on the online app marketplace Splunkbase. Splunk *does not* support all apps and add-ons on Splunkbase.

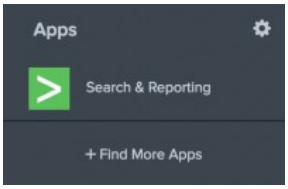
- For a list of the support options for apps and add-ons, see Support types for apps on Splunkbase on the Splunk Developer Portal.
- For guidance on developing apps, see Developer Guide for Splunk Cloud Platform and Splunk Enterprise on the Splunk Developer Portal.

Search and Reporting app

By default, Splunk Enterprise provides the Search and Reporting app. This interface provides the core functionality of Splunk Enterprise. The Splunk Home page provides a link to the app when you first log into Splunk Web.

Find Splunk Search and Reporting

1. If you are not on the Splunk Home page, click the **Splunk** logo on the Splunk bar to go to Splunk Home.
2. From Splunk Home, click **Search & Reporting** in the **Apps** panel.



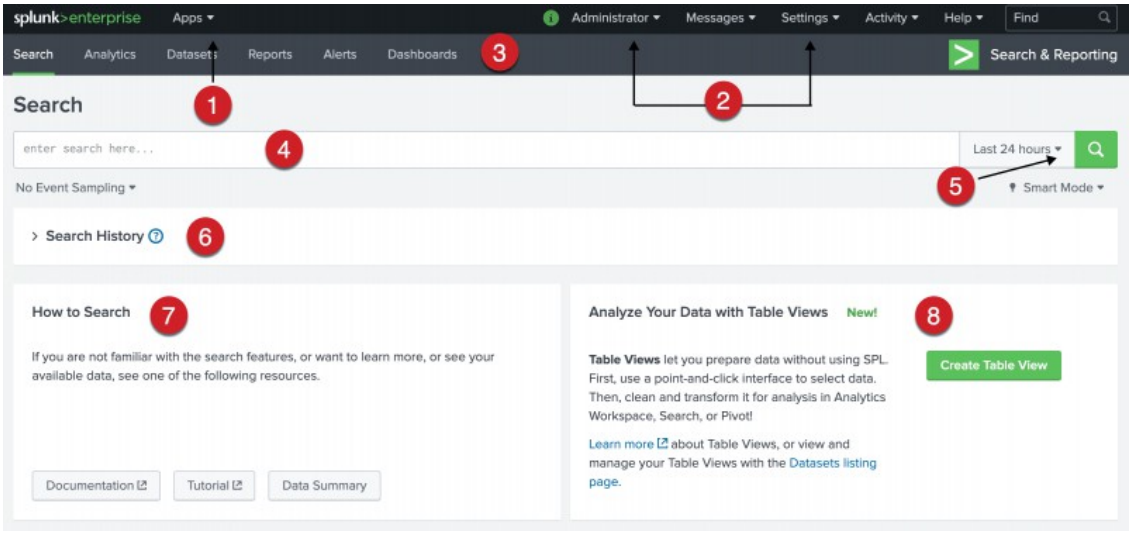
This opens the Search Summary view in the Search app.

Search Summary view

The Search Summary view includes common elements that you see on other views, including the Applications menu, the Splunk bar, the Apps bar, the Search bar, and the Time Range Picker. Elements that are unique to the Search Summary view are the panels below the Search bar: the **How to Search** panel, the **What to Search** panel, and the **Search History** panel.

Before you run a search, the Search summary view displays the following elements: App bar, Search bar, Time range picker, **How to Search** panel, **Search History** panel, and the **Analyze Your Data with Table Views** panel.

Some of these are common elements that you see on other views. Elements that are unique to the Search Summary view are the panels below the Search bar.



Number	Element	Description
1	**Applications menu**	Switch between Splunk applications that you have installed. The current application, Search & Reporting app, is listed. This menu is on the Splunk bar.
2	**Splunk bar**	Edit your Splunk configuration, view system-level messages, and get help on using the product.
3	**Apps bar**	Navigate between the different views in the application you are in. For the Search & Reporting app the views are: Search, Datasets, Reports, Alerts, and Dashboards.
4	**Search bar**	Specify your search criteria.
5	**Time range picker**	Specify the time period for the search, such as the last 30 minutes or yesterday. The default is **Last 24 hours**.
6	**Search history**	View a list of the searches that you have run. The search history appears after you run your first search.
7	**How to Search**	Use the links to learn more about how to start searching your data, as well a summary of the data that you have access to.
8	**Analyze Your Data with Table Views**	Create curated collections of event data into datasets that you design for a specific business purpose.

Configure Splunk Web to open directly to an app

You can configure Splunk Web so that it bypasses Splunk Home and instead opens in a specific app of your choosing. This is called setting a default app. While we recommend implementing a default app using roles, you can also set a default app for users or on a per-user basis. A default app set for a user will take precedence over the default app for that user's role.

Set a default app by role

You can set a default app for all users with a specific role. For example, you could send all users with the "user" role to an app you created, and direct all admin users to the Monitoring Console on login.

To bypass Splunk Home for all users of a specific role:

1. In Splunk Web, click **Settings > Roles**.
2. Click the name of the role you wish to configure.
3. Select the **5. Resources** tab.
4. Use the **Default app** dropdown to select the new default app for the role.
5. Click **Save**.

The change goes into effect without a restart.

Set a default app for all users

You can specify a default app for all users to land in when they log in. For example, to set the Search app as the global default:

1. Create or edit `$SPLUNK_HOME/etc/apps/user-prefs/local/user-prefs.conf` (*nix) or `%SPLUNK_HOME%\etc\apps\user-prefs\local\user-prefs.conf` (Windows).

2. Specify
   [general]
   default_namespace = search
3. Restart Splunk Enterprise for the change to take effect.

See user-prefs.conf.spec.

Set a default app for a single user

In most cases, you should set default apps by role. But if your use case requires you to set a default app for a specific user, you can do this through Splunk Web.

To make the Search app the default landing app for a user:

1. In Splunk Web, click **Settings > Users**.
2. Click the name of the user you wish to configure.
3. Use the **Default app** dropdown to select the new default app for the user.
4. Click **Save**.

The change goes into effect without a restart.

About app permissions

A user will see an error if:

- The user does not have permission to access their default app, or
- The default app does not exist (for example, if it is typed incorrectly in user-prefs.conf).

See Manage app and add-on configurations and properties for information about managing permissions on an app.

Where to get more apps and add-ons

You can find new apps and add-ons on **Splunkbase**. You can also browse new apps from the Splunk Enterprise home page.

How you obtain new apps and add-ons from your Splunk Enterprise instance depends on whether or not your instance has a connection to the Internet.

Get Splunk apps and add-ons when there is an Internet connection

If your Splunk Enterprise server or your client machine has a connection to the Internet, you can navigate to the app browser from the home page.

- You can click the **+** sign below your last installed app to go directly to the app browser.
- You can also click the gear next to **Apps** to go to the apps manager page. Click **Browse more apps** to go to the app browser.

Considerations for updating apps using instances that you have secured or that use proxied Internet connections

If Splunk Web is located behind a proxy server, you might have trouble accessing Splunkbase. To address this problem, set the `HTTP_PROXY` environment variable on the machine that runs Splunk Enterprise, as described in Use Splunk Web with a reverse proxy configuration.

If you secure your installation with Secure Sockets Layer and your own certificates, and especially if you configure the instance to explicitly verify those certificates for each connection, you might need to either perform additional configuration to ensure that your instance can access Splunkbase through Splunk Web or use the CLI to update the apps. See About securing Splunk Enterprise with SSL for information on the settings you need to change to ensure Splunk Web connects to Splunkbase when you have enabled certificates and explicit certificate checking..

Get Splunk apps and add-ons where there is no Internet connection

If your Splunk Enterprise instance and client do not have Internet connectivity, you must download apps from Splunkbase on a machine that does, and subsequently copy them over to the instance:

1. From a computer that has an internet connection, browse the Splunkbase website for the app or add-on you want.
2. Download the app or add-on.
3. After you download the app or add-on, use the file management tools on your machine to copy it to your Splunk Enterprise instance.
4. On the Splunk Enterprise instance, put the app or add on in the `$SPLUNK_HOME/etc/apps` directory.
5. Unpack the app or add-on, using a command-line or GUI tool like `tar -xvf` (on *nix) or WinZip on Windows.

   Splunk apps and add-ons are packaged with a .SPL extension, but the file format is a tarred and gzipped archive. You might need to configure your tool to recognize this extension.
6. Depending on the app or add-on contents, you might need to restart Splunk Enterprise.
7. Your app or add-on is now installed and will be available from Splunk Home if it has a Splunk Web component.

App deployment overview

This topic provides an overview of the methods you can use to deploy Splunk apps and add-ons in common Splunk software environments.

For more detailed app and add-on deployment information, see your specific Splunk app documentation, or see Where to install Splunk add-ons in the *Splunk Add-ons* manual.

Prerequisites

You must have an existing Splunk platform deployment on which to install Splunk apps and add-ons.

Deployment methods

There are several ways to deploy apps and add-ons to the Splunk platform. The correct deployment method to use depends on the following characteristics of your specific Splunk software deployment:

- Deployment architecture (single-instance or distributed)
- Cluster types (search head clusters and/or indexer clusters)
- Location (customer-managed Splunk Enterprise or hosted Splunk Cloud Platform)

Guided Data Onboarding

Guided Data Onboarding (GDO) provides end-to-end guidance for getting specific data sources into specific Splunk platform deployments. You must have a Splunk deployment up and running and if you have an admin or equivalent role so that you can install add-ons.

From your home page in Splunk Web, find the data onboarding guides by clicking **Add Data**. You can either search for a data source or explore different categories of data sources. After you select your data source, you select a deployment scenario. From there you can view diagrams and high-level steps to set up and to configure your data source.

Splunk Web links to documentation that explains how to set up and configure your data source in greater detail. You can find all the Guided Data Onboarding manuals by clicking the **Add data** tab on the Splunk Enterprise Documentation site.

Deployment architectures

There are two basic Splunk Enterprise deployment architectures:

- **Single-instance deployment:** In a single-instance deployment, one Splunk Enterprise instance acts as both search head and indexer.
- **Distributed deployment:** A distributed deployment can include multiple Splunk Enterprise **components**, including search heads, indexers, and forwarders. See Scale your deployment with Splunk Enterprise components in the *Distributed Deployment Manual*. A distributed deployment can also include standard individual components and/or clustered components, including search head clusters, indexer clusters, and multi-site clusters. See Distributed Splunk Enterprise overview in the *Distributed Deployment Manual*.

Single-instance deployment

To deploy an app on a single instance, download the app from **Splunkbase** to your local host, then install the app using **Splunk Web**.

Some apps currently do not support installation through Splunk Web. Make sure to check the installation instructions for your specific app prior to installation.

Distributed deployment

You can deploy apps in a distributed environment using the following methods:

- Install apps manually on each component using Splunk Web, or install apps manually from the command line.
- Install apps using the **deployment server**. The deployment server automatically distributes new apps, app updates, and certain configuration updates to search heads, indexers, and forwarders. See About deployment server and forwarder management in *Updating Splunk Enterprise Instances*.

Alternately, you can deploy apps using a third-party configuration management tool, such as:

- Chef
- Puppet
- Salt
- Windows configuration tools

For the most part, you must install Splunk apps on search heads, indexers, and forwarders. To determine the Splunk

Enterprise components on which you must install the app, see the installation instructions for the specific app.

Deploy apps to clusters

Splunk distributed deployments can include these cluster types:

- **Search head clusters**
- **Indexer clusters**

You deploy apps to both indexer and search head cluster members using the **configuration bundle** method.

Search head clusters

To deploy apps to a search head cluster, you must use the **deployer**. The deployer is a Splunk Enterprise instance that distributes apps and configuration updates to search head cluster members. The deployer cannot be a search head cluster member and must exist outside the search head cluster. See Use the deployer to distribute apps and configuration updates in the *Distributed Search* manual.

Caution: Do not deploy a configuration bundle to a search head cluster from any instance other then the deployer. If you run the `apply shcluster-bundle` command on a non-deployer instance, such as a cluster member, the command deletes all existing apps and user-generated content on all search head cluster members!

Indexer clusters

To deploy apps to peer nodes (indexers) in an indexer cluster, you must first place the apps in the proper location on the indexer cluster manager node, then use the configuration bundle method to distribute the apps to peer nodes. You can apply the configuration bundle to peer nodes using Splunk Web or the CLI. For more information, see Update common peer configurations and apps in *Managing Indexers and Clusters of Indexers*.

While you cannot use the deployment server to deploy apps to peer nodes, you can use it to distribute apps to the indexer cluster manager node. For more information, see Use deployment server to distribute apps to the manager node in *Managing Indexers and Clusters of Indexers*.

Deploy apps to Splunk Cloud Platform

If you want to deploy an app or add-on to Splunk Cloud Platform, see Install apps in your Splunk Cloud Platform deployment.

App architecture and object ownership

Apps are commonly built from Splunk **knowledge objects**. Splunk knowledge objects include saved searches, event types, tags -- data types that enrich your Splunk deployment and make it easier to find what you need.

> You might save objects to add-ons as well, though this is not common. Apps and add-ons are both stored in the apps directory. In the rare case that you save objects to an add-on, you manage the add-on as described for apps in this topic.

Any user logged into Splunk Web can create and save knowledge objects to the user's directory under the app the user is "in" (assuming that they have sufficient permissions). This is the default behavior. When a user saves an object, it goes

into the user's directory in the local directory of the currently running app: `$SPLUNK_HOME/etc/users/<user_name>/<app_name>/local`. When the user has saved the object in that app, it is available only to that user when they are in that app unless they do one of the following:

- Promote the object so that it is available to all users who have access.
- Restrict the object to specific roles or users (still within the app context).
- Mark the object as globally available to all apps, add-ons, and users (unless they have explicitly restricted it by role/user).

Users must have write permissions for an app or add-on before they can promote objects to that level.

Promote and share Splunk knowledge

Users can share their Splunk knowledge objects with other users through the Permissions dialog. This means users who have read permissions in an app or add-on can see the shared objects and use them. For example, if a user shares a saved search, other users can see that saved search, but only within the app in which the search was created. So if you create a saved search in the app "Fflanda" and share it, other users of Fflanda can see your saved search if they have read permission for Fflanda.

Users with write permission can promote their objects to the app level. This means the objects are copied from their user directory to the app's directory -- from:

`$SPLUNK_HOME/etc/users/<user_name>/<app_name>/local/`

to:

`$SPLUNK_HOME/etc/apps/<app_name>/local/`

Users can do this only if they have write permission in the app. For a discussion of app object permissions, and governing access to those objects, see Set app permissions using Splunk Web on the Splunk Developer Portal.

Make Splunk knowledge objects globally available

Finally, after promotion, users can decide if they want their object to be available globally, meaning that all apps are able to see it. The user must have permission to write to the original app. It's easiest to do this in Splunk Web, but a user can also do it later by moving the relevant object into the desired directory.

To make globally available an object "A" (defined in "B.conf") that belongs to user "C" in app "D":

1. Move the stanza defining the object A from `$SPLUNK_HOME/etc/users/C/D/B.conf` into `$SPLUNK_HOME/etc/apps/D/local/B.conf`.

2. Add a setting, `export = system`, to the object A's stanza in the app's `local.meta` file. If the stanza for that object doesn't already exist, you can just add one.

For example, to promote an event type called "rhallen" created by a user named "fflanda" in the *Nix app so that it is globally available:

1. Move the [rhallen] stanza from `$SPLUNK_HOME/etc/users/fflanda/unix/local/eventtypes.conf` to `$SPLUNK_HOME/etc/apps/unix/local/eventtypes.conf`.

2. Add the following stanza:

```
[eventtypes/rhallen]
export = system
```

to `$SPLUNK_HOME/etc/apps/unix/metadata/local.meta`.

Adding the export = system setting to local.meta isn't necessary when you share event types from the Search app, because it exports all of its events globally by default.

What objects does this apply to?

The knowledge objects discussed here are limited to those that are subject to access control. These objects are also known as app-level objects and users can view them by selecting **Apps > Manage Apps** from the Splunk bar. This page is available to all users to manage any objects they have created and shared. These objects include:

- Saved searches and Reports
- Event types
- Views and dashboards
- Field extractions

There are also system-level objects available only to users with admin privileges (or read/write permissions on the specific objects). These objects include:

- Users
- Roles
- Auth
- Distributed search
- Inputs
- Outputs
- Deployment
- License
- Server settings (for example: host name, port, etc)

If you add an input, Splunk adds that input to the copy of inputs.conf that belongs to the app you're currently in. This means that if you navigated to your app directly from Search, your input will be added to $SPLUNK_HOME/etc/apps/search/local/inputs.conf, which might not be the behavior you desire.

App configuration and knowledge precedence

When you add knowledge to Splunk, it's added in the context of the app you're in when you add it. When Splunk is evaluating configurations and knowledge, it evaluates them in a specific order of precedence, so that you can control what knowledge definitions and configurations are used in what context. Refer to About configuration files for more information about Splunk configuration files and the order of precedence.

Manage app and add-on objects

When an **app** or **add-on** is created by a Splunk user, a collection of objects is created that make up the app or add-on. These objects can include **views**, commands, navigation items, **event types**, **saved searches**, **reports**, and more. Each of these objects have permissions associated with them to determine who can view or alter them. By default, the admin user has **permissions** to alter all the objects in the Splunk system.

Refer to these topics for more information:

- For an overview of apps and add-ons, refer to What are apps and add-ons? in this manual.
- For more information about app and add-on permissions, refer to App architecture and object ownership in this manual.
- To learn more about how to create your own apps and add-ons, refer to *Developing Views and Apps for Splunk Web*.

View app or add-on objects in Splunk Web

You can use Splunk Web to view the objects in your Splunk platform deployment in the following ways:

- To see all the objects for all the apps and add-ons on your system at once: **Settings > All configurations**.
- To see all the saved searches and report objects: **Settings > Searches and reports**.
- To see all the event types: **Settings > Event types**.
- To see all the field extractions: **Settings > Fields**.

You can:

- View and manipulate the objects on any page with the sorting arrows
- Filter the view to see only the objects from a given app or add-on, owned by a particular user, or those that contain a certain string, with the App context bar.

Use the Search field on the App context bar to search for strings in fields. By default, the Splunk platform searches for the string in all available fields. To search within a particular field, specify that field. Wildcards are supported.

Note: For information about the individual search commands on the Search command page, refer to the *Search Reference Manual*.

Manage apps and add-ons in clustered environments

Manage apps and their configurations in clustered environments by changing the **configuration bundle** on the manager node for indexer clusters and the deployer for search head clusters. Access the relevant clustering documentation for details:

- Update common peer configurations and apps in *Managing Indexers and Clusters of Indexers*.
- Use the deployer to distribute apps and configuration updates in *Distributed Search*.

Manage apps and add-ons on standalone instances

Update an app or add-on in the CLI

To update an existing app on a standalone Splunk instance using the CLI:

```
./splunk install app <app_package_filename> -update 1 -auth <username>:<password>
```

Splunk updates the app or add-on based on the information found in the installation package.

Disable an app or add-on using the CLI

To disable an app on a standalone Splunk instance via the CLI:

```
./splunk disable app [app_name] -auth <username>:<password>
```

Note: If you are running Splunk Free, you do not have to provide a username and password.

Uninstall an app or add-on

To remove an installed app from a standalone Splunk platform installation:

1. (Optional) Remove the app or add-on's indexed data. Typically, the Splunk platform does not access indexed data from a deleted app or add-on. However, you can use the Splunk CLI clean command to remove indexed data from an app before deleting the app. See Remove data from indexes with the CLI command.
2. Delete the app and its directory. The app and its directory are typically located in `$SPLUNK_HOME/etc/apps/<appname>`. You can run the following command in the CLI: ./splunk remove app [appname] -auth <username>:<password>
3. You may need to remove user-specific directories created for your app or add-on by deleting any files found here: `$SPLUNK_HOME/etc/users/*/<appname>`
4. Restart the Splunk platform.

Managing app and add-on configurations and properties

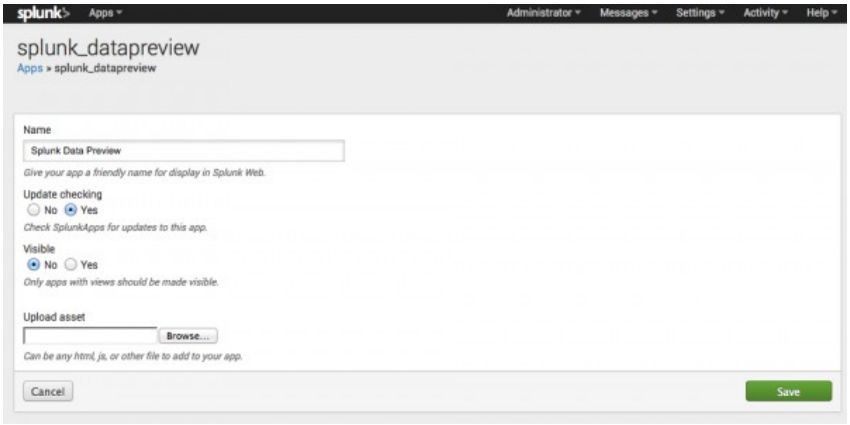
You can manage the configurations and properties for apps installed in your Splunk Enterprise instance from the Apps menu. Click on **Apps** in the User bar to select one of your installed apps or manage an app. From the Manage Apps page, you can do the following:

- Edit permissions for an app or add-on
- Enable or disable an app or add-on
- Perform actions, such as launch the app, edit the properties, and view app objects

Edit app and add-on properties

The edits you make to configuration and properties depend on whether you are the owner of the app or a user.

Select **Apps > Manage Apps** then click **Edit properties** for the app or add-on you want to edit. You can make the following edits for apps installed in this Splunk Enterprise instance.



- **Name:** Change the display name of the app or add-on in Splunk Web.
- **Update checking:** By default, update checking is enabled. You can override the default and disable update checking. See Checking for app an add-on updates below for details.
- **Visible:** Apps with views should be visible. Add-ons, which often do not have a view, should disable the visible property.
- **Upload asset:** Use this field to select a local file asset files, such as an HTML, JavaScript, or CSS file that can be accessed by the app or add-on. You can only upload one file at a time from this panel.

Refer to Develop Splunk apps for Splunk Cloud or Splunk Enterprise on the Splunk Developer Portal for details on the configuration and properties of apps and add-ons.

For a discussion of app object permissions, and governing access to those objects, see Set app permissions using Splunk Web on the Splunk Developer Portal.

Checking for updates

You can configure Splunk Enterprise to check Splunkbase for updates to an app or add-on. By default, checking for updates is enabled. You can disable checking for updates for an app by editing this property from **Settings > Apps > Edit properties**.

However, if this property is not available in Splunk Web, you can also manually edit the apps `app.conf` file to disable checking for updates. Create or edit the following stanza in `$SPLUNK_HOME/etc/apps/<app_name>/local/app.conf` to disable checking for updates:

```
[package]
check_for_updates = 0
```

Note: Edit the local version of `app.conf`, not the default version. This avoids overriding your setting with the next update of the app.

Install SPL2-based apps

Beta features are provided by Splunk to you "as is" without any warranties, maintenance and support, or service level commitments. Splunk makes this Beta feature available in its sole discretion and may discontinue it at any time. Use of Beta features is subject to the Splunk General Terms.

SPL, version 2 (SPL2) is a product-agnostic, intuitive language that has the best of both query and programming languages. SPL2 supports SPL and SQL syntax patterns.

For this Beta you must use a **pre-release version of Splunk Enterprise**. These Beta features are not included in version 9.2.1, or any other released version of Splunk Enterprise.

This documentation is designed for Splunk administrators who are participating in the Beta of SPL2-based application development. Refer to SPL2 Public Beta overview in the *Splunk Developer Guide* on dev.splunk.com for more information about using the pre-release version of the Beta, which can be found on the Splunk Voice of the Customer portal.

This Beta is supported on the following operating systems:

- Linux
- MacOS

You can install and run SPL2-based applications in a pre-release version of Splunk Enterprise as part of this Beta.

Supported architectures

This Beta is supported on the following architectures which are described in the PDF Validated Splunk Architectures:

- Single Server Deployment (SVA S1)
- Distributed Non-Clustered Deployment (D1)
- Distributed Clustered Deployment - Single Site (C1 / C11)
- Distributed Clustered Deployment + SHC - Single Site (C3 / C13)

New terminology

The following table describes some of the new terms you might encounter in this documentation:

Term	Description
dataset	A dataset is a collection of data. Indexes, lookups, views, and jobs are different kinds of datasets.
statements	SPL2 statements are searches and other types of data-related code, such as: • Import and export statements • Custom function statements • Custom data type statements
module	A module is like a file that contains one or more SPL2 statements.
data orchestrator	The Splunk data orchestrator is a new software component that parses and routes SPL2 modules to splunkd.

For more about modules, datasets, and statements, see the following documentation in the *SPL2 Search Manual*:

- Datasets
- Modules and SPL2 statements
- Extend and branch search statements

For more information about SPL2-based applications, see Create a SPL2-based app in the *Developer Guide* on dev.splunk.com.

Prerequisites

- A pre-release version of Splunk Enterprise.
  - To install an SPL2-based application, you must download the pre-release version of Splunk Enterprise. To get the pre-release version, access the **SPL2 Public Beta for Application Development** on the Splunk Voice of the Customer site. You must login with a `splunk.com` account to access the Beta.

- Port 9800.
  - The pre-release version has SPL2 enabled and uses port 9800 to connect to the Splunk data orchestrator. The Splunk data orchestrator is a new software component that parses and routes SPL2 modules to splunkd. The data orchestrator creates a log file in the `SPLUNK_HOME/var/log/splunk` directory.
  - If for some reason port 9800 is not available to use for this Beta, you can designate another port to connect to the data orchestrator. See Edit the SPL2 configuration on the pre-release instance in the *Splunk Developer Guide*.

Get help or provide feedback on the Beta

Use slack or email to request help or make comments about this Beta:

- Use the `#spl2` channel in the splunk-usergroups Slack workspace.
- Email us at `spl2@splunk.com`.

Beta limitations

The following sections describe the current limitations in this Beta. These sections are updated when a limitation is removed or changed.

Orchestrator service error on macOS

If you receive an error referencing the orchestrator service after you install the pre-release version of Splunk Enterprise on macOS, there is an issue with the $TMPDIR setting.

Try following steps to workaround this issue:

1. Open a different terminal window.
2. Restart the pre-release Splunk Enterprise instance. This might reset the $TMPDIR setting.

If restarting the pre-release version of Splunk Enterprise does not resolve the issue, then try these steps:

1. Set $TMPDIR to a known existing, writable directory.
   For example: `export TMPDIR=$SPLUNK_HOME/spl2_install_tmp`
2. Restart the pre-release Splunk Enterprise instance.

Dataset limitations

In this Beta, only the following types of datasets are supported:

- Indexes

- Lookups
- Saved searches

Knowledge object limitations

In this Beta, the supported knowledge objects (KOs) are identified in the following table:

Knowledge object	Supported in this Beta
Alerts	Yes
Dashboards	Yes
Data models	No
Event types	No
Fields	Yes
Field extractions	No
Lookups	Yes
Reports	Yes
Saved searches	Yes
Tags	No
Workflow actions	No

SPL2 function scope limitation

An SPL2 custom function cannot reference a search statement that is defined outside of the SPL2 function. For more information about this limitation, see SPL2 Public Beta overview in the *Splunk Developer Guide* on dev.splunk.com.

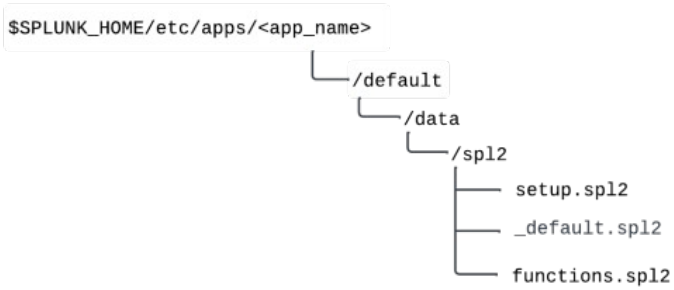
Install an SPL2-based app

In this Beta, Splunk administrators can install and use SPL2-based applications on the pre-release version of Splunk Enterprise.

Complete the following steps to install a SPL2-based application. For information about basic app installation, see About installing Splunk add-ons in the *Splunk Add-ons* manual.

1. Save the SPL2-app on your pre-release version of Splunk Enterprise.
2. On the Splunk Web home screen, select the **Apps** drop-down and then select **Manage apps**.
3. Select the **Install app from file** button.
4. Locate the app file and select **Upload**. You might be prompted to restart the Splunk Enterprise instance.
5. Verify that the app appears in the list of apps and add-ons. You can also find the app on your pre-release instance at $SPLUNK_HOME/etc/apps/<app_name>.
6. Read the README file that is included with the app.

The application is installed in the /apps/default/data/spl2 directory. Modules are not installed on indexers. The following image shows an app that consists of 3 modules: setup, _default, and functions.



After installation, all application modules in the /apps/default/data/spl2 directory are automatically uploaded and stored in your instance. If the files in your /apps/local/data/spl2 and /apps/default/data/spl2 directories have the same name, then the local directory takes precedence. The file in the local directory is uploaded instead, but both files are preserved in their respective directories.

If you make changes to these modules in these directories later, they will not automatically upload unless you re-install the app. This process occurs only at installation. To learn how to modify an app later, see Manage SPL2-based apps.

See also

- To learn how to modify an SPL2-based app, see Manage SPL2-based apps.
- To learn how to create an SPL2-based app, see Create a SPL2-based app in the *Developer Guide for Splunk Cloud Platform and Splunk Enterprise* on the Splunk Developer Portal.

Manage SPL2-based apps

Beta features are provided by Splunk to you "as is" without any warranties, maintenance and support, or service level commitments. Splunk makes this Beta feature available in its sole discretion and may discontinue it at any time. Use of Beta features is subject to the Splunk General Terms.

For this Beta you must use a **pre-release version of Splunk Enterprise**. These Beta features are not included in version 9.2.1, or any other released version of Splunk Enterprise.

This documentation is designed for Splunk administrators who are participating in the Beta of SPL2-based application development. Refer to SPL2 Public Beta overview in the *Splunk Developer Guide* on dev.splunk.com for more information about using the pre-release version of the Beta, which can be found on the Splunk Voice of the Customer portal.

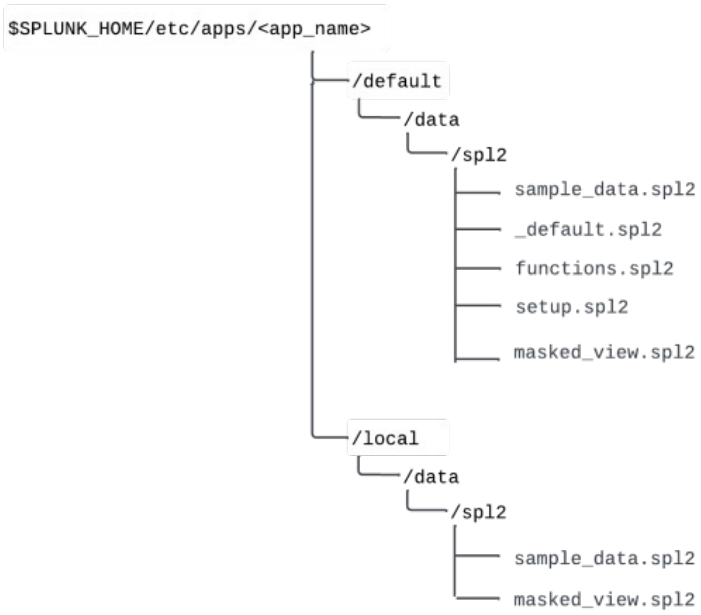
Modify an SPL2-based application

To made changes to an SPL2-based application beyond just updating permissions, as a best practice, update and reinstall the app. Modifying the app contents in the local directory alone is not sufficient to cause the app to update. If you just modify the local directory, completely reinstalling the app is necessary for changes to take effect.

When you reinstall an app, modules with the exact same name are overwritten.

Complete the following steps to change an SPL2-based app:

1. Copy the modules that you want to change into a local directory under the `<app_name>` directory. The local directory structure must be `/local/data/spl2`. The following image shows only the `sample_data` and `masked_view` modules copied into the local directory path:



2. Modify the files in the local directory. The local directory takes precedence over the default directory. Note that the default module must be named `_default` to provide implicit SPL2 imports to the app's search interface.
3. Reinstall the app.

As an alternative, to avoid reinstalling the app, use the **modules** endpoints to modify a module. For examples of how to use these endpoints to modify a module, see Endpoints for SPL2-based application in the *REST API Reference Manual*.

To fully customize an SPL2-based app by downloading your modules, re-packaging them, and re-editing them, use the same steps as the application developer. See Package your SPL2-based application in the *Splunk Developer Guide* on the Splunk Developer Portal.

Modify permissions for modules

Permissions for SPL2-based apps are set by role and on a per-module basis. Module-level permissions are set by using the Splunk Enterprise API, and they are distinct from app-level permissions, which are set in Splunk Web.

By default, the admin and power roles have read, write, and execute permissions on all modules, while individuals with the user role have no permissions on any modules. As a result, during this Beta, users can create modules that, by default, they cannot see, modify, or run.

- The `execute` permission enables the role to run a search.
- The `read` permission enables the role both to run a search and to see a module definition. The read permission supersedes the execute permission.
- The `write` permission enables the role to perform create, update, and delete operations.

You can modify role-based permissions for SPL2-based apps at the module level by using the permissions API endpoints. For details on the permissions endpoints, see Search endpoint descriptions in the *REST API Reference Manual*.

> Calls to the permissions endpoints must specify all of the operations, even if you are only changing one of the operations.

Example of setting role based permissions

Suppose an SPL2-based app contains a module called "masked_data" that takes events from an index and uses a search to return the same set of events with the email addresses masked out. The search that masks the events is exported as a view which can be used in other modules in the app. While you can't set permissions on the masked view itself, you can set permissions on the "masked_data" module.

Here's an example of the permissions endpoint that you use to add the `user` role to the `execute` operation for the `masked_data` module. The other permissions remain unchanged from the default permissions:

```
curl -k -u admin:pass https://localhost:8089/services/spl2/permissions \
--data '{
  "resourceType": "modules",
  "resourceName": "masked_data",
  "permissions": [
     {
      "operation": "execute",
      "roles": [
        "admin",
        "power",
        "user"
      ]
    },
    {
      "operation": "read",
      "roles": [
        "admin",
        "power"
      ]
    },
    {
      "operation": "write",
      "roles": [
        "admin",
        "power"
      ]
    }
  ]
}'
```

Run a module as if you are its owner

Modules can be marked with `@run_as_owner;` as the first line of that module to enable the module to be run by others as if they were the module's owner.

> For this Beta, anyone with write permission on a module can add the @run_as_owner; line to it, not only the module's owner. Before users can run a module as if they were the module's owner, however, a Splunk administrator must set run_as_owner_enabled to true in the [spl2] stanza of the server.conf file. The option to use @run_as_owner; is disabled by default.

Complete the following steps to add `@run_as_owner;` to a module.

1. Set the `run_as_owner_enabled` option to true in the `[spl2]` stanza of the `server.conf` file. A Splunk administrator must complete this step.
2. Ensure you have write permission for the module that you want to modify.
3. Add the line `@run_as_owner;` as the first line of that module in the local directory of the <app_name> directory.
4. Re-install the app.

A Splunk administrator still needs to grant users execute permission for the module before they can run it. Users can run a module without read permission, but they need read permission to see its definition.

The following requirements and limitations apply to modules marked with `@run_as_owner;`.

- If you add the `@run_as_owner;` line to a module, and then another user runs the module, any indexes or other datasets that the module references are accessed on the owner's behalf. If a user imports a module with this line and runs a search, then search quotas are also calculated on the owner's behalf.
- Only private apps can be marked `@run_as_owner;`. For this Beta, any user with write permission on a module can add the `@run_as_owner;` line, but as a best practice, only the owner of a module should modify it.
- Modules marked with `@run_as_owner;` can only import other modules marked with `@run_as_owner;` if they have the same owner. Imported modules can have a different owner as long as they are not also marked `@run_as_owner;`.
- Only exported views are run as the owner of the module. Exported functions still run as the user importing the function.

The @run_as_owner; option can pose security risks or cause data loss if a module uses any risky commands from SPL1, or uses the into command in SPL2. If the module owner does not have safeguards enabled, safeguards for risky commands are not applied when a user runs a module as owner.

For more information about risky commands, see SPL safeguards for risky commands in the *Securing Splunk Cloud Platform* manual.

See also

- To learn how to install an SPL2-based app, see Install SPL2-based apps.
- To learn how to create an SPL2-based app, see Create a SPL2-based app in the *Developer Guide for Splunk Cloud Platform and Splunk Enterprise* on the Splunk Developer Portal.

Manage users

About users and roles

You can create users with passwords and assign them to **roles** that you have created. Splunk Enterprise Free does not support user authentication.

Create users

Splunk Enterprise supports three types of authentication systems, which are described in the *Securing Splunk Enterprise* manual.

- **Native authentication.** See "Set up user authentication with Splunk Enterprise native authentication" for more information.
- **LDAP.** Splunk supports authentication with its internal authentication services or your existing LDAP server. See "Set up user authentication with LDAP" for more information.
- **Scripted authentication API.** Use scripted authentication to connect Splunk native authentication with an external authentication system, such as RADIUS or PAM. See "Set up user authentication with external systems" for more information.

About roles

Users are assigned to roles. A role contains a set of **capabilities**. Capabilities specify what actions are available to roles. For example, capabilities determine whether someone with a particular role is allowed to add inputs or edit saved searches. The various capabilities are listed in "About defining roles with capabilities" in the *Securing Splunk Enterprise* manual.

By default, Splunk Enterprise comes with the following roles predefined:

- admin -- this role has the most capabilities assigned to it.
- power -- this role can edit all shared objects (saved searches, etc) and alerts, tag events, and other similar tasks.
- user -- this role can create and edit its own saved searches, run searches, edit its own preferences, create and edit event types, and other similar tasks.
- can_delete -- This role allows the user to delete by keyword. This capability is necessary when using the delete search operator.

Note Do not edit the predefined roles. Instead, create custom roles that inherit from the built-in roles, and modify the custom roles as required.

For detailed information on roles and how to assign users to roles, see the chapter "Users and role-based access control" in the *Securing Splunk Enterprise* manual.

Find existing users and roles

To locate an existing user or role in Splunk Web, use the Search bar at the top of the Users or Roles page in the Access Controls section by selecting **Settings > Access Controls**. Wildcards are supported. By default Splunk Enterprise searches in all available fields for the string that you enter. To search a particular field, specify that field. For example, to

search only email addresses, type "email=<*email address or address fragment*>:, or to search only the "Full name" field, type "realname=<*name or name fragment*>. To search for users in a given role, use "roles=".

Configure user language and locale

When a user logs in, Splunk automatically uses the language that the user's browser is set to. To switch languages, change the browser's locale setting. Locale configurations are browser-specific.

Splunk detects locale strings. A locale string contains two components: a language specifier and a localization specifier. This is usually presented as two lowercase letters and two uppercase letters linked by an underscore. For example, "en_US" means US English and "en_GB" means British English.

The user's locale also affects how dates, times, numbers, etc., are formatted, as different countries have different standards for formatting these entities.

Splunk provides built-in support for these locales:

```
de_DE
en_GB
en_US
fr_FR
it_IT
ja_JP
ko_KR
zh_CN
zh_TW
```

If you want to add localization for additional languages, refer to "Translate Splunk" in the Developer manual for guidance. You can then tell your users to specify the appropriate locale in their browsers.

How browser locale affects timestamp formatting

By default, timestamps in Splunk are formatted according the browser locale. If the browser is configured for US English, the timestamps are presented in American fashion: `MM/DD/YYYY:HH:MM:SS`. If the browser is configured for British English, then the timestamps will be presented in the European date format: `DD/MM/YYYY:HH:MM:SS`.

For more information on timestamp formatting, see Configure timestamp recognition in *Getting Data In*.

You can also specify how the timestamps appear in your search output by including formatting directly in your search. See Date and time format variables in the *Search Reference*.

Override the browser locale

The locale that Splunk uses for a given session can be changed by modifying the URL that you use to access Splunk. Splunk URLs follow the form `http://host:port/locale/...`. For example, when you access Splunk to log in, the URL might appear as `https://hostname:8000/en-US/account/login` for US English. To use British English settings, you can change the locale string to `https://hostname:8000/en-GB/account/login`. This session then presents and accepts

timestamps in British English format for its duration.

Requesting a locale for which the Splunk interface has not been localized results in the message: `Invalid language Specified`.

Refer to "Translate Splunk" in the Developer Manual for more information about localizing Splunk.

Configure user session timeouts

The amount of time that elapses before a user session with a Splunk platform instance times out depends on the interaction among three timeout settings:

- The `splunkweb` session timeout.
- The `splunkd` session timeout.
- The browser session timeout.

After the session times out, the next time the user sends a network request to the Splunk platform instance, it prompts them to log in again.

The `splunkweb` and `splunkd` timeouts determine the maximum idle time in the interaction between browser and the Splunk platform instance. The browser session timeout determines the maximum idle time in interaction between the user and browser.

The `splunkweb` and `splunkd` timeouts generally have the same value, as the same field sets both of them.

Set the user session timeout in Splunk Web

1. Click **Settings** in the upper right-hand corner of Splunk Web.
2. Under System, click **Server settings**.
3. Click **General settings**.
4. In the **Session timeout** field, enter a timeout value.
5. Click **Save**.

This sets the user session timeout value for both the `splunkweb` and `splunkd` services. Initially, they share the same value of 60 minutes. They will continue to maintain identical values if you change the value through Splunk Web.

If you want to set the timeouts for `splunkweb` and `splunkd` to different values, you can do so by editing the configuration files, `web.conf` setting `tools.sessions.timeout`, and the `server.conf` setting `sessionTimeout`. There's no specific reason to give them different values. If the user is using Splunk Web to access the Splunk Enterprise instance, the smaller of the two timeout attributes prevails. For example, if the `web.conf` setting `tools.sessions.timeout` is set to "90" (minutes), and the `server.conf` setting `sessionTimeout` is set to "1h" (1 hour, or 60 minutes), the session uses the smallest timeout of 60 minutes.

In addition to setting the `splunkweb`/`splunkd` session value, you can also specify the timeout for the user browser session by editing the `ui_inactivity_timeout` value in `web.conf`. The Splunk browser session will time out once this value is reached. The default is 60 minutes. If `ui_inactivity_timeout` is set to less than 1, there's no timeout -- the session will stay alive while the browser is open.

The countdown for the `splunkweb`/`splunkd` session timeout does not begin until the browser session reaches its timeout value. So, to determine how long the user has before timeout, add the value of `ui_inactivity_timeout` to the smaller of

the timeout values for `splunkweb` and `splunkd`. For example, assume the following:

- `splunkweb` timeout: 15m
- `splunkd` timeout: 20m
- browser (`ui_inactivity_timeout`) timeout: 10m

The user session stays active for 25 minutes (15m+10m). After 25 minutes of no activity, the session ends, and the instance prompts the user to log in again the next time they send a network request to the instance.

If you change a timeout value, either in Splunk Web or in configuration files, you must restart the Splunk platform instance for the change to take effect.

Configure Splunk Enterprise to use proxies

Use a forward Proxy Server for splunkd

You can set up an HTTP/S proxy server so that all HTTP/S traffic originating from splunkd flows through that proxy server. This lets you manage and control communication between different splunkd instances and lets you manage requests that splunkd makes over the Internet.

How it works

When a client (splunkd) sends a request to the HTTP proxy server, the forward proxy server validates the request.

- If a request is not valid, the proxy rejects the request and the client receives an error or is redirected.
- If a request is valid, the forward proxy checks whether the requested information is cached.
    - If a cached copy is available, the forward proxy serves the cached information.
    - If the requested information is not cached, the request is sent to an actual content server which sends the information to the forward proxy. The forward proxy then relays the response to the client.

This process configures Splunk to Splunk communication through a Proxy. The settings documented here do not support interactions outside of Splunk, for example:

- Access to Splunkbase via Splunk Web
- Splunk external lookups
- Actions that make a REST API call to an external service outside of a firewall

Configure a forward Proxy Server for splunkd

To set up HTTP Proxy Server support for splunkd:

1. Download and configure a HTTP proxy server and configure it to talk to splunkd on a Splunk node. Splunk Enterprise supports the following proxy servers:

- Apache Server 2.4
- Apache Server 2.2
- Squid Server 3.5

2. Configure splunkd proxy settings by setting the proxy variables in `server.conf` or using the REST endpoints

Note: TLS Proxying is currently not supported, the proxy server must be configured to listen on a non-SSL port.

Install and configure your HTTP Proxy Server for splunkd

You can set up an HTTP proxy server for splunkd so that all HTTP/S traffic originating from splunkd flows through the proxy server, making your traffic easier to manage.

Splunk Software officially supports the following HTTP proxy servers:

- Apache Server 2.4
- Apache Server 2.2
- Squid Server 3.5

Note: Splunk Enterprise supports the HTTP CONNECT method for HTTPS requests. TLS proxying is not supported, and the proxy server cannot listen on an SSL port.

Configure Apache Server 2.4

1. Download the latest version of Apache server 2.4 at http://httpd.apache.org/download.cgi.

2. Extract and install it on the machine that will run the proxy server. The following example compiles the server from source.

```
gzip -d httpd-2.4.25.tar.gz
tar xvf httpd-2.4.25.tar
cd httpd-NN
./configure --prefix=$PROXY_HOME
make install
```

3. Customize the the Apache server `httpd.conf` file.

```
Listen = 8000 <IP addresses and ports that the server listens to>
ProxyRequests = On < Enables forward (standard) proxy requests>
SSLProxyEngine = On <This directive toggles the usage of the SSL/TLS Protocol Engine for proxy>
AllowCONNECT = 443 <Ports that are allowed to CONNECT through the proxy>
```

Additional configuration (optional)

Before you configure or disable these values, please read the Apache documentation for additional information.

```
SSLProxyVerify = optional <When a proxy is configured to forward requests to a remote SSL server, this
setting can configure certificate verification of the remote server>
SSLProxyCheckPeerCN = on <determines whether the remote server certificate's CN field is compared against
the hostname of the request URL>
SSLProxyCheckPeerName = on <turns on host name checking for server certificates when mod_ssl is acting as
an SSL client>
SSLProxyCheckPeerExpire = on <enables certificate expiration checking>
```

Configure Apache Server 2.2

1. Download the latest version of Apache server 2.2 at http://httpd.apache.org/download.cgi.

2. Extract and install it on the machine that will run the proxy server. The following example compiles the server from source.

```
$ gzip -d httpd-2.2.32.tar.gz
$ tar xvf httpd-2.2.32.tar
$ cd httpd-NN
$ ./configure --prefix="PROXY_HOME" --enable-ssl --enable-proxy --enable-proxy-connect --enable-proxy-http
$ make install
```

3. Customize the Apache server's `httpd.conf` file:

```
Listen 8000 <This is the list of IP addresses and ports that the server listens to>
ProxyRequests = On <Enables forward (standard) proxy requests>
SSLProxyEngine = On <This directive toggles the usage of the SSL/TLS Protocol Engine for proxy>
AllowCONNECT 443 <Ports that are allowed to CONNECT through the proxy>
```

Additional configuration (optional)

Before you modify or disable these settings in your environment, please read the Apache documentation for additional information.

```
SSLProxyVerify = optional <When a proxy is configured to forward requests to a remote SSL server, this
directive can be used to configure certificate verification for the remote server.>
SSLProxyCheckPeerCN = on <Determines whether the remote server certificate's Common Name field is compared
against the hostname of the request URL>
SSLProxyCheckPeerName = on <Configures host name checking for server certificates when mod_ssl is acting as
an SSL client>
SSLProxyCheckPeerExpire = on <when turned on, the systems checks whether if the remote server certificate is
expired or not>
```

Configure Squid 3.5

1. Download the latest version of Squid server 3.5 at http://www.squid-cache.org/Download/.

2. Extract and install the download on the machine that will run the proxy server. The following example compiles Squid server 3.5 from source.

```
$ tar xzf squid-3.5.23.tar.gz
$ cd squid-3.5.23
$ ./configure --with-openssl
$ make
$ make install
```

3. Configure the Squid server's `squid.conf` file

```
acl localnet src = <configure all possible internal network ports, a new line for each port>
acl SSL_ports = <configure all acl SSL_ports, a new line for each port>
acl CONNECT method CONNECT <ACL for CONNECT method>
http_port 8000 <Port on which the Squid server will listen for requests>
```

Additional configuration (optional)

Before you configure or disable these settings in your environment, please read the Squid documentation for additional information.

```
sslproxy_cert_error deny all <Use this ACL to bypass server certificate validation errors>
sslproxy_flags DONT_VERIFY_PEER <Various flags modifying the use of SSL while proxying https URLs>
hosts_file PROXY_HOME/hosts <Location of the host-local IP name-address associations database>
```

Configure splunkd to use your HTTP Proxy Server

You can set up an HTTP proxy server for splunkd so that all HTTP/S traffic originating from splunkd flows through the proxy server.

To set up a proxy server for splunkd, you can either configure Splunk's proxy variables in `server.conf` or configure the REST endpoints.

This process configures Splunk to Splunk communication through a Proxy. The settings documented here do not support interactions outside of Splunk, for example:

- Access to Splunkbase via Splunk Web
- Splunk external lookups
- Actions that make a REST API call to an external service outside of a firewall

Edit `server.conf` to configure splunkd to work with your server proxy

For a single Splunk Enterprise instance, you can add the proxy configs under `%SPLUNK_HOME/etc/system/local`, or deploy a custom app that includes a server.conf file with your proxy settings. To configure multiple instances (pool of indexers, search head cluster, etc.) use a deployment management tool such as the deployer, deployment server, or cluster manager node to deploy an app that includes a server.conf file with your proxy settings.

```
[proxyConfig]
http_proxy = <string that identifies the server proxy. When set, splunkd sends all HTTP requests through
this proxy server. The default value is unset.>
https_proxy = <string that identifies the server proxy. When set, splunkd sends all HTTPS requests through
the proxy server defined here. If not set, splunkd uses the proxy defined in http_proxy. The default value
is unset.>
no_proxy = <string that identifies the no proxy rules. When set, splunkd uses the [no_proxy] rules to decide
whether the proxy server needs to be bypassed for matching hosts and IP Addresses. Requests going to
localhost/loopback address are not proxied. Default is "localhost, 127.0.0.1, ::1">
```

Use REST endpoints to configure splunkd to work with your server proxy

You can also configure splunkd to work with your HTTP proxy server by modifying the `/services/server/httpsettings/proxysettings` REST endpoint. To set variables using a REST endpoint, you must have the `edit_server` capability.

Create the `[proxyConfig]` stanza:

```
curl -k /services/server/httpsettings/proxysettings --data name="proxyConfig"
```

Write to the stanza:

```
curl -k /services/server/httpsettings/proxysettings/proxyConfig --data
"http_proxy=....&https_proxy=...&no_proxy=...."
```

Read from stanza:

```
curl -k /services/server/httpsettings/proxysettings/proxyConfig
```

Delete the stanza:

```
curl -k -X DELETE /services/server/httpsettings/proxysettings/proxyConfig
```

For more details and example requests and responses, see server/httpsettings/proxysettings and server/httpsettings/proxysettings/proxyConfig in the *REST API Reference*.

Configure clusters to work with a proxy

To use a proxy server for communication in an indexer cluster or search head cluster, configure the `register_replication_address` setting under the clustering or shclustering stanza in server.conf.

Best practices when configuring an HTTP Proxy Server for splunkd

You can set up an HTTP proxy server for splunkd so that all HTTP/S traffic originating from splunkd flows through the proxy server.

Points to Remember

1. Splunk supports only non-TLS proxying. Proxy servers listening directly on HTTPS are not supported.

2. Verify your proxy settings for accuracy and make sure they comply with your organization's network policies.

3. For performance issues with the proxy server, see the performance tuning tips below.

Performance Tuning with Apache Server

If you have a large number of clients communicating through the proxy server, you might see a performance impact for those clients. In the case of performance impact:

- Check that the proxy server is adequately provisioned in terms of CPU and memory resources.
- Use the different multi-processing modules (MPM) and tune the following settings depending on the requirements of your environment. Check the Apache documentation for additional information.

```
ServerLimit  <Upper limit on configurable number of processes>
StartServers  <Number of child server processes created at startup>
MaxRequestWorkers  <Maximum number of connections that will be processed simultaneously>
MinSpareThreads  <Minimum number of idle threads available to handle request spikes>
MaxSpareThreads  <Maximum number of idle threads>
ThreadsPerChild  <Number of threads created by each child process>
```

Performance Profiling with Squid Server

If you have a large number of clients communicating through the proxy server, you might see a performance impact for those clients. Please make sure that the proxy server is adequately provisioned in terms of CPU & Memory resources. Please check the Squid profiling documentation for additional information.

Use Splunk Web with a reverse proxy configuration

Splunk web can be placed behind a proxy in a reverse proxy type of configuration. If you are hosting Splunk Web behind a proxy that does not place Splunk Web at the proxy's root, you may need to configure the `root_endpoint` setting in `$SPLUNK_HOME/etc/system/local/web.conf`.

For example if your proxy hosts Splunk Web at "yourhost.com:9000/splunk", `root_endpoint` should be set to `/splunk`.

Note: The App Manager is not supported for use with a proxy server, if you use a proxy server with Splunk Web, you must download and update apps manually.

Typical Reverse Proxy Configuration

Lets take an example where Splunk Web is accessed via `http://splunk.example.com:8000/lzone` instead of `http://splunk.example.com:8000/`.

For enable this behavior, please set the following in `web.conf`

```
root_endpoint=/lzone
```

For a Apache proxy server, you would then make it visible to the proxy by mapping it in `httpd.conf`. Please check the documentation for additional information.

```
# Maps remote servers into the local server URL-space
ProxyPass /lzone http://splunkweb.splunk.com:8000/lzone

#Adjusts the URL in HTTP response headers sent from a reverse proxied server
ProxyPassReverse /lzone http://splunkweb.splunk.com:8000/lzone
```

Meet the Splunk AMI

About the Splunk Enterprise AMI

Splunk Enterprise is available as an Amazon Machine Image on the Amazon Web Services Marketplace.

What is the Splunk Enterprise AMI?

The Splunk Enterprise AMI is an Amazon Machine Image consisting of Splunk Enterprise running on Amazon Linux.

The image includes a Splunk Enterprise Trial license. To learn about the license features and time limits, see Types of Splunk Enterprise licenses.

Get the Splunk Enterprise AMI with 1-click

1. From the AWS Marketplace, select Splunk Enterprise AMI.
2. On the overview tab, select "Continue to subscribe."
3. Once the subscription authorization is complete, select "Continue to Configuration."
4. Confirm Splunk Enterprise version and the region selected. Select "Continue to Launch."
5. On the Launch this software page:
    1. Choose an EC2 instance type. Select an instance type with sufficient storage and resources to support your use-case. See Introduction to capacity planning for Splunk Enterprise in the *Capacity Planning Manual* for more information.
    2. In "Security Group Settings" select a security group.
    3. In "Key Pair Settings" select or create a key pair.
6. Select "Launch"
7. Make note of the ports that are opened in your chosen security group. The typical ports are: 8089 (Splunk Enterprise Management), 8000 (Splunk Web), 9997 (Splunk Forwarder listener), 22 (SSH), and 443 (SSL/HTTPS). For more information about open ports and security, see About securing Splunk software and How to secure and harden your Splunk software installation in *Securing Splunk Enterprise*.

Start using the Splunk Enterprise AMI

If you've already started a copy of the Splunk Enterprise AMI on the AWS Marketplace, then you'll have an instance of Splunk Enterprise running as the Splunk user. The Splunk Enterprise services will start when the machine starts.

Find Splunk Web

1. In your EC2 Management Console, find your instance running Splunk Enterprise. Note the instance ID and public IP address.
2. Paste the public IP into a new browser tab. Do not hit enter yet.
    1. Append the Splunk Web port to the end of the IP address. Example: http://$aws_public_ip:8000
    2. Hit enter.
3. Log into Splunk Enterprise with the default AMI credentials:
    1. For Splunk Enterprise version 7.2.5 and later:
        1. username: `admin`
        2. password: `SPLUNK-$instance id$`
        3. It is recommended that you change your password after login.
    2. For older Splunk Enterprise versions:

1. username: `admin`
2. password: `$instance id$`
3. On the next screen, set a new password.

Next tasks

- Learn how to run simple searches and generate reports from data in Splunk Enterprise by following along with the Search Tutorial.
- Learn how to access your AMI instance file system using SSH in the Connect to your Linux instance at the Amazon Elastic Compute Cloud documentation.
- Learn about Splunk Enterprise knowledge objects in the Knowledge Manager Manual.
- For an overview of tasks in Splunk Enterprise and where you can find more information about them, see Splunk administration: the big picture in the *Admin Manual* .

Upgrade

Upgrade Splunk Enterprise version

See "How to upgrade Splunk" in the *Installation Manual*. Be sure to run a backup before you begin the upgrade.

Upgrade your AWS storage capacity

See the AWS documentation about Amazon EBS.

Upgrade your AWS compute capacity

See the AWS documentation about Amazon EC2.

Get help

To find community resources and get help, see Get Started with Splunk Community. To purchase a Splunk Enterprise license and support, contact sales@splunk.com.

Configuration file reference

alert_actions.conf

The following are the spec and example files for `alert_actions.conf`.

alert_actions.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure global saved search actions in the alert_actions.conf file.
# Saved searches are configured in the savedsearches.conf file.
#
# There is an alert_actions.conf file in the $SPLUNK_HOME/etc/system/default/
# directory. Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name
# alert_actions.conf in the $SPLUNK_HOME/etc/system/local/ directory.
# Then add the specific settings that you want to customize to the local
# configuration file.
# For examples, see alert_actions.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top
#    of the file.
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, settings are combined. In the case of
#    multiple definitions of the same setting, the last definition in the
#    file wins.
#  * If a setting is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.

maxresults = <integer>
* The global maximum number of search results sent through alerts.
* Default: 10000

hostname = [protocol]<host>[:<port>]
```

```
* The hostname in the web link (URL) that is sent in alerts.
* This value accepts two forms:
  * hostname
       examples: splunkserver, splunkserver.example.com
  * protocol://hostname:port
       examples: http://splunkserver:8000, https://splunkserver.example.com:443
* When this value is a hostname, the protocol and port that
  are configured in the Splunk platform are used to construct the base of
  the URL.
* When this value begins with 'http://', it is used verbatim.
  NOTE: This means the correct port must be specified if it is not
  the default port for http or https.
* This is useful in cases when the Splunk server is not aware of
  how to construct an externally referenceable URL, such as SSO
  environments, other proxies, or when the Splunk server hostname
  is not generally resolvable.
* Default: The current hostname provided by the operating system,
  or if that fails, "localhost".

ttl = <integer>[p]
* The minimum time to live, in seconds, of the search artifacts,
  if this action is triggered.
* If 'p' follows '<integer>', then '<integer>' is the number of scheduled periods.
* If no actions are triggered, the ttl for the artifacts are determined
  by the 'dispatch.ttl' setting in the savedsearches.conf file.
* Default: 10p
* Default (for email, rss)                    : 86400 (24 hours)
* Default (for script)                        :   600 (10 minutes)
* Default (for summary_index):   120 (2 minutes)

maxtime = <integer>[m|s|h|d]
* The maximum amount of time that the execution of an action is allowed to
  take before the action is aborted.
* Use the d, h, m and s suffixes to define the period of time:
  d = day, h = hour, m = minute and s = second.
  For example: 5d means 5 days.
* Default (for all stanzas except 'rss': 5m
* Default (for the 'rss' stanza): 1m

track_alert = <boolean>
* Whether or not the execution of this action signifies a trackable alert.
* Default: 0 (false).

command = <string>
* The search command (or pipeline) that is responsible for executing
  the action.
* Generally the command is a template search pipeline which is realized
  with values from the saved search. To reference saved search
  field values enclose the values in dollar signs ($).
* For example, to reference the saved search name, use "$name$". To
  reference the search, use "$search$"

is_custom = <boolean>
* Whether or not the alert action is based on the custom alert
  actions framework and is supposed to be listed in the search UI.

payload_format = [xml|json]
* Configure the format the alert script receives the configuration via
  STDIN.
* Default: xml

label = <string>
```

```
* For custom alert actions, defines the label that is shown in the UI.
  If not specified, the stanza name is used instead.
* Default: The stanza name for the custom alert action.

description = <string>
* For custom alert actions, specifies the description shown in the UI.

icon_path = <string>
* For custom alert actions, defines the icon shown in the UI for the alert
  action. The path refers to the 'appserver/static' directory in the app
  that the alert action is defined in.

forceCsvResults = [auto|true|false]
* If set to "true", any saved search that includes this action
  always stores results in CSV format, instead of the internal SRS format.
* If set to "false", results are always serialized using the internal SRS format.
* If set to "auto", results are serialized as CSV if the 'command' setting
  in this stanza starts with "sendalert" or contains the string
  "$results.file$".
* Default: auto

alert.execute.cmd = <string>
* For custom alert actions, explicitly specifies the command to run
  when the alert action is triggered. This refers to a binary or script
  in the 'bin' folder of the app that the alert action is defined in, or to a
  path pointer file, also located in the 'bin' folder.
* If a path pointer file (*.path) is specified, the contents of the file
  is read and the result is used as the command to run.
  Environment variables in the path pointer file are substituted.
* If a python (*.py) script is specified, it is prefixed with the
  bundled python interpreter.

alert.execute.cmd.arg.<n> = <string>
* Provide additional arguments to the 'alert.execute.cmd'.
  Environment variables are substituted.

python.version = {default|python|python2|python3}
* For Python scripts only, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.
```

EMAIL: these settings are prefaced by the [email] stanza name

```
[email]
* Set email notification options under this stanza name.
* Follow this stanza name with any number of the following
  setting/value pairs.
* If you do not specify an entry for each setting, the default value is used.

from = <string>
* Email address from which the alert originates.
* Default: splunk

to      = <string>
* The To email address receiving the alert.

cc      = <string>
```

```
* Any courtesy copy (cc) email addresses receiving the alert.

bcc     = <string>
* Any blind courtesy copy (bcc) email addresses receiving the alert.

allowedDomainList = <comma-separated list of domains>
* Optional. This setting specifies a list of domains to which users are allowed
  to send email.
* If this setting is set for an alert, and a user adds an address with a domain
  not on this list, the Splunk software removes that address from the
  recipients list.
* 'action.email.allowedDomainList' in savedsearches.conf will not be honored.
* No default.

message.report = <string>
* Specify a custom email message for scheduled reports.
* Includes the ability to reference settings from the result,
  saved search, or job.

message.alert = <string>
* Specify a custom email message for alerts.
* Includes the ability to reference settings from result,
  saved search, or job.

subject = <string>
* Specify an alternate email subject if useNSSubject is "false".
* Default: Splunk Alert: $name$

subject.alert = <string>
* Specify an alternate email subject for an alert.
* Default: Splunk Alert: $name$

subject.report = <string>
* Specify an alternate email subject for a scheduled report.
* Default: Splunk Report: $name$

useNSSubject = <boolean>
* Whether or not to use the namespaced subject, for example, subject.report or the
  subject.
* Default: 0

escapeCSVNewline = <boolean>
* Whether to escape newlines as "\r\n" or "\n" or not in emailed CSV files.
* Default: true

footer.text = <string>
* Specify an alternate email footer.
* Default: "If you believe you've received this email in error, please
see your Splunk administrator.\\ Splunk > the key to enterprise resilience"

format = [table|raw|csv]
* Specify the format of inline results in the email.
* Previously accepted values "plain" and "html" are no longer respected
  and equate to "table".
* To make emails plain or HTML use the 'content_type' setting.
* Default: table

include.results_link = <boolean>
* Whether or not to include a link to the results.

include.search = <boolean>
* Whether or not to include the search that caused an email to be sent.
```

```
include.trigger = <boolean>
* Whether or not to show the trigger condition that caused the alert to
  fire.

include.trigger_time = <boolean>
* Whether or not to show the time that the alert was fired.

include.view_link = <boolean>
* Whether or not to show the title and a link to enable the user to edit
  the saved search.

content_type = [html|plain]
* Specify the content type of the email.
* When set to "plain", sends email as plain text.
* When set to "html", sends email as a multipart email that includes both
  text and HTML.

sendresults = <boolean>
* Whether or not the search results are included in the email. The
  results can be attached or inline, see inline (action.email.inline)
* Default: 0 (false)

inline = <boolean>
* Whether or not the search results are contained in the body of the alert
  email.
* If the events are not sent inline, they are attached as a CSV file.
* Default:  0 (false).

priority = [1|2|3|4|5]
* Set the priority of the email as it appears in the email client.
* Value mapping: 1 highest, 2 high, 3 normal, 4 low, 5 lowest.
* Default: 3

mailserver = <host>[:<port>]
* You must have a Simple Mail Transfer Protocol (SMTP) server available
  to send email. This is not included with the Splunk instance.
* Specifies the SMTP mail server to use when sending emails.
* <host> can be either the hostname or the IP address.
* Optionally, specify the SMTP <port> that the Splunk instance should connect to.
* When the 'use_ssl' setting (see below) is set to 1 (true), you
  must specify both <host> and <port>.
  (Example: "example.com:465")
* Default: localhost

use_ssl    = <boolean>
* Whether to use SSL when communicating with the SMTP server.
* When set to 1 (true), you must also specify both the server name or
  IP address and the TCP port in the 'mailserver' setting.
* Default: 0 (false)

use_tls    = <boolean>
* Whether or not to use TLS (transport layer security) when
  communicating with the SMTP server (starttls).
* Default: 0 (false)

auth_username   = <string>
* The username to use when authenticating with the SMTP server. If this is
  not defined or is set to an empty string, no authentication is attempted.
  NOTE: your SMTP server might reject unauthenticated emails.
* Default: an empty string
```

```
auth_password   = <password>
* The password to use when authenticating with the SMTP server.
  Normally this value is set when editing the email settings, however
  you can set a clear text password here and it is encrypted on the
  next Splunk software restart.
* Default: an empty string

sendpdf = <boolean>
* Whether or not to create and send the results as a PDF file.
* Default: 0 (false)

sendcsv = <boolean>
* Whether or not to create and send the results as a CSV file.
* Default: 0 (false)

allow_empty_attachment = <boolean>
* Whether or not the Splunk software attaches a CSV or PDF file to
  an alert email even when the triggering alert search does not have
  results.
* This setting sets a default for alerts that use the email alert
  action.  Override it for specific alerts by setting
  'action.email.allow_empty_attachment' for those alerts in
  'savedsearches.conf'.
* Default: true

pdfview = <string>
* The name of the view to send as a PDF file.

reportPaperSize = [letter|legal|ledger|a2|a3|a4|a5]
* Default paper size for PDFs.
* Accepted values: letter, legal, ledger, a2, a3, a4, a5
* Default: letter

reportPaperOrientation = [portrait|landscape]
* The orientation of the paper.
* Default: portrait

reportIncludeSplunkLogo = <boolean>
* Whether or not to include a Splunk logo in Integrated PDF Rendering.
* Default: 1 (true)

reportCIDFontList = <string>
* Specify the set (and load order) of CID fonts for handling
  Simplified Chinese(gb), Traditional Chinese(cns),
  Japanese(jp), and Korean(kor) in Integrated PDF Rendering.
* Specify in a space-separated list.
* If multiple fonts provide a glyph for a given character code, the glyph
  from the first font specified in the list is used.
* To skip loading any CID fonts, specify the empty string.
* Default: gb cns jp kor

reportFileName = <string>
* Specify the name of the attached PDF or CSV file.
* Default: $name$-$time:%Y-%m-%d$

width_sort_columns = <boolean>
* Whether or not columns should be sorted from least wide
  to most wide, left to right.
* Valid only if "format=text".
* Default: true

preprocess_results = <search-string>
```

```
* Supply a search string to preprocess results before emailing the results.
  Usually the preprocessing consists of filtering out unwanted internal fields.
* Default: an empty string (no preprocessing)

pdf.footer_enabled = [1 or 0]
  * Set whether or not to display a footer in the PDF.
  * Default: 1 (true)

pdf.header_enabled = [1 or 0]
  * Set whether or not to display a header in the PDF.
  * Default: 1 (true)

pdf.logo_path = <string>
* Define the PDF logo using the syntax <app>:<path-to-image>.
* If set, the PDF is rendered with this logo instead of the Splunk logo.
* If not set, the Splunk logo is used by default.
* The logo is read from the
  $SPLUNK_HOME/etc/apps/<app>/appserver/static/<path-to-image>
  path if <app> is provided.
* The current app is used if <app> is not provided.
* Default: the Splunk logo

pdf.header_left = [logo|title|description|timestamp|pagination|none]
* Set which element is displayed on the left side of header.
* Nothing is displayed if this option is not set, or set to "none".
* Default: none

pdf.header_center = [logo|title|description|timestamp|pagination|none]
* Set which element is displayed on the center of header.
* Nothing is displayed if this option is not set, or set to "none".
* Default: description

pdf.header_right = [logo|title|description|timestamp|pagination|none]
* Set which element is displayed on the right side of header.
* Nothing is displayed if this setting is not set, or set to "none".
* Default: none

pdf.footer_left = [logo|title|description|timestamp|pagination|none]
* Set which element is displayed on the left side of footer.
* Nothing is displayed if this setting is not set, or set to "none".
* Default: logo

pdf.footer_center = [logo|title|description|timestamp|pagination|none]
* Set which element is displayed on the center of footer.
* Nothing is displayed if this setting is not set, or set to "none".
* Default: title

pdf.footer_right = [logo|title|description|timestamp|pagination|none]
* Set which element is displayed on the right side of footer.
* Nothing is displayed if this setting is not set, or set to "none".
* Default: timestamp,pagination

pdf.html_image_rendering = <boolean>
* Whether or not images in HTML should be rendered in the PDF file.
* If rendering images in HTML breaks the PDF for whatever reason,
  change this setting to "false". The old HTML rendering is used.
* Default: true

sslVersions = <string>
* Comma-separated list of SSL versions to support.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
```

```
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Used exclusively for the email alert action and the sendemail search command.
* The default can vary. See the 'sslVersions' setting in the
  $SPLUNK_HOME/etc/system/default/alert_actions.conf file for the current default.

sslVerifyServerCert = <boolean>
* If set to "true", make sure that the server that is being connected to is
  a valid server (authenticated). Both the common name and the alternate
  name of the server are then checked for a match if they are specified in this
  configuration file. A certificate is considered verified if either is matched.
* If set to "true", make sure 'server.conf/[sslConfig]/sslRootCAPath'
  has been set correctly.
* Used exclusively for the email alert action and the sendemail search command.
* Default: false

sslVerifyServerName = <boolean>
* Whether or not splunkd, as a client, performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

sslCommonNameToCheck = <commonName1>, <commonName2>, ...
* Optional.
* Check the common name of the server's certificate against this list of names.
* 'sslVerifyServerCert' must be set to "true" for this setting to work.
* Used exclusively for the email alert action and the sendemail search command.
* Default: no common name checking is performed

sslAltNameToCheck =  <alternateName1>, <alternateName2>, ...
* Optional.
* Check the alternate name of the server's certificate against this list of names.
* If there is no match, assume that Splunk is not authenticated against this
  server.
* 'sslVerifyServerCert' must be set to "true" for this setting to work.
* Used exclusively for the email alert action and the sendemail search command.
* Default: no alternate name checking is performed

cipherSuite = <cipher suite string>
```

```
* If set, the specified cipher string is used for the communication with
  with the SMTP server.
* Used exclusively for the email alert action and the sendemail search command.
* The default can vary. See the 'cipherSuite' setting in the
* $SPLUNK_HOME/etc/system/default/alert_actions.conf file for the current default.
```

RSS: these settings are prefaced by the [rss] stanza

```
[rss]
* Set RSS notification options under this stanza name.
* Follow this stanza name with any number of the following setting/value pairs.
* If you do not specify an entry for each setting, the default value is used.

items_count = <number>
* The number of saved RSS feeds.
* Cannot be more than 'maxresults' (in the global settings).
* Default: 30
```

script: Used to configure any scripts that the alert triggers.

```
[script]
filename = <string>
* The filename, with no path, of the script to trigger.
* The script should be located in: $SPLUNK_HOME/bin/scripts/
* For system shell scripts on UNIX, or .bat or .cmd on Windows, there
  are no further requirements.
* For other types of scripts, the first line should begin with a '#!' marker,
  followed by a path to the interpreter that runs the script.
  * Example: #!C:\Python27\python.exe
* Default: an empty string

############################################################################
# lookup: These settings are prefaced by the [lookup] stanza. They enable the
          Splunk software to write scheduled search results to a new or existing
          CSV lookup file.
############################################################################
[lookup]
filename = <string>
* The filename, with no path, of the CSV lookup file. Filename must end with
  ".csv".
* If this file does not yet exist, Splunk software creates the file on
  the next scheduled run of the search. If the file currently exists, the
  file is overwritten on each run of the search unless "append=1".
* The file is placed in the same path as other CSV lookup files:
  $SPLUNK_HOME/etc/apps/search/lookups.
* Default: an empty string

append = <boolean>
* Whether or not to append results to the lookup file defined for the
  'filename' setting.
* Default: 0 (false)
```

summary_index: these settings are prefaced by the [summary_index] stanza

```
[summary_index]
inline = <boolean>
* Whether or not the summary index search command is run as part of the
  scheduled search or as a follow-on action. When the results of the scheduled
  search are expected to be large, keep the default setting "inline=true".
* Default: 1 (true)

_name = <string>
* The name of the summary index where the events are written to.
* Default: summary
```

summary_metric_index: these settings are prefaced by the [summary_metric_index] stanza

```
[summary_metric_index]
inline = <boolean>
* Whether or not the summary index search command is run as part of the
  scheduled search or as a follow-on action. When the results of the scheduled
  search are expected to be large, keep the default setting "inline=true".
* Default: 1 (true)

_name = <string>
* The name of the summary index where the events are written to.
* Default: summary
```

populate_lookup: these settings are prefaced by the [populate_lookup] stanza

```
[populate_lookup]
* This alert action is deprecated and will be disabled
  in a future release. Use the 'lookup' alert action instead.
dest = <string>
* Name of the lookup table to populate (stanza name in the transforms.conf file),
  or the lookup file path where you want the data written to. If a path is
  specified it MUST be relative to $SPLUNK_HOME and a valid lookups
  directory.
  For example: "etc/system/lookups/<file-name>" or
  "etc/apps/<app>/lookups/<file-name>"
* The user executing this action MUST have write permissions to the app for
  this action to work properly.

[<custom_alert_action>]
```

alert_actions.conf.example

```
#   Version 9.2.2
#
# This is an example alert_actions.conf.  Use this file to configure alert
# actions for saved searches.
#
# To use one or more of these configurations, copy the configuration block into
# alert_actions.conf in $SPLUNK_HOME/etc/system/local/.  You must restart
# Splunk to enable configurations.
```

```
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[email]
# keep the search artifacts around for 24 hours
ttl = 86400

# if no @ is found in the address the hostname of the current machine is appended
from = splunk

format = table

inline = false

sendresults = true

hostname = CanAccessFromTheWorld.com

command = sendemail "to=$action.email.to$" "server=$action.email.mailserver{default=localhost}$"
"from=$action.email.from{default=splunk@localhost}$" "subject=$action.email.subject{recurse=yes}$"
"format=$action.email.format{default=csv}$" "sssummary=Saved Search [$name$]: $counttype$($results.count$)"
"sslink=$results.url$" "ssquery=$search$" "ssname=$name$" "inline=$action.email.inline{default=False}$"
"sendresults=$action.email.sendresults{default=False}$" "sendpdf=$action.email.sendpdf{default=False}$"
"pdfview=$action.email.pdfview$" "searchid=$search_id$" "graceful=$graceful{default=True}$"
maxinputs="$maxinputs{default=1000}$" maxtime="$action.email.maxtime{default=5m}$"

use_tls = 1
sslVersions = tls1.2
sslVerifyServerCert = true
sslCommonNameToCheck = host1, host2

[rss]
# at most 30 items in the feed
items_count=30

# keep the search artifacts around for 24 hours
ttl = 86400

command = createrss "path=$name$.xml" "name=$name$" "link=$results.url$" "descr=Alert trigger: $name$,
results.count=$results.count$ " "count=30" "graceful=$graceful{default=1}$"
maxtime="$action.rss.maxtime{default=1m}$"

[summary_index]
# don't need the artifacts anytime after they're in the summary index
ttl = 120

# make sure the following keys are not added to marker (command, ttl, maxresults, _*)
command = summaryindex addtime=true index="$action.summary_index._name{required=yes}$"
file="$name$_$#random$.stash" name="$name$" marker="$action.summary_index*{format=$KEY=\\\"$VAL\\\",
key_regex="action.summary_index.(?!(?:command|maxresults|ttl|(?:_.*))$)(.*)"}$"

[summary_metric_index]
# don't need the artifacts anytime after they're in the summary index
ttl = 120

# make sure that "mcollect" is the SPL command and has the option "split=allnums"
command = mcollect index="$action.summary_index._name{required=yes}$" file="$name_hash$_$#random$.stash"
name="$name$" marker="$action.summary_index*{format=$KEY=\\\"$VAL\\\",
key_regex="action.summary_index.(?!(?:command|forceCsvResults|inline|maxresults|maxtime|python\\.version|ttl|track
_alert|(?:_.*))$)(.*)"}$" split=allnums $action.summary_index._metric_dims$
```

```
[custom_action]
# flag the action as custom alert action
is_custom = 1

# configure appearance in the UI
label = Custom Alert Action
description = Triggers a custom alert action
icon_path = custom_alert.png

# override default script execution
# java.path is a path pointer file in <app>/bin pointing to the actual java executable
alert.execute.cmd = java.path
alert.execute.cmd.arg.1 = -jar
alert.execute.cmd.arg.2 = $SPLUNK_HOME/etc/apps/myapp/bin/custom.jar
alert.execute.cmd.arg.3 = --execute
```

app.conf

The following are the spec and example files for `app.conf`.

app.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file maintains the state of a given app in the Splunk platform. It can
# also be used to customize certain aspects of an app.
#
# An app.conf file can exist within each app on the Splunk platform.
#
# You must restart the Splunk platform to reload manual changes to app.conf.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[author=<name>]

```
email = <email-address>
company = <company-name>
```

[id]

```
group = <group-name>
name = <app-name>
version = <version-number>
```

[launcher]

```
* Settings in this stanza determine how an app appears in the Launcher in the Splunk
  platform and online on Splunkbase.

# Global Settings:

remote_tab = <boolean>
* Determines whether the Launcher interface connects to apps.splunk.com
  (Splunkbase).
* This setting only applies to the Launcher app. Do not set it in any
  other app.
* Default: true

# Per-application Settings:

version = <string>
* Version numbers are a number followed by a sequence of dots and numbers.
* The best practice for version numbers for releases is to use three digits
  formatted as Major.Minor.Revision.
* Pre-release versions can append a single-word suffix like "beta" or
  "preview".
* Use lower case and no spaces when you designate a pre-release version.
* Example versions:
  * 1.2.0
  * 3.2.1
  * 11.0.34
  * 2.0beta
  * 1.3beta2
  * 1.0preview

description = <string>
* A short explanatory string that appears below the title of the app in
  Launcher.
* Limit descriptions to 200 characters or less for user readability.

author = <string>
* For apps that you intend to upload to Splunkbase, list the username of your
  splunk.com account.
* For apps that are for internal use only, include your full name and/or contact
  info, such as your email address.

# Your app can include an icon which appears next to your app in Launcher
# and on Splunkbase. You can also include a screenshot, which shows up on
# Splunkbase when the user views information about your app before downloading it.
# If you include an icon file, the file name must end with "Icon" before the
# file extension and the "I" must be capitalized. For example, "mynewIcon.png".
# Screenshots are optional.
#
# There is no setting in app.conf for screenshot or icon images.
# Splunk Web places files you upload with your app into
# the $SPLUNK_HOME/etc/apps/<app_name>/static/ directory.
# These images do not appear in your app.
#
# Move or place icon images in the $SPLUNK_HOME/etc/apps/<app_name>/static/ directory.
# Move or place screenshot images in the $SPLUNK_HOME/etc/apps/<app_name>/static/ directory.
# Launcher and Splunkbase automatically detect the images in those locations.
#
# For example:
#
#     <app_name>/static/appIcon.png    (the capital "I" is required!)
```

```
#     <app_name>/static/screenshot.png
#
# An icon image must be a 36px by 36px PNG file.
# An app screenshot must be a 623px by 350px PNG file.
```

[package]

```
* This stanza defines upgrade-related metadata that streamlines app upgrade
  to future versions of Splunk Enterprise.

id = <string>
* Omit this setting for apps that are for internal use only and not intended
  for upload to Splunkbase.
* id is required for all new apps that you upload to Splunkbase. Future versions of
  Splunk Enterprise will use appid to correlate locally-installed apps and the
  same app on Splunkbase (e.g. to notify users about app updates).
* id must be the same as the folder name in which your app lives in
  $SPLUNK_HOME/etc/apps.
* id must adhere to these cross-platform folder name restrictions:
  * must contain only letters, numbers, "." (dot), and "_" (underscore)
    characters.
  * must not end with a dot character.
  * must not be any of the following names: CON, PRN, AUX, NUL,
      COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9,
      LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

check_for_updates = <boolean>
* Determines whether Splunk Enterprise checks Splunkbase for updates to this
  app.
* Default: true

show_upgrade_notification = <boolean>
* Determines whether Splunk Enterprise shows an upgrade notification in Splunk
  Web for this app.
* Default: false
```

[install]

```
* This stanza defines install settings for this app.

state = disabled | enabled
* Determines whether an app is disabled or enabled on the Splunk platform.
* If an app is disabled, its configurations are ignored.
* Default: enabled

state_change_requires_restart = <boolean>
* Determines whether changing an app's state ALWAYS requires a restart of Splunk
  Enterprise.
* State changes include enabling or disabling an app.
* When set to true, changing an app's state always requires a restart.
* When set to false, modifying an app's state might or might not require a
  restart, depending on what the app contains. This setting cannot be used to
  avoid all restart requirements.
* Default: false

is_configured = <boolean>
* Stores an indication of whether the application's custom setup has been
  performed.
* Default: false
```

```
build = <integer>
* Required.
* Must be a positive integer.
* Increment this whenever you change files in <app_name>/static.
* Every release must change both 'version' and 'build' settings.
* Ensures browsers don't use cached copies of old static files
  in new versions of your app.
* 'build' is a single integer, unlike 'version' which can be a complex string,
  such as 1.5.18.

allows_disable = <boolean>
* Determines whether an app allows itself to be disabled.
* Default: true

install_source_checksum = <string>
* Records a checksum of the tarball from which a given app was installed.
* Splunk Enterprise automatically populates this value upon install.
* Do not set this value explicitly within your app!

install_source_local_checksum = <string>
* Records a checksum of the tarball from which a given app's local configuration
  was installed.
* Splunk Enterprise automatically populates this value upon install.
* Do not set this value explicitly within your app!

python.version = {default|python|python2|python3}
* When 'installit.py' exists, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.
```

[triggers]

```
* This stanza controls reloading of custom configuration files included in
  the app (4.2+ versions only).
* Include this stanza if your app includes custom configuration files.

# Conf-level reload triggers
reload.<conf_file_name> = [ simple | never | rest_endpoints | access_endpoints <handler_url> | http_get
<handler_url> | http_post <handler_url> ]
* Splunk Enterprise reloads app configuration after every app-state change:
  install, update, enable, and disable.
* If your app does not use a custom config file (e.g.myconffile.conf)
  then it does not require a [triggers] stanza. This is because
  $SPLUNK_HOME/etc/system/default/app.conf includes a [triggers]
  stanza, which automatically reloads config files used by Splunk Enterprise.
* If your app uses a custom config file (e.g. myconffile.conf) and you want to
  avoid unnecessary Splunk Enterprise restarts, you can add a reload value in
  the [triggers] stanza.
* If you do not include [triggers] settings and your app uses a custom config
  file, Splunk Enterprise requires a restart after every state change.
* If set to "simple", Splunk Enterprise takes no special action
  to reload your custom configuration file.
* If you specify "access_endpoints" with a URL to a REST endpoint, Splunk
  Enterprise calls its _reload() method at every app state change.
* If you specify "http_get" with a URL to a REST endpoint, Splunk Enterprise
  simulates an HTTP GET request against the URL at every app state change.
* If you specify "http_post" with a URL to a REST endpoint, Splunk Enterprise
  simulates an HTTP POST request against the URL at every app state change.
```

```
* If set to "never", Splunk Enterprise initiates a restart after any state change.
* "rest_endpoints" is reserved for Splunk Enterprise internal use for reloading
  restmap.conf.
* NOTE: The "conf_file_name" value does not include the file extension ".conf".

# Stanza-level reload triggers
reload.<conf_file_name>.<conf_stanza_prefix> = [ simple | never | access_endpoints <handler_url> | http_get
<handler_url> | http_post <handler_url> ]
* Stanza-level reload triggers for indexer-cluster peers to reload only the
  config file stanzas that are changed in the newly pushed cluster bundle.
* With the stanza level reload triggers, we can have more granular control over
  which subset of existing reload handlers to invoke depending on which stanzas
  of a given config file have changed in the newly pushed cluster bundle. See
  example below for more information.
* Stanza level reload trigger values operate identically to conf-level reload
  trigger values, i.e. "simple", "never","access_endpoints", "http_get", "http_post".
* For any stanza of <conf_file_name> that does NOT have a corresponding stanza-level
  reload trigger listed under the [triggers] section of app.conf, the cluster peer
  will fallback to the "rolling restart behavior" upon detecting changes of those
  "missing" stanzas in the newly pushed cluster bundle.
* NOTE: This setting is ONLY used by indexer-cluster peers and ONLY supported
  by inputs.conf and server.conf.
* NOTE: The "conf_file_name" value does not include the file extension ".conf".
```

[shclustering]

```
deployer_lookups_push_mode = preserve_lookups | always_preserve | always_overwrite | overwrite_on_change
* Determines the deployer_lookups_push_mode for the 'splunk apply
  shcluster-bundle' command.
* If set to "preserve_lookups", the 'splunk apply shcluster-bundle' command
  honors the '-preserve-lookups' option as it appears on the command line. If
  '-preserve-lookups' is flagged as "true", then lookup tables for this app are
  preserved. Otherwise, lookup tables are overwritten.
* If set to "always_preserve", the 'splunk apply shcluster-bundle' command ignores
  the '-preserve-lookups' option as it appears on the command line and lookup
  tables for this app are always preserved.
* If set to "always_overwrite", the 'splunk apply shcluster-bundle' command
  ignores the '-preserve-lookups' option as it appears on the command line and
  lookup tables for this app are always overwritten.
* If set to "overwrite_on_change", the 'splunk apply shcluster-bundle' command
  ignores the '-preserve-lookups' option as it appears on the command line and
  lookup tables for this app are overwritten if the app contents have changed.
* Default: always_preserve

deployer_push_mode = full | merge_to_default | local_only | default_only
* How the deployer pushes the configuration bundle to search head cluster
  members.
* If set to "full": Bundles all of the app's contents located in default/,
  local/, users/<app>/, and other app subdirs. It then pushes the bundle to
  the members. When applying the bundle on a member, the non-local and
  non-user configurations from the deployer's app folder are copied to the
  member's app folder, overwriting existing contents. Local and user
  configurations are merged with the corresponding folders on the member,
  such that member configuration takes precedence.  This option should not
  be used for built-in apps, as overwriting the member's built-in apps can
  result in adverse behavior.
* If set to "merge_to_default": Merges the local and default folders into
  the default folder and pushes the merged app to the members. When
  applying the bundle on a member, the default configuration on the member
  is overwritten. User configurations are copied and merged with the user
  folder on the member, such that the existing configuration on the member
```

```
  takes precedence. In versions 7.2 and prior, this was the only behavior.
* If set to "local_only": This option bundles the app's local directory (and its
  metadata) and pushes it to the cluster. When applying the bundle to a
  member, the local configuration from the deployer is merged with the
  local configuration on the member, such that the member's existing
  configuration takes precedence. Use this option to push the local
  configuration of built-in apps, such as search. If used to push an app
  that relies on non-local content (such as default/ or bin/), these
  contents must already exist on the member.
* If set to "default_only": Bundles all of the configuration files except
  for local and users/<app>/.  When applying the bundle on a member, the
  contents in the member's default folder are overwritten.
* Default (all apps except built-in apps): "merge_to_default"
* Default (built-in apps): "local_only"

#
# Set UI-specific settings for this app
#
```

[ui]

```
* This stanza defines UI-specific settings for this app.

is_visible = <boolean>
* Indicates if this app is visible/navigable as an app in Splunk Web.
* Apps require at least one view to be available in Splunk Web.

show_in_nav = <boolean>
* Determines whether this app appears in the global app dropdown.

is_manageable = <boolean>
* Support for this setting has been removed. It no longer has any effect.

label = <string>
* Defines the name of the app shown in Splunk Web and Launcher.
* Recommended length between 5 and 80 characters.
* Must not include "Splunk For" prefix.
* Label is required.
* Examples of good labels:
    IMAP Monitor
    SQL Server Integration Services
    FISMA Compliance

docs_section_override = <string>
* Defines override for auto-generated app-specific documentation links.
* If not specified, app-specific documentation link includes
  [<app-name>:<app-version>].
* If specified, app-specific documentation link includes
  [<docs_section_override>].
* This setting only applies to apps with documentation on the Splunk
  documentation site.

attribution_link = <string>
* URL that users can visit to find third-party software credits and attributions
  for assets the app uses.
* External links must start with http:// or https://.
* Values that do not start with http:// or https:// get interpreted as Quickdraw
  location strings and translated to internal documentation references.

setup_view = <string>
* Optional.
```

```
* Defines custom setup view found within the /data/ui/views REST endpoint.

supported_themes = <comma-separated list>
* A comma-separated list of themes supported by the app.
* Supported values are "dark" and "light".
* This setting is optional.
* If you specify this setting, you must give it a value of "light".
* No default.
```

[credentials_settings]

```
* This stanza controls credential-verification scripting (4.2+ versions only).
* Credential entries are superseded by passwords.conf from 6.3 onwards.
* While the entries here are still honored post-6.3, updates to these occur in
  passwords.conf, which overrides any values present here.

verify_script = <string>
* Optional setting.
* Command line to invoke to verify credentials used for this app.
* For scripts, the command line must include both the interpreter and the
  script for it to run.
    * Example: "$SPLUNK_HOME/bin/python" "$SPLUNK_HOME/etc/apps/<myapp>/bin/$MY_SCRIPT"
* The invoked program is communicated with over standard in / standard out via
  the same protocol as splunk scripted auth.
* Paths incorporating variable expansion or explicit spaces must be quoted.
  * For example, a path including $SPLUNK_HOME should be quoted, as likely
    will expand to C:\Program Files\Splunk

python.version = {default|python|python2|python3}
* This property is used only when verify_script begins with the canonical path
  to the Python interpreter, in other words, $SPLUNK_HOME/bin/python.  If any
  other path is used, this property is ignored.
* For Python scripts only, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.
```

[credential:<realm>:<username>]

```
password = <string>
* Password that corresponds to the given username for the given realm.
* Realm is optional.
* The password can be in clear text, but when saved from splunkd the
  password is always encrypted.
```

[diag]

```
* This stanza applies to diag app extensions, 6.4+ only.

extension_script = <filename>
* Setting this variable declares that this app puts additional information
  into the troubleshooting & support oriented output of the 'splunk diag'
  command.
* Must be a python script.
* Must be a simple filename, with no directory separators.
* The script must exist in the 'bin' subdirectory in the app.
```

```
* Full discussion of the interface is located on the Splunk developer portal.
  See http://dev.splunk.com/view/SP-CAAAE8H
* Default: not set (no app-specific data collection will occur).

data_limit = <positive integer>[b|kb|MB|GB]
* Defines a soft ceiling for the amount of uncompressed data that can be
  added to the diag by the app extension.
* Large diags damage the main functionality of the tool by creating data blobs
  too large to copy around or upload.
* Use this setting to ensure that your extension script does not accidentally
  produce far too much data.
* After data produced by this app extension reaches the limit, diag does not add
  any further files on behalf of the extension.
* After diag has finished adding a file which goes over this limit, all further files
  are not be added.
* Must be a positive number followed by a size suffix.
  * Valid suffixes: b: bytes, kb: kilobytes, mb: megabytes, gb: gigabytes
  * Suffixes are case insensitive.
* Default: 100MB

# Other diag settings

default_gather_lookups = <filename> [, <filename> ...]
* Set this variable to declare that the app contains lookups that diag must
  always gather by default.
* Essentially, if there are lookups which are useful for troubleshooting an
  app, and will never contain sensitive (user) data, add the lookups to this
  list so that they appear in generated diags for use when troubleshooting
  the app from customer diags.
* Any files in lookup directories that are not listed here are not gathered by
  default. You can override this behavior with the diag flag --include-lookups.
* This setting is new in Splunk Enterprise/Light version 6.5. Older versions
  gather all lookups by default.
* This does not override the size-ceiling on files in etc. Large lookups are
  still excluded unless the etc-filesize-limit is raised or disabled.
* This only controls files in the same app directory as this conf file.  For
  example, if you have an app directory in etc/peer-apps (index clustering),
  this setting must appear in etc/peer-apps/appname/default/app.conf or
  local/app.conf
* Additional lists can be created with default_gather_lookups-classname = ...
* Default: not set
```

app.conf.example

```
#   Version 9.2.2
#
# The following are example app.conf configurations. Configure properties for
# your custom application.
#
# There is NO DEFAULT app.conf.
#
# To use one or more of these configurations, copy the configuration block into
# app.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

```
[launcher]
author=<author of app>
description=<textual description of app>
version=<version of app>

[triggers]
########## Conf-level reload triggers ##########
# Do not force a restart of Splunk Enterprise for state changes of MyApp
# Do not run special code to tell MyApp to reload myconffile.conf
# Apps with custom config files usually pick this option:
reload.myconffile = simple

# Do not force a restart of Splunk Enterprise for state changes of MyApp.
# Splunk Enterprise calls the /admin/myendpoint/_reload method in my custom
# EAI handler.
# Use this advanced option only if MyApp requires custom code to reload
# its configuration when its state changes
reload.myotherconffile = access_endpoints /admin/myendpoint

########## Stanza-level reload triggers ##########
# For any changed inputs.conf stanzas in the newly pushed cluster
# bundle that start with the "monitor" stanza prefix, e.g.
# [monitor://*], invoke the corresponding monitor input reload handler
# as specified, i.e. /data/inputs/monitor/_reload
#
# NOTE: The scripted input reload handler and the http input reload
# will NOT be invoked if the only changed inputs stanzas in the
# newly pushed cluster bundle are monitor inputs.
reload.inputs.monitor = access_endpoints /data/inputs/monitor
reload.inputs.script  = access_endpoints /data/inputs/script
reload.inputs.http    = access_endpoints /data/inputs/http
```

audit.conf

The following are the spec and example files for `audit.conf`.

audit.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attributes and values you can use to configure
# auditing in audit.conf.
#
# There is NO DEFAULT audit.conf. To set custom configurations, place an
# audit.conf in $SPLUNK_HOME/etc/system/local/. For examples, see
# audit.conf.example.  You must restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top of the file.
#  * Each conf file should have at most one default stanza. If there are
```

```
#    multiple default stanzas, attributes are combined. In the case of multiple
#    definitions of the same attribute, the last definition in the file wins.
#  * If an attribute is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.
```

[auditTrail]

```
queueing = <boolean>
* Whether or not audit events are sent to the indexQueue.
* If set to "true", audit events are sent to the indexQueue.
* If set to "false", you must add an inputs.conf stanza to tail the
  audit log for the events reach your index.
* Default: true
```

audit.conf.example

```
#   Version 9.2.2
#
# This is an example audit.conf.  Use this file to configure auditing.
#
# There is NO DEFAULT audit.conf.
#
# To use one or more of these configurations, copy the configuration block into
# audit.conf in $SPLUNK_HOME/etc/system/local/.  You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

authentication.conf

The following are the spec and example files for `authentication.conf`.

authentication.conf.spec

```
#   Version 9.2.2
#
# This file contains possible settings and values for configuring
# authentication via authentication.conf.
#
# There is an authentication.conf file in $SPLUNK_HOME/etc/system/default/. To
# set custom configurations, place an authentication.conf in
# $SPLUNK_HOME/etc/system/local/. For examples, see
# authentication.conf.example. You must restart the Splunk platform to enable
# configurations.
#
# To learn more about configuration files, including precedence, see
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles.
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top
#     of the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file wins.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

[authentication]
* Follow this stanza name with any number of the following setting/value
  pairs.

authType = [Splunk|LDAP|Scripted|SAML|ProxySSO]
* Specify which authentication system to use.
* Supported values: Splunk, LDAP, Scripted, SAML, ProxySSO.
* Default: Splunk

authSettings = <authSettings-key>,<authSettings-key>,...
* Key to look up the specific configurations of chosen authentication
  system.
* <authSettings-key> is the name of a stanza header that specifies
  settings for scripted authentication, SAML, ProxySSO and for an LDAP
  strategy. Those stanzas are defined below.
* For LDAP, specify the LDAP strategy name(s) here. If you want Splunk
  software to query multiple LDAP servers, provide a comma-separated list
  of all strategies. Each strategy must be defined in its own stanza.
  The order in which you specify the strategy names is the order Splunk
  software uses to query their servers when looking for a user.
* For scripted authentication, <authSettings-key> should be a single
  stanza name.

passwordHashAlgorithm =
[SHA512-crypt|SHA256-crypt|SHA512-crypt-<num_rounds>|SHA256-crypt-<num_rounds>|MD5-crypt]
* This controls how hashed passwords are stored in the
  $SPLUNK_HOME/etc/passwd file for the default "Splunk" authType.
* "MD5-crypt" is an algorithm originally developed for FreeBSD in the early
  1990s, which became a widely used standard among UNIX machines. Splunk
  Enterprise also used it through the 5.0.x releases. MD5-crypt runs the
  salted password through a sequence of 1000 MD5 operations.
* "SHA256-crypt" and "SHA512-crypt" are newer versions that use 5000 rounds
  of the Secure Hash Algorithm-256 (SHA256) or SHA512 hash functions.
  This is slower than MD5-crypt and therefore more resistant to dictionary
  attacks.  SHA512-crypt is used for system passwords on many versions of Linux.
* These SHA-based algorithm can optionally be followed by a number of rounds
  to use. For example, "SHA512-crypt-10000" uses twice as many rounds
  of hashing as the default implementation. The number of rounds must be at
  least 1000.
  If you specify a very large number of rounds (i.e. more than 20x the
  default value of 5000), splunkd might become unresponsive and connections to
  splunkd (from Splunk Web or CLI) time out.
* This setting only affects new password settings (either when a user is
  added or a user's password is changed).  Existing passwords work but retain their
  previous hashing algorithm.
* Default: SHA512-crypt

defaultRoleIfMissing = <splunk role>
```

```
* Applicable for LDAP authType. If the LDAP server does not return any groups, or if
  groups cannot be mapped to Splunk roles, then this value is used, if provided.
* This setting is optional.
* Default: empty string

externalTwoFactorAuthVendor = <string>
* A valid multifactor vendor string enables multifactor authentication
  and loads support for the corresponding vendor if supported by the the Splunk platform.
* An empty string disables multifactor authentication in the the Splunk platform.
* Currently Splunk supports Duo and RSA as multifactor authentication vendors.
* This setting is optional.
* No default.

externalTwoFactorAuthSettings = <externalTwoFactorAuthSettings-key>
* Key to look up the specific configuration of chosen multifactor
  authentication vendor.
* This setting is optional.
* No default.
```

LDAP settings

```
[<authSettings-key>]
* Follow this stanza name with the following setting/value pairs.
* For multiple strategies, specify multiple instances of
  this stanza, each with its own stanza name and a separate set of
  settings.
* The <authSettings-key> must be one of the values listed in the
  authSettings setting, which must be specified in the previous [authentication]
  stanza.

host = <string>
* The hostname of the LDAP server.
* Confirm that your Splunk server can resolve the host name through DNS.
* Required.
* No default.

SSLEnabled = [0|1]
* Specifies whether SSL is enabled.
* See the file $SPLUNK_HOME/etc/openldap/ldap.conf for SSL LDAP settings
* This setting is optional.
* Default: 0 (disabled)

port = <integer>
* The port that the Splunk platform should use to connect to your LDAP server.
* This setting is optional.
* Default (non-SSL): 389
* Default (SSL): 636

bindDN = <string>
* The LDAP Distinguished Name of the user that retrieves the LDAP entries.
* This user must have read access to all LDAP users and groups you wish to
  use in the auth system.
* This setting is optional.
* Leave this setting blank to retrieve your LDAP entries using
  anonymous bind (which must be supported by the LDAP server)
* No default.

bindDNpassword = <password>
```

```
* Password for the bindDN user.
* This setting is optional.
* Leave this blank if anonymous bind is sufficient.
* No default.

userBaseDN = <string>
* The distinguished names of LDAP entries whose subtrees contain the users.
* Enter a ';' delimited list to search multiple trees.
* Required.
* No default.

userBaseFilter = <string>
* The LDAP search filter to use when searching for users.
* Highly recommended, especially when there are many entries in your LDAP
  user subtrees.
* When used properly, search filters can significantly speed up LDAP queries
* Here is an example that matches users in the IT or HR department:
    * userBaseFilter = (|(department=IT)(department=HR))
    * See RFC 2254 for more detailed information on search filter syntax
* This setting is optional.
* Default: empty string (no filtering)

userNameAttribute = <string>
* This is the username.
* NOTE: This setting should use case insensitive matching for its values,
  and the values should not contain whitespace
    * Usernames are case insensitive in the the Splunk platform
* In Active Directory, this is 'sAMAccountName'
* Required.
* A typical value is 'uid'.
* No default.

realNameAttribute = <string>
* The user's real, human readable name.
* Required.
* A typical value is 'cn'.
* No default.

emailAttribute = <string>
* The user's email address.
* This setting is optional.
* Default: mail

groupMappingAttribute = <string>
* The value that group entries use to declare membership.
* Groups are often mapped with user DN, so this defaults to 'dn'
* Set this if groups are mapped using a different setting
  * Usually only needed for OpenLDAP servers.
  * A typical setting is 'uid'
    * For example, assume a group declares that one of its members is
      'splunkuser' — every user with the 'uid' value 'splunkuser' is
      mapped to that group.
* This setting is optional.
* No default.

groupBaseDN = [<string>;<string>;...]
* The LDAP Distinguished Names of LDAP entries whose subtrees contain
  the groups.
* Required.
* Enter a semicolon (;) delimited list to search multiple trees.
* If your LDAP environment does not have group entries, there is a
  configuration that can treat each user as its own group:
```

```
  * Set groupBaseDN to the same as userBaseDN, which means you search
    for groups in the same place as users.
  * Next, set the groupMemberAttribute and groupMappingAttribute to the same
    setting as userNameAttribute.
    * This means the entry, when treated as a group, uses the username
      value as its only member.
  * For clarity, also set groupNameAttribute to the same
    value as userNameAttribute.
* No default.

groupBaseFilter = <string>
* The LDAP search filter the Splunk platform uses when searching for static groups
* Like 'userBaseFilter', this is highly recommended to speed up LDAP queries
* See Request for Comments (RFC) 2254 on the Internet Engineering Task Force
  (IETF) website for more information.
* This setting is optional.
* Default: empty string (no filtering).

dynamicGroupFilter = <string>
* The LDAP search filter the Splunk platform uses when searching for dynamic groups.
* Configure this setting only if you intend to retrieve dynamic groups
  on your LDAP server.
* Example: '(objectclass=groupOfURLs)'
* This setting is optional.
* Default: empty string

dynamicMemberAttribute = <string>
* This setting contains the LDAP URL needed to retrieve members dynamically.
* Only configure this if you intend to retrieve dynamic groups on your
  LDAP server.
* This setting is required if you want to retrieve dynamic groups.
* Otherwise, it is optional.
* Example: 'memberURL'
* No default.

groupNameAttribute = <string>
* This is the group entry setting whose value stores the group name.
* A typical setting for this is 'cn' (common name)
* Recall that if you are configuring LDAP to treat user entries as their own
  group, user entries must have this setting
* Required.
* Default: empty string

groupMemberAttribute = <string>
* This is the group entry setting whose values are the groups members
* Typical setting for this are 'member' and 'memberUid'
* For example, consider the groupMappingAttribute example above using
  groupMemberAttribute 'member'
  * To declare 'splunkuser' as a group member, its setting 'member' must
    have the value 'splunkuser'
* Required.
* Default: empty string

nestedGroups = <boolean>
* Controls whether the Splunk platform expands nested groups using the
  'memberof' extension.
* Set to 1 if you have nested groups you want to expand and the 'memberof'
  extension on your LDAP server.
* This setting is optional.

charset = <string>
* Only set this for an LDAP setup that returns non-UTF-8 encoded data. LDAP
```

```
  is supposed to always return UTF-8 encoded data (See RFC 2251), but some
  tools incorrectly return other encodings.
* Follows the same format as 'CHARSET' in props.conf (see props.conf.spec)
* An example value would be "latin-1"
* This setting is optional.
* Default: empty string

anonymous_referrals = [0|1]
* Set this to 0 to turn off referral chasing
* Set this to 1 to turn on anonymous referral chasing
* NOTE: the Splunk platform only chases referrals using anonymous bind.
        It does not support rebinding using credentials.
* If you do not need referral support, set this to 0.
* If you wish to make referrals work, set this to 1 and confirm your server
  allows anonymous searching
* This setting is optional.
* Default: 1

sizelimit = <integer>
* Limits the amount of entries that the Splunk platform requests in LDAP search.
* NOTE: The max entries returned is still subject to the maximum
        imposed by your LDAP server.
  * Example: If you set this to 5000 and the server limits it to 1000,
             the software only returns 1000 entries.
* This setting is optional.
* Default: 1000

pagelimit = <integer>
* The maximum number of entries to return in each page.
* Enables result sets that exceed the maximum number of entries defined for the
  LDAP server.
* If set to -1, ldap pagination is off.
* IMPORTANT: The maximum number of entries a page returns is subject to
  the maximum page size limit of the LDAP server. For example: If you set 'pagelimit =
  5000' and the server limit is 1000, you cannot receive more than 1000 entries in
  a page.
* This setting is optional.
* Default: -1

enableRangeRetrieval = <boolean>
* The maximum number of values that can be retrieved from one attribute in a
  single LDAP search request is determined by the LDAP server. If the number of
  users in a group exceeds the LDAP server limit, enabling this setting fetches all
  users by using the "range retrieval" mechanism.
* Enables result sets for a given attribute that exceed the maximum number of
  values defined for the LDAP server.
* If set to false, ldap range retrieval is off.
* This setting is optional.
* Default: false

timelimit = <integer>
* Limits the amount of time, in seconds, that the Splunk platform waits for an LDAP search
  request to complete.
* If your searches finish quickly, lower this value from the default.
* Maximum value is 30 seconds
* Default: 15

network_timeout = <integer>
* Limits the amount of time a socket polls a connection without activity
* This is useful for determining if your LDAP server cannot be reached
* NOTE: As a connection could be waiting for search results, this value
        must be higher than 'timelimit'.
```

```
* Like 'timelimit', if you have a fast connection to your LDAP server,
  lower this value.
* Maximum value is -1 (unlimited)
* This setting is optional.
* Default: 20

ldap_negative_cache_timeout = <nonnegative decimal>
* The amount of time, in seconds, that the Splunk platform remembers that a non-existent
  user on an LDAP provider does not exist.
* This setting is useful when you want to avoid frequent LDAP queries for users
  that do not exist on the LDAP provider.
* This setting does not prevent LDAP queries on login. Login always queries the LDAP
  provider to confirm that a user exists.
* Default: 86400
```

Map roles

```
[roleMap_<authSettings-key>]
* The mapping of Splunk roles to LDAP groups for the LDAP strategy specified
  by <authSettings-key>
* Follow this stanza name with several Role-to-Group(s) mappings as defined
  below.
* NOTE: This role mapping ONLY applies to the specified strategy.
* Importing groups for the same user from different strategies is not
  supported.

<Splunk RoleName> = <semicolon-separated list>
* Maps a Splunk role from the authorize.conf configuration file to one or more LDAP groups.
* Separate multiple LDAP groups with semicolons, not spaces.
* List several of these setting/value pairs to map several Splunk roles to
  LDAP Groups.
* LDAP group names are case sensitive.
```

Scripted authentication

```
[<authSettings-key>]
* Follow this stanza name with the following setting/value pairs:

python.version = {default|python|python2|python3}
* For Python scripts only, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.

scriptSearchFilters = [1|0]
* Whether or not to call the script to add search filters.
* Set this to 1 to call the script to add search filters.
* Default: 0

[cacheTiming]
* Use these settings to adjust how long the Splunk platform uses the answers returned
  from script functions before calling them again.
* All timeouts can be expressed in seconds or as a search-like time range
* Examples include "30" (30 seconds), "2mins" (2 minutes), "24h" (24 hours), etc.
```

```
* You can opt to use no caching for a particular function by setting the
  value to "0".
  * Be aware that this can severely hinder performance as a result of heavy
    script invocation.
* Choosing the correct values for cache timing involves a tradeoff between
  new information latency and general performance.
  * High values yield better performance from calling the script less, but
    introduces a latency in picking up changes.
  * Low values pick up changes in your external auth system more
    quickly, but can slow down performance due to increased script
    invocations.

userLoginTTL = <time range string>
* The timeout for the 'userLogin' script function.
* These return values are cached on a per-user basis.
* Default: 0 (no caching)

userInfoTTL = <time range string>
* How long the auth system caches information that it retrieves with the
  'getUserInfo' and 'getUsers' scripts.
* These return values are cached on a per-user basis.
* Default (if you have configured either 'getUserInfoTTL' or 'getUsersTTL'): the larger value of these
settings
* Default (otherwise): 10s

getUserInfoTTL = <time range string>
* DEPRECATED; use 'userInfoTTL' instead.
* How long the auth system caches information that it retrieves with the
  'getUserInfo' script.
* These return values are cached on a per-user basis.
* Default: 10s

getUsersTTL = <time range string>
* DEPRECATED; use 'userInfoTTL' instead.
* The timeout for the getUsers script function.
* There is only one global getUsers cache (it is not tied to a
  specific user).
* Default: 10s
```

Settings for Splunk Authentication mode

```
[splunk_auth]
* Settings for Splunk's internal authentication system.


minPasswordLength = <positive integer>
* Specifies the minimum permitted password length in characters when
  passwords are set or modified.
* Password modification attempts which do not meet this requirement are
* explicitly rejected.
* Values less than 1 are ignored.
* This setting is optional.
* Default: 8

minPasswordUppercase = <positive integer>
* Specifies the minimum permitted uppercase characters when passwords are set
  or modified.
* The Splunk platform ignores negative values.
* This setting is optional.
```

```
* Password modification attempts which do not meet this requirement are
  explicitly rejected.
* Default: 0

minPasswordLowercase = <positive integer>
* Specifies the minimum permitted lowercase characters when passwords are set
  or modified.
* The the Splunk platform ignores negative values.
* This setting is optional.
* Password modification attempts which do not meet this requirement are
  explicitly rejected.
* Default: 0

minPasswordDigit = <positive integer>
* Specifies the minimum permitted digit or number characters when passwords are
  set or modified.
* The Splunk platform ignores negative values.
* This setting is optional.
* Password modification attempts which do not meet this requirement are
  explicitly rejected.
* Default: 0

minPasswordSpecial = <positive integer>
* Specifies the minimum permitted special characters when passwords are set
  or modified.
* The semicolon character is not allowed.
* The Splunk platform ignores negative values.
* This setting is optional.
* Password modification attempts which do not meet this requirement are
  explicitly rejected.
* Default: 0

expirePasswordDays = <positive integer>
* Specifies the number of days before the password expires after a reset.
* Minimum value: 0
* Maximum value: 3650
* the Splunk platform ignores negative values.
* This setting is optional.
* Default: 90

expireAlertDays = <positive integer>
* Specifies the number of days to issue alerts before password expires.
* Minimum value: 0
* Maximum value: 120
* The Splunk platform ignores negative values.
* This setting is optional.
* Alerts appear in splunkd.log.
* Default: 15

expireUserAccounts = <boolean>
* Specifies whether password expiration is enabled.
* This setting is optional.
* Default: false (user passwords do not expire)

forceWeakPasswordChange = <boolean>
* Specifies whether users must change a weak password.
* This setting is optional.
* Default: false (users can keep weak password)

lockoutUsers = <boolean>
* Specifies whether locking out users is enabled.
* This setting is optional.
```

```
* If you enable this setting on members of a search head cluster, user lockout
  state applies only per SHC member, not to the entire cluster.
* Default: true (users are locked out on incorrect logins)

lockoutMins = <positive integer>
* The number of minutes that a user is locked out after entering an incorrect
  password more than 'lockoutAttempts' times in 'lockoutThresholdMins' minutes.
* Any value less than 1 is ignored.
* Minimum value: 1
* Maximum value: 1440
* This setting is optional.
* If you enable this setting on members of a search head cluster, user lockout
  state applies only per SHC member, not to the entire cluster.
* Default: 30

lockoutAttempts = <positive integer>
* The number of unsuccessful login attempts that can occur before a user is locked out.
* The unsuccessful login attempts must occur within 'lockoutThresholdMins' minutes.
* Any value less than 1 is ignored.
* Minimum value: 1
* Maximum value: 64
* This setting is optional.
* If you enable this setting on members of a search head cluster, user lockout
  state applies only per SHC member, not to the entire cluster.
* Default: 5

lockoutThresholdMins = <positive integer>
* Specifies the number of minutes that must pass from the time of the first failed
  login before the failed login attempt counter resets.
* Any value less than 1 is ignored.
* Minimum value: 1
* Maximum value: 120
* This setting is optional.
* If you enable this setting on members of a search head cluster, user lockout
  state applies only per SHC member, not to the entire cluster.
* Default: 5

enablePasswordHistory = <boolean>
* Specifies whether password history is enabled.
* When set to "true", the Splunk platform maintains a history of passwords
  that have been used previously.
* This setting is optional.
* Default: false

passwordHistoryCount = <positive integer>
* The number of passwords that are stored in history. If password
  history is enabled, on password change, user is not allowed to pick an
  old password.
* This setting is optional.
* Minimum value: 1
* Maximum value: 128
* Default: 24

constantLoginTime = <decimal>
* The amount of time, in seconds, that the authentication manager
  waits before returning any kind of response to a login request.
* This setting helps mitigate login timing attacks. If you want to use the
  setting, test it in your environment first to determine the appropriate
  value.
* When you configure this setting, a login failure is guaranteed to take at least the
  amount of time you specify. The authentication manager
  adds a delay to the actual response time to keep this guarantee.
```

```
* The values can use decimals. "0.025" would make responses take a
  consistent 25 milliseconds or slightly more.
* This setting is optional.
* Minimum value: 0 (Disables login time guarantee)
* Maximum value: 5.0
* Default: 0

verboseLoginFailMsg = <boolean>
* Specifies whether or not the login failure message explains
  the failure reason.
* When set to true, the Splunk platform displays a message on login
  along with the failure reason.
* When set to false, the Splunk platform displays a generic failure
  message without a specific failure reason.
* This setting is optional.
* Default: true
```

Security Assertion Markup Language (SAML) settings

```
[<saml-authSettings-key>]
* Follow this stanza name with the following setting/value pairs.
* The <authSettings-key> must be one of the values listed in the
* authSettings setting, specified above in the [authentication] stanza.

fqdn = <string>
* The fully qualified domain name where this splunk instance is running.
* If this value is not specified, the Splunk platform uses the value specified
  in server.conf.
* If this value is specified and 'http://' or 'https://' prefix is not
  present, the Splunk platform uses the SSL setting for Splunk Web.
* This setting is optional.
* the Splunk platform uses this information to populate the 'assertionConsumerServiceUrl'.
* Default: empty string

redirectPort = <port number>
* The port where SAML responses are sent.
* Typically, this is the web port.
* If internal port redirection is needed, set this port and the
  'assertionconsumerServiceUrl' in the AuthNRequest contains this port
  instead of the Splunk Web port.
* To prevent any port information to be appended in the
  'assertionConsumerServiceUrl' setting, set this to 0.
* No default.

idpSSOUrl = <url>
* The protocol endpoint on the IDP (Identity Provider) where the
  AuthNRequests should be sent.
* Required.
* SAML requests fail if this information is missing.
* No default.

idpAttributeQueryUrl = <url>
* The protocol endpoint on the IDP (Identity Provider) where the setting
  query requests should be sent.
* Attribute queries can be used to get the latest 'role' information,
  if there is support for Attribute queries on the IDP.
* This setting is optional.
* When this setting is absent, the Splunk platform caches the role information
  from the SAML assertion and use it to run saved searches.
```

```
* No default.

idpCertPath = <string>
* This value is relative to $SPLUNK_HOME/etc/auth/idpCerts.
* The value for this setting can be the name of the certificate file or a directory.
* If it is empty, the Splunk platform automatically verify with certificates in all
  subdirectories present in $SPLUNK_HOME/etc/auth/idpCerts.
* If the SAML response is to be verified with a IdP (Identity Provider) certificate that
  is self signed, then this setting holds the filename of the certificate.
* If the SAML response is to be verified with a certificate that is a part of a
  certificate chain(root, intermediate(s), leaf), create a subdirectory and place the
  certificate chain as files in the subdirectory.
* If there are multiple end certificates, create a subdirectory such that, one
  subdirectory holds one certificate chain.
* If multiple such certificate chains are present, the assertion is considered verified,
  if validation succeeds with any certificate chain.
* The file names within a certificate chain should be such that root certificate
  is alphabetically before the intermediate which is alphabetically before of
  the end cert.
  ex. cert_1.pem has the root, cert_2.pem has the first intermediate cert,
      cert_3.pem has the second intermediate certificate and cert_4.pem has the
      end certificate.
* This setting is required if 'signedAssertion' is set to true.
* Otherwise, it is optional.
* No default.

idpCertExpirationWarningDays = <positive integer>
* The number of days before an identity provider certificate expires. During this period, when a
  SAML login occurs, the Splunk platform generates a certificate expiration warning log.
* You can control how often the Splunk platform generates warning logs for the
  same certificate with the 'IdpCertExpirationCheckInterval' setting.
* Minimum value: 1
* Maximum value: 365
* This setting is optional.
* If you enable this setting on members of a search head cluster, the instance that
  processes the login request generates the certificate expiration warning log.
* Default: 90

idpCertExpirationCheckInterval = <interval><unit>
* How long a Splunk platform instance must wait, after generating a certificate
  expiration warning log after a login, to generate another one.
* The Splunk platform caches the certificate fingerprint when a SAML user logs in.
  If the client sends the same certificate on another login, the Splunk platform reviews the
  cache. If at least 'idpCertExpirationCheckInterval' has not passed since the last time
  it generated a log for a certificate that is in the cache, it won't generate another log.
* Default: 1d

idpSLOUrl = <string>
* The protocol endpoint on the IDP (Identity Provider) where a SP
  (Service Provider) initiated Single logout request should be sent.
* This setting is optional.
* No default.

errorUrl = <string>
* The URL to be displayed for a SAML error.
* Errors may be due to erroneous or incomplete configuration in either
  the IDP or the Splunk platform.
* This URL can be absolute or relative.
  * Absolute URLs should follow the pattern
    <protocol>://[//]<host> e.g. https://www.external-site.com.
  * Relative URLs should start with '/'. A relative url shows up as an
    internal link of the Splunk instance, for
```

```
    example: https://splunkhost:port/relativeUrlWithSlash
* No default.

errorUrlLabel = <string>
* Label or title of the content pointed to by errorUrl.
* This setting is optional.
* No default.

entityId = <string>
* The entity ID for SP connection as configured on the IDP.
* Required.
* No default.

issuerId = <string>
* Required.
* The unique identifier of the identity provider.
  The value of this setting corresponds to the setting "entityID" of
  "EntityDescriptor" node in IdP metadata document.
* If you configure SAML using IdP metadata, this field is extracted from
  the metadata.
* If you configure SAML manually, then you must configure this setting.
* When the Splunk platform tries to verify the SAML response, the issuerId
  specified here must match the 'Issuer' field in the SAML response. Otherwise,
  validation of the SAML response fails.

signAuthnRequest = <boolean>
* Whether or not the Splunk platform should sign AuthNRequests.
* This setting is optional.
* Default: true

signedAssertion = <boolean>
* Whether or not the SAML assertion has been signed by the IDP.
* If set to false, the Splunk platform does not verify the signature
  of the assertion using the certificate of the IDP.
* The software accepts both signed and encrypted assertions.
* Changing this to false will not affect encrypted assertions.
* This setting is optional.
* Default: true

attributeQuerySoapPassword = <password>
* The password to be used when making an attribute query request.
* Attribute query requests are made using SOAP using basic authentication
* This setting is required if 'attributeQueryUrl' is specified.
* Otherwise, it is optional.
* This string is obfuscated upon splunkd startup.
* No default.

attributeQuerySoapUsername = <string>
* The username to be used when making an attribute query request.
* Attribute Query requests are made using SOAP using basic authentication
* This setting is required if 'attributeQueryUrl' is specified.
* Otherwise, it is optional.
* No default.

attributeQueryRequestSigned = <boolean>
* Whether or not to sign attribute query requests.
* Default: true

attributeQueryResponseSigned = <boolean>
* Specifies whether attribute query responses are signed.
* If set to false, the Splunk platform does not verify the signature in
  the response using the certificate of the IDP.
```

```
* This setting is optional.
* Default: true

partialChainCertVerification = <boolean>
* Whether or not authentication uses the OpenSSL X509_V_FLAG_PARTIAL_CHAIN
* flag when performing validation on a SAML certificate chain.
* Configuring this setting to "true" lets verification of SAML certificates
* succeed even in cases where a complete certificate chain cannot be built
* back to a self-signed trust anchor certificate.
* When set to "true", intermediate certificates in the trust store are
* treated as trust-anchors in the same way as self-signed root certificate
* authority certificates.
* Uses X509_V_FLAG_PARTIAL_CHAIN flag during certificate verification.
* This setting is optional.
* Default: false

redirectAfterLogoutToUrl = <string>
* The user is redirected to this url after logging out of the Splunk platform.
* If this is not specified, and 'idpSLO' is also not set, the user is
  redirected to splunk.com after logout.
* This setting is optional.
* No default.

defaultRoleIfMissing = <string>
* If the IdP does not return any AD groups or Splunk roles as a part of the
  assertion, the Splunk platform uses this value if provided.
* This setting is required when you configure 'skipAttributeQueryRequestForUsers'. Otherwise, it is
optional.
* No default.

skipAttributeQueryRequestForUsers = <comma-separated list of users>
* To skip attribute query requests being sent to the IdP for certain users,
  add them with this setting.
* By default, attribute query requests are skipped for local users.
* If you configure this setting for non-local users, you must also configure 'defaultRoleIfMissing'.
* No default.

maxAttributeQueryThreads = <integer>
* Number of threads to use to make attribute query requests.
* Changes to this setting require a restart to take effect.
* This setting is optional.
* Maximum value: 10
* Default: 2

maxAttributeQueryQueueSize = <integer>
* The number of attribute query requests to queue, set to 0 for infinite
  size.
* Changes to this setting require a restart to take effect.
* This setting is optional.
* Default: 50

attributeQueryTTL = <integer>
* Determines the time for which the Splunk platform caches the user and role
  information (time to live).
* After the ttl expires, the Splunk platform makes an attribute query request to
  retrieve the role information.
* This setting is optional.
* Default: 3600

saml_negative_cache_timeout = <nonnegative decimal>
* The amount of time, in seconds, that the Splunk platform remembers that a non-existent
  user on a SAML provider does not exist.
```

```
* This setting is useful when you want to avoid frequent SAML queries for users
  that do not exist on the SAML provider.
* This setting does not prevent SAML queries on login. Login always queries the SAML
  provider to confirm that a user exists.
* Default: 3600

scriptPath = <string>
* The name of the authentication extension script to run.
* The auth system expects the script to be in Python version 3, and looks for
  it in the $SPLUNK_HOME/etc/auth/scripts directory.
* No default.

python.version = {default|python|python2|python3}
* For Python scripts only, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.

scriptTimeout = <string>
* The maximum time the script can run before the auth system forcefully terminates it.
* If you set to zero, the auth system never kills the script.
* If you set to below 500ms, the auth system uses a minimum of 500 ms.
* Optional
* Default: 10s

scriptFunctions = <semicolon-separated list>
* Script functions to be enabled for authentication extensions.
* Expressed as a list.
* Supported values are 'getUsers', 'getUserInfo', and 'login'.
* To use the 'getUsers' function, you must also enable the 'getUserInfo' function.
* You must set this if you define 'scriptPath'.
* No default.

getUsersPrecacheLimit = <integer>
* The number of users to pre-cache on startup for the 'getUsers' script function.
* If you enable the 'getUsers' function, the script executes when splunkd starts up.
* As part of startup, splunkd caches user information that the 'getUsers' script returns,
  and this setting specifies how many users to cache.
* If you set 'getUsersPrecacheLimit' to 0, splunkd caches all user information that
  the 'getUsers' function returns.
* Default: 1000

getUserInfoTtl = <string>
* When you configure the auth system to use SAML as an authentication method,
  it runs the 'getUserInfo' script function to retrieve information from the
  SAML identity provider when users perform ad-hoc operations such as working
  with tokens and saved searches.
* This setting controls how long the auth system caches information that it
  retrieves with the 'getUserInfo' script function.
* This setting does not control how the method retrieves user information
  when one logs in using the standard SAML login flow through a browser.
* These return values are cached on a per-user basis.
* This value also applies if users are retrieved en masse using the scripts
  getUsers() function.
* If you configure both AQR and authentication extensions (meaning, you configure
  both 'attributeQueryTTL' and 'getUserInfoTTL', this setting takes precedence.
* This setting is optional.
* Default: 10s

scriptSecureArguments = <key:value>;[<key:value>;]...
* A list of inputs, expressed as key-value pairs, that will be made available
```

```
  in plaintext to the custom user information retrieval script.
* On startup, the auth system encrypts the values you specify here.
* Use this setting to safely store passwords, tokens, or other credentials
  that the script needs to function.
* If you use the 'commonAuth.py' sample script to read in the inputs, these values
  are available as normal arguments for all functions.
* This setting is optional.
* No default.

useAuthExtForTokenAuthOnly = <boolean>
* Whether authentication extension scripts run for all types of authentication,
  or only for token based authentication.
* If set to "true", the 'getUserInfo' script only runs when making token based authentication calls.
* Other calls that rely on fetching SAML user information,
  such as saved searches and displaying SAML users,
  will use the persistent cache that is defined in the [userToRoleMap_<saml-authSettings-key>] stanza.
* This setting is optional.
* Default: true

assertionTimeSkew = <integer>
* The amount of clock skew, in seconds, that can occur between the Splunk platform and
  an identity provider that presents SAML assertions that contain 'NotBefore'
  and 'NotOnOrAfter' attributes.
* If you set this, the Splunk platform accepts a SAML assertion as valid if
  the clock skew between the assertion validity interval and the system time on the
  Splunk instance is not greater than the value of this setting.
* NOTE: Setting this to too high a value can allow for replay attacks and is a security risk.
* This setting is optional.
* Default: 120

allowSslCompression = <boolean>
* If set to true, the server allows clients to negotiate SSL-layer
  data compression.
* This setting is optional.
* Default: The value of 'allowSslCompression' in the server.conf file

cipherSuite = <cipher suite string>
* If set, the Splunk platform uses the specified cipher string for the HTTP server.
* Attribute query requests might fail if the IDP requires a relaxed
  ciphersuite.
* Use "openssl s_client -cipher 'TLSv1+HIGH:@STRENGTH' -host <IDP host> -port 443"
  to determine if the Splunk platform can connect to the IDP.
* This setting is optional.
* Default: The value or 'cipherSuite' in the server.conf file

sslVersions = <versions_list>
* Comma-separated list of SSL versions to support.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2"
* Default: The value of 'sslVersions' in the server.conf file

sslCommonNameToCheck = <commonName>
* If set, and 'sslVerifyServerCert' is set to true,
  splunkd limits most outbound HTTPS connections to hosts which use
  a cert with this common name.
* This setting is optional.
* Default: The value of 'cipherSuite' in the server.conf file

sslAltNameToCheck = <alternateName1>, <alternateName2>, ...
* If this value is set, and 'sslVerifyServerCert' is set to true,
  splunkd is also willing to verify certificates which have a so-called
  "Subject Alternate Name" that matches any of the alternate names in this list.
* This setting is optional.
```

```
* Default: The value of 'sslAltNametoCheck' in the server.conf file

ecdhCurveName = <string>
* DEPRECATED; use 'ecdhCurves' instead.
* Elliptic Curve-Diffie Hellman (ECDH) curve to use for ECDH key negotiation.
* Default: The value of 'ecdhCurveName' in the server.conf file

ecdhCurves = <comma separated list>
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* The server supports only the curves specified in the list.
* The Splunk platform only supports named curves that have been
  specified by their SHORT names.
* The list of valid named curves by their short/long names can be obtained
  by executing this CLI command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Example setting: ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: The value of 'ecdhCurves' in the server.conf file

clientCert = <path>
* Full path to the client certificate Privacy-Enhanced Mail (PEM) format file.
* Certificates are auto-generated upon first starting the Splunk platform.
* You may replace the auto-generated certificate with your own.
* If not set, Splunk uses the setting specified in
  server.conf/[sslConfig]/'serverCert'.
* Default: $SPLUNK_HOME/etc/auth/server.pem

sslKeysfile = <filename>
* DEPRECATED; use 'clientCert' instead.
* Location of the PEM file in the directory specified by 'caPath'.
* Default: server.pem

sslPassword = <password>
* The server certificate password.
* If not set, the Splunk platform uses the setting specified in server.conf.
* This setting is optional.
* Default: password

sslKeysfilePassword = <password>
* DEPRECATED; use 'sslPassword' instead.

caCertFile = <filename>
* The public key of the signing authority.
* If not set, the Splunk platform uses the setting specified in server.conf.
* This setting is optional.
* Default: cacert.pem

caPath = <path>
* DEPRECATED; use absolute paths for all certificate files.
* If certificate files given by other settings in this stanza are not absolute
  paths, then they are relative to this path.
* Default: $SPLUNK_HOME/etc/auth

sslVerifyServerCert = <boolean>
* Used by distributed search: when making a search request to another
  server in the search cluster.
* If not set, the Splunk platform uses the setting specified in server.conf.
* This setting is optional.
* No default.

sslVerifyServerName = <boolean>
```

```
* Whether or not splunkd, as a client, performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

blacklistedAutoMappedRoles = <comma separated list>
* DEPRECATED; use 'excludedAutoMappedRoles' instead.

excludedAutoMappedRoles = <comma separated list>
* A list of Splunk roles for which the Splunk platform is not to
  auto-map from the identity provider response.
* This setting is optional.
* Default: admin, power

blacklistedUsers = <comma separated list>
* DEPRECATED; use 'excludedUsers' instead.

excludedUsers = <comma separated list>
* Comma separated list of user names from the IDP response to be
  excluded by splunk platform.
* This setting is optional.
* No default.

nameIdFormat = <string>
* If supported by IDP, while making SAML Authentication request this value can
  be used to specify the format of the Subject returned in SAML Assertion.
* This setting is optional.
* No default.

ssoBinding = <string>
* The binding that is used when making a SP-initiated SAML request.
* Acceptable options are "HTTPPost" and "HTTPRedirect".
* This binding must match the one configured on the IDP.
* This setting is optional.
* Default: HTTPPost

sloBinding = <string>
* The binding that is used when making a logout request or sending a logout
  response to complete the logout workflow.
* Acceptable options are "HTTPPost" and "HTTPRedirect".
* This binding must match the one configured on the IDP.
* This setting is optional.
```

```
* Default: HTTPPost

signatureAlgorithm = RSA-SHA1 | RSA-SHA256 | RSA-SHA384 | RSA-SHA512
* The signature algorithm that is used for outbound SAML messages,
  for example, SP-initiated SAML request.
* This setting is only used when 'signAuthnRequest' is set to "true".
* This setting is applicable for both HTTP POST and HTTP Redirect binding.
* RSA-SHA1 corresponds to 'http://www.w3.org/2000/09/xmldsig#rsa-sha1'.
* RSA-SHA256, RSA-SHA384, and RSA-SHA512 correspond to 'http://www.w3.org/2001/04/xmldsig-more'.
* This algorithm is sent as a part of 'sigAlg'.
* For improved security, set to "RSA-SHA256", "RSA-SHA384", or "RSA-SHA512".
* This setting is optional.
* Default: RSA-SHA1

inboundSignatureAlgorithm = RSA-SHA1;RSA-SHA256;RSA-SHA384;RSA-SHA512
* A semicolon-separated list of signature algorithms for the SAML responses
  that you want Splunk Web to accept.
* The Splunk platform rejects any SAML responses that are not signed by
  any one of the specified algorithms.
* This setting is applicable for both HTTP POST and HTTP Redirect binding.
* For improved security, set to "RSA-SHA256", "RSA-SHA384", or "RSA-SHA512".
* This setting is optional.
* Default: RSA-SHA1;RSA-SHA256;RSA-SHA384;RSA-SHA512

inboundDigestMethod = SHA1;SHA256;SHA384;SHA512
* A semicolon-separated list of digest methods for the SAML responses
  that you want Splunk Web to accept.
* The Splunk platform rejects any SAML responses that are not hashed by
  any one of the specified methods.
* This setting is applicable for HTTP POST binding only.
* For improved security, set to "SHA256", "SHA384", or "SHA512".
* This setting is optional.
* Default: SHA1;SHA256;SHA384;SHA512

replicateCertificates = <boolean>
* If set to "true", IdP certificate files are replicated across search head cluster setup.
* If disabled, IdP certificate files need to be replicated manually across SHC,
  otherwise verification of SAML-signed assertions fails.
* This setting has no effect if search head clustering is disabled.
* This setting is optional.
* Default: true

lockRoleToFullDN = <boolean>
* Determines how the auth system handles authentication when it receives a
  Security Assertion Markup Language (SAML) assertion from an identity
  provider (IdP) in specific cases.
* This setting applies only under the following conditions:
  * You have configured a Common Name (CN) mapping to a Splunk role
    under a [roleMap_SAML] stanza in authentication.conf. The auth system
    ignores this setting if you have configured a full Distinguished Name (DN)
    role mapping.
  * The IdP returns a full DN as part of the SAML assertion. The auth system
    ignores this setting if the IdP does not return a full DN in the assertion.
* If set to "false", the auth system uses the first part of the DN that the IdP
  provides in the assertion, and ignores the rest of the DN.
* If set to "true", the auth system does the following:
  * If you have configured a role mapping under the [roleMap_SAML] stanza that
    contains the full DN, the auth system uses the DN and logs the user in.
  * If you have configured a role mapping under the [roleMap_SAML] stanza that
    contains the CN, but not the full DN, the auth system successfully logs in
    the first user whose CN matches the role mapping, and records the full
    DN into a [lockedRoleToFullDNMap_SAML] stanza in authentication.conf.
```

```
   * The auth system then rejects subsequent authentication attempts by users
     that have a matching CN but do not have a full DN. It logs such rejections
     in splunkd.log.
   * To stop authentication failures in this case, as a Splunk admin, you must
     add the DN to the [roleMap_SAML] stanza in authentication.conf. Editing the
     [lockedRoleToFullDNMap_SAML] stanza to have different DNs with identical CNs
     map to different roles is not supported.
* Example: if this setting is "true" and you map a role in authentication.conf
  as follows:

  [roleMap_SAML]
  power=CN=PowerUsers

  and later, a SAML assertion arrives with the following DN:
  CN=PowerUsers,OU=Americas,DC=splunkcorp,DC=com

  then the auth system logs in the user who presented this assertion,
  writes an entry to authentication.conf like the following:

  [lockedRoleToFullDNMap_SAML]
  power=CN=PowerUsers,OU=Americas,DC=splunkcorp,DC=com

  and rejects further login attempts from users that present an assertion with
  the same CN ("CN=PowerUsers"), that is part of a different DN (for example,
  "CN=PowerUsers,OU=EMEA,DC=splunkcorp,DC=com",
  rather than "CN=PowerUsers,OU=Americas,DC=splunkcorp,DC=com").
* Default: true

allowPartialSignatures = <boolean>
* OPTIONAL
* When enabled, the Splunk authentication system only requires the SAML assertion block to be
  signed (but not necessarily the entire SAML response).
* When disabled, the entire SAML response must be signed for the login to succeed.
* Defaults to 'true'

allowEntities = <boolean>
* Whether or not the Splunk authentication system considers
  SAML assertions with XML entity references as valid.
* A value of "true" means the Splunk authentication system
  considers SAML assertions with XML entity references as
  valid assertions.
* A value of "false" means the Splunk authentication system
  considers SAML assertions with XML entity references as invalid
  assertions.
* CAUTION: Changing this setting from its default value could
  potentially pose a security risk. Do not change the value without
  explicit permission from Splunk Support.
* Default: false
```

Map roles

```
[roleMap_<saml-authSettings-key>]
* The mapping of Splunk roles to SAML groups for the SAML stanza specified
  by '<authSettings-key>'.
* If a SAML group is not explicitly mapped to a Splunk role, but has
  the same name as a valid Splunk role then for ease of configuration,
  it is auto-mapped to that Splunk role.
* Follow this stanza name with several Role-to-Group(s) mappings as defined
  below.
```

```
<Splunk RoleName> = <SAML group string>
* Maps a Splunk role (from authorize.conf) to SAML groups
* This SAML group list is semicolon delimited (no spaces).
* List several of these setting/value pairs to map several Splunk roles to
  SAML Groups.
* If the role mapping is not specified, Splunk expects Splunk roles in the
  assertion and attribute query response returned from the IDP.
```

SAML User Roles Map

```
[userToRoleMap_<saml-authSettings-key>]
* The mapping of SAML user to Splunk roles, real names, and emails,
  for the SAML stanza specified by '<authSettings-key>'.
* Follow this stanza name with several User-to-Role::Realname::Email mappings
  as defined below.
* The auth system uses this stanza only in the following scenarios:
  * The IdP that the auth system interacts with supports neither Attribute Query Requests nor authentication
extension scripts.
  * The IdP does support authentication scripts, but the 'useAuthExtForTokenAuthOnly' setting has a value
of "true".

<SAML User> = <Splunk Roles string>::<Realname>::<Email>
* Maps a SAML user to a Splunk role(from authorize.conf), real name, and email
* The Splunk Roles string is semicolon delimited (no spaces).
* The Splunk Roles string, Realname and Email are :: delimited (no spaces).
```

Locked up map of roles to SAML group DNs

```
[lockedRoleToFullDNMap_<saml-authSettings-key>]
* This stanza is an output stanza that the Splunk auth system creates
  only under certain conditions.
* The stanza applies only if you have set 'lockRoleToFullDN' to "true".
  Nothing happens if 'lockRoleToFullDN' is "false".
* See the 'lockRoleToFullDN' setting for information on the acronyms that
  are used in this setting description.
* When the auth system receives a SAML assertion from an IdP that includes
  a group DN, it performs several checks:
  * First, it checks to see if the CN portion of the group DN that the IdP
    provided in the assertion is a match to any CN that you have configured
    in authentication.conf under the '[roleMap_SAML]' stanza.
  * If a CN matches, and you have not previously performed a mapping
    of SAML group DN to Splunk role, the auth system creates an entry underneath
    this stanza, in the following format:

    <Splunk role name> = <SAML group DN string>
  * This means that the auth system has locked the Splunk role name that
    you configured in the '[roleMap_SAML]' stanza to the DN that the IdP
    provided in the assertion.
  * After creating the entry, the auth system maps a user with the group
    DN that the IdP provided to the corresponding Splunk role and lets this
    user - and only this user - log in.
  * It then rejects users that present the same CN, but that do not provide a
    DN that exactly matches what was written under this stanza, for this
    Splunk role, on future login attempts.
```

```
  * It also writes a warning message to splunkd.log stating that the DN that
    the IdP presented has already been locked to a Splunk role.
  * Entries in this stanza map a Splunk role to a semicolon separated list of
    group DNs. DNs referenced in this stanza are enforced to have unique CNs
    (a CN cannot map to multiple DNs).
```

Authentication Response Attribute Map

```
[authenticationResponseAttrMap_SAML]
* The Splunk platform expects emails, real names, and roles to be returned as SAML
  attributes in SAML assertion. This stanza can be used to map attribute names
  to what is expected. These are optional settings, and are only needed for
  certain IDPs.

role = <string>
* Attribute name to be used as role in SAML Assertion.
* This setting is optional.
* Default: role

realName = <string>
* Attribute name to be used as realName in SAML Assertion.
* This setting is optional.
* Default: realName

mail = <string>
* Attribute name to be used as email in SAML Assertion.
* This setting is optional.
* Default: mail
```

Settings for Proxy SSO mode

```
[roleMap_proxySSO]
* The mapping of Splunk roles to groups passed in headers from the proxy server.
* If a group is not explicitly mapped to a Splunk role, but has
  the same name as a valid Splunk role, then, for ease of configuration, it is
  auto-mapped to that Splunk role.
* Follow this stanza name with several Role-to-Group(s) mappings as defined
  later in this section.

<Splunk RoleName> = <Group string>
* Maps a Splunk role (from authorize.conf) to one or more groups.
* This group list is semicolon delimited (no spaces).
* List several of these setting value pairs to map several Splunk roles to
  groups.
* If role mapping is not specified, the user is logged in with the
  default User role.
* No default.

[userToRoleMap_proxySSO]
* The mapping of ProxySSO user to Splunk roles
* Follow this stanza name with several User-to-Role(s) mappings as defined
  later in this section.

<ProxySSO User> = <Splunk Roles string>
* Maps a ProxySSO user to Splunk role (from authorize.conf).
```

```
* This Splunk Role list is semicolon delimited (no spaces).
* No default.

[proxysso-authsettings-key]
* Follow this stanza name with the setting/value pairs listed below.

defaultRoleIfMissing = <splunk role>
* If Splunk roles cannot be determined based on role mapping, the Splunk platform
  uses the default configured splunk role.
* This setting is optional.

blacklistedAutoMappedRoles = <comma separated list>
* DEPRECATED; use 'excludedAutoMappedRoles' instead.

excludedAutoMappedRoles = <comma separated list>
* Comma-separated list of Splunk roles that should be prevented
  from being auto-mapped by the Splunk platform from the proxy server headers.
* This setting is optional.

blacklistedUsers = <comma separated list>
* DEPRECATED; use 'excludedUsers' instead.

excludedUsers = <comma separated list>
* Comma-separated list of user names from the proxy server headers to be
  excluded by the Splunk platform.
* This setting is optional.
```

Secret Storage

```
[secrets]

disabled = <boolean>
* Toggles integration with platform-provided secret storage facilities.
* NOTE: Splunk plans to submit Splunk Enterprise for Common Criteria
  evaluation. Splunk does not support using the product in Common
  Criteria mode until it has been certified by NIAP. See the "Securing
  Splunk Enterprise" manual for information on the status of Common
  Criteria certification.
* Default (if Common Criteria mode is enabled): false
* Default (if Common Criteria mode is disabled): true


filename = <filename>
* Designates a Python script that integrates with platform-provided
  secret storage facilities, like the GNOME keyring software for the
  GNOME desktop manager.
* Set <filename> to the name of a Python script located in one of the
  following directories:
    $SPLUNK_HOME/etc/apps/*/bin
    $SPLUNK_HOME/etc/system/bin
* Set <filename> to a basename. Do not user a name with path separators.
* Ensure <filename> ends with a .py file extension.
* No default.

namespace = <string>
* Use an instance-specific string as a namespace within secret storage.
* When using GNOME keyring, this namespace is used as a keyring name.
* If multiple Splunk instances must store separate sets of secrets within the
  same storage backend, customize this value to be unique for each
```

```
  Splunk instance.
* Default: splunk
```

Duo Multi-Factor Authentication (MFA) vendor settings

```
[<duo-externalTwoFactorAuthSettings-key>]
* <duo-externalTwoFactorAuthSettings-key> must be the value listed in the
  'externalTwoFactorAuthSettings' setting, specified in the [authentication]
  stanza.
* This stanza contains Duo specific multifactor authentication settings and is
  activated only when you set 'externalTwoFactorAuthVendor' to "Duo".
* All the following settings, except 'appSecretKey', are provided by Duo.

apiHostname = <string>
* Duo's API endpoint which performs the actual multifactor authentication.
* Example: apiHostname = api-xyz.duosecurity.com
* Required.
* No default.

integrationKey = <string>
* Duo's integration key for the Splunk platform.
* Must be of size = 20.
* Integration key is obfuscated before being saved here for security.
* Required.
* No default.

secretKey = <string>
* Duo's secret key for the Splunk platform.
* Must be of size = 40.
* Secret key is obfuscated before being saved here for security.
* Required.
* No default.

appSecretKey = <string>
* The Splunk application specific secret key which should be random and locally generated.
* Must be at least of size = 40 or longer.
* This secret key is not shared with Duo.
* Application secret key is obfuscated before being saved here for security.
* Required.
* No default.

failOpen = <boolean>
* If set to "true", the Splunk platform bypasses Duo multifactor authentication when
  the service is unavailable.
* This setting is optional.
* Default: false

timeout = <integer>
* The connection timeout, in seconds, for the outbound Duo HTTPS connection.
* This setting is optional.
* Default: The default Splunk HTTPS connection timeout

sslVersions = <versions_list>
* Comma-separated list of SSL versions to support for incoming connections.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* This setting is optional.
* Default: The value of 'sslVersions in the server.conf file

cipherSuite = <cipher suite string>
* The cipher string for the HTTP server.
```

```
* This setting is optional.
* Default: The value of 'cipherSuite' in the server.conf file

ecdhCurves = <comma separated list of ec curves>
* ECDH curves to use for ECDH key negotiation.
* This setting is optional.
* Default: The value of 'ecdhCurves' in the server.conf file

sslVerifyServerCert = <boolean>
* If set to true, the Splunk platform confirms the server that is
  being connected to is a valid server (authenticated).
* Both the common name and the alternate name of the server are then
  checked for a match, if they are specified in this configuration file.
* A certificate is considered verified if either is matched.
* This setting is optional.
* Default: false

sslVerifyServerName = <boolean>
* Whether or not splunkd, as a client, performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

sslCommonNameToCheck = <commonName1>, <commonName2>, ...
* If set, the Splunk platform limits outbound Duo HTTPS connections
  to a host which use a certificate with one of the listed common names.
* 'sslVerifyServerCert' must be set to "true" for this setting to work.
* This setting is optional.
* No default.

sslAltNameToCheck =  <alternateName1>, <alternateName2>, ...
* If set, the Splunk platform limits outbound duo HTTPS connections
  to host which use a certificate with one of the listed alternate names.
* 'sslVerifyServerCert' must be set to true for this setting to work.
* This setting is optional.
* No default.

sslRootCAPath = <path>
* The full path of a PEM format file containing one or more
  root CA certificates concatenated together.
* This Root CA must match the CA in the certificate chain of the SSL certificate
  returned by the Duo server.
```

```
* This setting is optional.
* No default.

useClientSSLCompression = <boolean>
* Whether or not compression is enabled between the Splunk instance and a Duo server.
* If set to "true" on client side, compression is enabled between the server and client
  as long as the server also supports it.
* If not set, the Splunk platform uses the client SSL compression setting provided in server.conf
* This setting is optional.
* Default: false

enableMfaAuthRest = <boolean>
* Determines whether splunkd requires Duo multifactor authentication against REST endpoints.
* When Duo multifactor authentication is enabled for REST endpoints, you must log in to
  the Splunk platform instance with a valid Duo multifactor authentication factor to get a
  valid session key, or requests to those endpoints must include a valid session key
  in the following format:
  'curl -k -H "Authorization:Splunk sessionKey" -X GET <resource>'
* A value of "true" means splunkd requires Duo multifactor authentication against REST endpoints.
* A value of "false" means splunkd does not require Duo multifactor authentication against REST endpoints.
* Optional.
* Default: false
```

RSA MFA vendor settings

```
[<rsa-externalTwoFactorAuthSettings-key>]
* <rsa-externalTwoFactorAuthSettings-key> must be the value listed in the
  externalTwoFactorAuthSettings setting specified in the [authentication]
  stanza.
* This stanza contains RSA-specific multifactor authentication settings and is
  activated only when you set 'externalTwoFactorAuthVendor' to "RSA".
* All the following settings can be obtained from RSA Authentication Manager 8.2 SP1.

authManagerUrl = <string>
* URL of the REST endpoint of RSA Authentication Manager.
* The Splunk platform sends authentication requests to this URL.
* Specify a HTTPS-based URL. the Splunk platform does not support communication over HTTP.
* Required.
* No default.

accessKey = <string>
* Access key needed by the Splunk platform to communicate with RSA Authentication Manager.
* Required.
* No default.

clientId = <string>
* The clientId is the agent name created on RSA Authentication Manager.
* Required.
* No default.

failOpen = <boolean>
* Whether or not the Splunk platform allows login if the RSA MFA server is unavailable.
* If set to "true", allow login in case authentication server is unavailable.
* This setting is optional.
* Default: false

timeout = <integer>
* The connection timeout, in seconds, for the outbound HTTPS connection to the RSA
  server.
* This setting is optional.
```

```
* Default: 5

messageOnError = <string>
* The message that the Splunk platform shows to the user in the case of a login failure.
* You can specify contact of admin or link to a diagnostic page.
* This setting is optional.
* No default.

sslVersions = <versions_list>
* Comma-separated list of SSL versions to support for incoming connections.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* If not set, the Splunk platform uses the value of 'sslVersions' in server.conf.
* This setting is optional.
* Default: tls1.2

cipherSuite = <cipher suite string>
* If set, the Splunk platform uses the specified cipher string for the HTTP server.
* If not set, the Splunk platform uses the value for 'cipherSuite' specified in server.conf
* This setting is optional.

ecdhCurves = <comma separated list of ec curves>
* ECDH curves to use for ECDH key negotiation.
* This setting is optional.
* Default: The value of 'ecdhCurves' in the server.conf file

sslVerifyServerCert = <boolean>
* Determines whether to verify the server being connected to is authenticated.
* If this is set to true, you should make sure that the server that is
  being connected to is a valid one (authenticated). Both the common
  name and the alternate name of the server are then checked for a
  match if they are specified in this configuration file.  A
  certificate is considered verified if either is matched.
* This setting is optional.
* Default: true

sslVerifyServerName = <boolean>
* Whether or not splunkd, as a client, performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

sslCommonNameToCheck = <commonName1>, <commonName2>, ...
```

```
* If this value is set, the Splunk platform limits outbound RSA HTTPS connections
  to host which use a cert with one of the listed common names.
* 'sslVerifyServerCert' must be set to true for this setting to work.
* This setting is optional.
* No default.

sslAltNameToCheck =  <alternateName1>, <alternateName2>, ...
* If this value is set, the Splunk platform limits outbound RSA HTTPS connections
  to host which use a cert with one of the listed alternate names.
* 'sslVerifyServerCert' must be set to true for this setting to work.
* This setting is optional.
* No default.


sslRootCAPath = <path>
* The <path> must refer to full path of a PEM format file containing one or more
  root CA certificates concatenated together.
* Required.
* This Root CA must match the CA in the certificate chain of the SSL certificate
  returned by RSA server.
* No default.

sslVersionsForClient = <versions_list>
* Comma-separated list of SSL versions to support for outgoing HTTP connections.
* If not set, Splunk uses the value for 'sslVersionsForClient' in server.conf.
* This setting is optional.
* Default: tls1.2

replicateCertificates = <boolean>
* Whether or not RSA certificate files are automatically replicated across search head
  cluster nodes.
* If set to "true", RSA certificate files are replicated across nodes in a search head
  cluster.
* If disabled, RSA certificate files need to be replicated manually across SHC or else
  MFA verification fails.
* This setting has no effect if search head clustering is disabled.
* Default: true

enableMfaAuthRest = <boolean>
* Determines whether splunkd requires RSA two-factor authentication against REST endpoints.
* When two-factor authentication is enabled for REST endpoints, either you
  must log in to the Splunk instance with a valid RSA passcode, or requests
  to those endpoints must include a valid token in the following format:
  "curl -k -u <username>:<password>:<token> -X GET <resource>"
* If set to "true", splunkd requires RSA REST two-factor authentication.
* If set to "false", splunkd does not require REST two-factor authentication.
* This setting is optional.
* Default: false
```

authentication.conf.example

```
#   Version 9.2.2
#
# This is an example authentication.conf. authentication.conf is used to
# configure LDAP, Scripted, SAML and Proxy SSO authentication in addition
# to Splunk's native authentication.
#
```

```
# To use one of these configurations, copy the configuration block into
# authentication.conf in $SPLUNK_HOME/etc/system/local/.  You must reload
# auth in manager or restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

##### Use just Splunk's built-in authentication (default):
[authentication]
authType = Splunk


##### LDAP examples

#### Basic LDAP configuration example
[authentication]
authType = LDAP
authSettings = ldaphost

[ldaphost]
host = ldaphost.domain.com
port = 389
SSLEnabled = 0
bindDN = cn=Directory Manager
bindDNpassword = password
userBaseDN = ou=People,dc=splunk,dc=com
userBaseFilter = (objectclass=splunkusers)
groupBaseDN = ou=Groups,dc=splunk,dc=com
groupBaseFilter = (objectclass=splunkgroups)
userNameAttribute = uid
realNameAttribute = givenName
groupMappingAttribute = dn
groupMemberAttribute = uniqueMember
groupNameAttribute = cn
timelimit = 10
network_timeout = 15

# This stanza maps roles you have created in authorize.conf to LDAP Groups
[roleMap_ldaphost]
admin = SplunkAdmins

#### Example using the same server as 'ldaphost', but treating each user as
#### their own group
[authentication]
authType = LDAP
authSettings = ldaphost_usergroups

[ldaphost_usergroups]
host = ldaphost.domain.com
port = 389
SSLEnabled = 0
bindDN = cn=Directory Manager
bindDNpassword = password
userBaseDN = ou=People,dc=splunk,dc=com
userBaseFilter = (objectclass=splunkusers)
groupBaseDN = ou=People,dc=splunk,dc=com
groupBaseFilter = (objectclass=splunkusers)
userNameAttribute = uid
realNameAttribute = givenName
groupMappingAttribute = uid
groupMemberAttribute = uid
```

```
groupNameAttribute = uid
timelimit = 10
network_timeout = 15

[roleMap_ldaphost_usergroups]
admin = admin_user1;admin_user2;admin_user3;admin_user4
power = power_user1;power_user2
user = user1;user2;user3

#### Sample Configuration for Active Directory (AD)
[authentication]
authSettings = AD
authType = LDAP

[AD]
SSLEnabled = 1
bindDN = ldap_bind@splunksupport.kom
bindDNpassword = ldap_bind_user_password
groupBaseDN = CN=Groups,DC=splunksupport,DC=kom
groupBaseFilter =
groupMappingAttribute = dn
groupMemberAttribute = member
groupNameAttribute = cn
host = ADbogus.splunksupport.kom
port = 636
realNameAttribute = cn
userBaseDN = CN=Users,DC=splunksupport,DC=kom
userBaseFilter =
userNameAttribute = sAMAccountName
timelimit = 15
network_timeout = 20
anonymous_referrals = 0

[roleMap_AD]
admin = SplunkAdmins
power = SplunkPowerUsers
user = SplunkUsers

#### Sample Configuration for Sun LDAP Server
[authentication]
authSettings = SunLDAP
authType = LDAP

[SunLDAP]
SSLEnabled = 0
bindDN = cn=Directory Manager
bindDNpassword = Directory_Manager_Password
groupBaseDN = ou=Groups,dc=splunksupport,dc=com
groupBaseFilter =
groupMappingAttribute = dn
groupMemberAttribute = uniqueMember
groupNameAttribute = cn
host = ldapbogus.splunksupport.com
port = 389
realNameAttribute = givenName
userBaseDN = ou=People,dc=splunksupport,dc=com
userBaseFilter =
userNameAttribute = uid
timelimit = 5
network_timeout = 8

[roleMap_SunLDAP]
```

```
admin = SplunkAdmins
power = SplunkPowerUsers
user = SplunkUsers

#### Sample Configuration for OpenLDAP
[authentication]
authSettings = OpenLDAP
authType = LDAP

[OpenLDAP]
bindDN = uid=directory_bind,cn=users,dc=osx,dc=company,dc=com
bindDNpassword = directory_bind_account_password
groupBaseFilter =
groupNameAttribute = cn
SSLEnabled = 0
port = 389
userBaseDN = cn=users,dc=osx,dc=company,dc=com
host = hostname_OR_IP
userBaseFilter =
userNameAttribute = uid
groupMappingAttribute = uid
groupBaseDN = dc=osx,dc=company,dc=com
groupMemberAttribute = memberUid
realNameAttribute = cn
timelimit = 5
network_timeout = 8
dynamicGroupFilter = (objectclass=groupOfURLs)
dynamicMemberAttribute = memberURL
nestedGroups = 1

[roleMap_OpenLDAP]
admin = SplunkAdmins
power = SplunkPowerUsers
user = SplunkUsers


##### Scripted Auth examples

#### The following example is for RADIUS authentication:
[authentication]
authType = Scripted
authSettings = script

[script]
scriptPath = "$SPLUNK_HOME/bin/python" "$SPLUNK_HOME/share/splunk/authScriptSamples/radiusScripted.py"

# Cache results for 1 second per call
[cacheTiming]
userLoginTTL    = 1
userInfoTTL     = 1


#### The following example works with PAM authentication:
[authentication]
authType = Scripted
authSettings = script

[script]
scriptPath = "$SPLUNK_HOME/bin/python" "$SPLUNK_HOME/share/splunk/authScriptSamples/pamScripted.py"

# Cache results for different times per function
[cacheTiming]
```

```
userLoginTTL    = 30s
userInfoTTL     = 1min


##### SAML auth example

[authentication]
authSettings = samlv2
authType = SAML

[samlv2]
attributeQuerySoapPassword = changeme
attributeQuerySoapUsername = test
entityId = test-splunk
idpAttributeQueryUrl = https://exsso/idp/attrsvc.ssaml2
idpCertPath = /home/splunk/etc/auth/idp.crt
idpSSOUrl = https://exsso/idp/SSO.saml2
idpSLOUrl = https://exsso/idp/SLO.saml2
signAuthnRequest = true
signedAssertion = true
attributeQueryRequestSigned = true
attributeQueryResponseSigned = true
redirectPort = 9332
cipherSuite = TLSv1 MEDIUM:@STRENGTH
nameIdFormat = urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress

[roleMap_SAML]
admin = SplunkAdmins
power = SplunkPowerUsers
user = all

[userToRoleMap_SAML]
samluser = user::Saml Real Name::samluser@domain.com

[authenticationResponseAttrMap_SAML]
role = "http://schemas.microsoft.com/ws/2008/06/identity/claims/groups"
mail = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
realName = "http://schemas.microsoft.com/identity/claims/displayname"

# Multifactor authentication example
[authentication]
externalTwoFactorAuthVendor = duo
externalTwoFactorAuthSettings = duo-mfa

# Duo specific authentication setting example
[duo-mfa]
apiHostname = api-xyz.duosecurity.com
appSecretKey = mustBeARandomStringOfSize40OrLonger
integrationKey = mustBeADuoProvidedStringOfSize20
secretKey = mustBeADuoProvidedStringOfSize40
enableMfaAuthRest = true

##### Proxy SSO auth example

[authentication]
authSettings = my_proxy
authType = ProxySSO

[my_proxy]
excludedUsers = user1,user2
excludedAutoMappedRoles = admin
defaultRoleIfMissing = user
```

```
[roleMap_proxySSO]
admin = group1;group2
user = group1;group3

[userToRoleMap_proxySSO]
proxy_user1 = user
proxy_user2 = power;can_delete

[splunk_auth]
minPasswordLength = 8
minPasswordUppercase = 1
minPasswordLowercase = 1
minPasswordSpecial = 1
minPasswordDigit = 0
expirePasswordDays = 90
expireAlertDays = 15
expireUserAccounts = true
forceWeakPasswordChange = false
lockoutUsers = true
lockoutAttempts = 5
lockoutThresholdMins = 5
lockoutMins = 30
enablePasswordHistory = false
passwordHistoryCount = 24
```

authorize.conf

The following are the spec and example files for `authorize.conf`.

authorize.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# create roles in authorize.conf.
#
# There is an authorize.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name authorize.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see authorize.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file takes precedence.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

[default]

```
srchFilterSelecting = <boolean>
* Determines whether a role's search filters are used for selecting or
  eliminating during role inheritance.
* If "true", the search filters are used for selecting. The filters are joined
  with an OR clause when combined.
* If "false", the search filters are used for eliminating. The filters are joined
  with an AND clause when combined.
* Example:
  * role1 srchFilter = sourcetype!=ex1 with selecting=true
  * role2 srchFilter = sourcetype=ex2 with selecting = false
  * role3 srchFilter = sourcetype!=ex3 AND index=main with selecting = true
  * role3 inherits from role2 and role 2 inherits from role1
  * Resulting srchFilter = ((sourcetype!=ex1) OR
    (sourcetype!=ex3 AND index=main)) AND ((sourcetype=ex2))
* Default: true
```

[capability::<capability>]

```
* DO NOT edit, remove, or add capability stanzas. The existing capabilities
  are the full set of Splunk system capabilities.
* the Splunk platform adds all of its capabilities this way.
* For the default list of capabilities and assignments, see authorize.conf
  under the 'default' directory.
* Only alphanumeric characters and "_" (underscore) are allowed in
  capability names.
  Examples:
  * edit_visualizations
  * view_license1
* Descriptions of specific capabilities are listed below.
```

[role_<roleName>]

```
<capability> = <enabled>
* A capability that is enabled for this role. You can list many capabilities
  for each role.
* NOTE: 'enabled' is the only accepted value here, as capabilities are
  disabled by default.
* Roles inherit all capabilities from imported roles, and you cannot disable
  inherited capabilities.
* Role names cannot have uppercase characters. Usernames, however, are
  case-insensitive.
```

```
* Role names cannot contain spaces, colons, semicolons, or forward slashes.

importRoles = <semicolon-separated list>
* A list of other roles and their associated capabilities that the Splunk
  platform should import.
* Importing other roles also imports the other aspects of that role, such as
  allowed indexes to search.
* Default: A role imports no other roles

grantableRoles = <semicolon-separated list>
* A list of roles that determines which users, roles, and capabilities
  that a user with a specific set of permissions can manage.
* This setting lets you limit the scope of user, role, and capability
  management that these users can perform.
* When you set 'grantableRoles', a user that holds a role with the
  'edit_roles_grantable' and 'edit_user' capabilities can do only the
  following with regards to access control management for the Splunk
  Enterprise instance:
  * They can edit only the roles that contain capabilities that are a
    union of the capabilities in the roles that you specify
    with this setting.
  * Any new roles that they create can contain only the capabilities
    that are a union of these capabilities.
  * Any new roles that they create can search only the indexes that
    have been assigned to all roles that have been specified with
    this setting.
  * They can see only users who have been assigned roles that contain
    capabilities that are a union of these capabilities.
  * They can assign users only to roles whose assigned capabilities are a
    union of these capabilities.
* For this setting to work, you must assign a user at least one role
  that:
  * Has both the 'edit_roles_grantable' and 'edit_user' capabilities
    assigned to it, and
  * Does NOT have the 'edit_roles' capability assigned to it.
* Example:
  * Consider a Splunk instance where role1-role4 have the
    following capabilities:

    role1: cap1, cap2, cap3
    role2: cap4, cap5, cap6
    role3: cap1, cap6
    role4: cap4, cap8

  * And user1-user4 have been assigned the following roles:
    user1: role1
    user2: role2
    user3: role3
    user4: role4

  * If you define the 'grantableRoles' setting as follows for
    the 'power' role:

  *      [role_power]
  *      grantableRoles = role1;role2

  * and edit the role so that the 'edit_roles_grantable'
    capability is selected, and the 'edit_roles' capability
    is not selected, then a user that has been assigned the 'power' role
    can make only the following access control changes on the instance:
    * View or edit the following users: user1, user2, user3
    * Assign the following roles: role1, role2, role3
```

```
    * Create roles with the following capabilities: cap1, cap2, cap3,
    cap4, cap5, cap6
* Only the 'admin' role holds the 'edit_roles_grantable' capability on
  a new Splunk Enterprise installation.
* If you make changes to the 'admin' role, 'grantableRoles' is set to
  "admin".
* This setting does not work if you use tokens to authenticate into a
  Splunk Enterprise instance.
* Default (if 'admin' role is edited): admin
* Default (otherwise): No default

srchFilter = <semicolon-delimited list>
* A list of search filters for this role.
* To override any search filters from imported roles, set this to "*", as
  the 'admin' role does.
* Default: the Splunk platform does not perform search filtering

fieldFilter-<fieldname> = <option>
* Use the 'fieldFilter' configuration to apply a field filter to a specific role
  at search time. This field filter affects the results of searches run by
  users that have the role. The field filter can remove indexed or default
  fields from the results, or it can censor values of specific fields when
  those fields appear in the results.
  * NOTE: Role-based field filters do not support searches that use generating
    commands other than the 'search' command.
* The values available for <option> depend on whether the value of <fieldname>
  is "_raw" or any other field name.
  * When the value of <fieldname> is "_raw", <option> is a sed expression.
    * The sed expression acts on searches to which this filter is applied. The
      sed expression replaces strings in search results that are matched by a
      regular expression (s) or transliterates characters found in search
      results with corresponding characters provided by the sed expression (y).
      * The syntax for using the sed (s) command to replace strings in search
        results that are matched by a regular expression is:
          s/<regex>/<replacement>/<flags>
      * <regex> is a PCRE regular expression, which can include capturing
        groups.
      * <replacement> is a string that replaces the regular expression match.
        Use \<n> for back references, where <n> is a single digit.
      * <flags> can either be "g", to globally replace all matches, or a
        number to replace a specified number of matches. Other sed flags for
        the (s) command are not supported.
    * The syntax for using the sed (y) command to transliterate characters
      that the Splunk software finds in search results with corresponding
      characters that you provide is:
        y/<source_characters>/<destination_characters>/
      * The (y) command syntax transliterates the <source_characters> in
        search results with corresponding <destination_characters> that you
        provide in the expression.
      * For example, 'y/abc/def/' replaces 'a' with 'd', 'b' with 'e', and 'c'
        with 'f'. This expression would change the string 'aaabbc' to
        'dddeef'.
      * The lists of <source_characters> and <destination_characters> must
        contain the same number of characters.
  * When the value of <fieldname> is any field name other than "_raw", <option>
    can be [NULL|SHA256|SHA512|<string>].
    * NULL: If <option> is NULL, the Splunk software removes the <fieldname>
      from results of searches to which this filter is applied.
    * SHA256: The Splunk software hashes the <fieldname> value with SHA-256
      encryption wherever the <fieldname> appears in results of searches to
      which this filter is applied.
    * SHA512: The Splunk software hashes the <fieldname> value with SHA-512
```

```
      encryption wherever the <fieldname> appears in results of searches to
      which this filter is applied.
    * <string>: The Splunk software replaces the <fieldname> value with the
      specified <string> wherever the <fieldname> appears in results of
      searches to which this filter is applied.
* The Splunk software processes 'fieldFilter' configurations at search time
  ahead of all other search-time operations that add fields to events,
  including field extractions.
  * This means that <fieldname> must be an indexed or default field. Fields
    that are extracted or added at search time do not exist when 'fieldFilter'
    configurations are processed.
* You cannot use wildcards to specify multiple fields for <fieldname>.
* The following example shows how you can use the 'fieldFilter' configuration
  to perform operations on fields in searches run by users with a specific role:
  * At your organization, the indexed field 'user_name' is sensitive for
    security reasons. You have a role named A, and you want users with the A
    role to be unable to access the 'user_name' field in their search results.
    Meanwhile, users with other roles should be able to see 'user_name' fields
    and values as usual.
    * If you want to remove the field from the results of searches run by
      people with role A, apply the following configuration to role A. This
      configuration provides a NULL value for <option>, which means that
      'user_name' is removed from the results of searches by people with role A:
        fieldFilter-user_name = NULL
    * If you want users with role A to see the 'user_name' field in results,
      but with censored values, such as 'user_name = XXXX', apply the following
      configuration to role A:
        fieldFilter-user_name = XXXX
* When you specify 'fieldFilter' configurations for a role that is importing
  other roles (also with 'fieldFilter' configurations), the Splunk software
  processes 'fieldFilter' configurations for the imported roles before it
  processes 'fieldFilter' configurations for roles that are importing other
  roles.
  * For example, say role A has 'fieldFilter-user_name = YYY' and role B has
    'fieldFilter-user_name = XXXX'. If role B imports role A, the Splunk
    software will process the 'fieldFilter' defined for role A first, and
    then it will process the 'fieldFilter' defined for role B. This means that
    users with role B always see 'user_name = XXXX' in their results because
    the role B 'fieldFilter' configuration is processed last.
* The Splunk software runs each role in an import hierarchy only once. If
  multiple roles in an import hierarchy apply a 'fieldFilter' configuration to
  a field, the Splunk software runs them in the order of imported roles to
  roles that are importing other roles in the import hierarchy, from left to
  right as listed in 'importRoles'.
* Do not use the 'fieldFilter' to add new fields. Use calculated fields if you
  want to add fields at search time.
* No default.

fieldFilterLimit = [sourcetype::<sourcetype>|host::<host>|source::<source>]
* Use the 'fieldFilterLimit' configuration to limit the field filters that are
  specified in a role to events with a specific 'host', 'source', or 'source
  type'.
* For example, say role A has this 'fieldFilter' configuration, which
  censors values of the 'user_name' field in searches run by users with that
  role:
     fieldFilter-user_name = xxxx
  * By itself, 'fieldFilter-user_name' configuration applies to all events with
    the 'user_name' field.
  * To apply 'fieldFilter-user_name' only to events that have the 'user_name'
    field and the "zebra" 'source type', you can add this 'fieldFilterLimit'
    configuration to role A:
     fieldFilterLimit = sourcetype::zebra
```

* When a 'fieldFilterLimit' setting is associated with a role, it applies to
  all 'fieldFilter' settings also associated with that role.
* You can specify only one value. 'fieldFilterLimit' does not support
  statements that include wildcards or the following operators: AND, OR.
* No default.

srchTimeWin = <integer>
* Maximum time range, in seconds, of a search.
* The Splunk platform applies this search time range limit backwards from the
  latest time specified for a search.
* If a user has multiple roles with distinct search time range limits, or has
  roles that inherit from roles with distinct search time range limits,
  the Splunk platform applies the least restrictive search time range limits to
  the role.
  * For example, if user X has role A (srchTimeWin = 30s), role B (srchTimeWin
    = 60s), and role C (srchTimeWin = 3600s), user X gets a maximum search time
    range of 1 hour.
* When set to '-1', the role does not have a search time range limit. This
  value can be overidden by the maximum search time range value of an inherited
  role.
* When set to '0' (infinite), the role does not have a search time range limit.
  This value cannot be overidden by the maximum search time range value of an
  inherited role.
* This setting does not apply to real-time searches.
* Default: -1

srchTimeEarliest = <integer>
* The earliest event time that can be searched, in seconds before the current
  wall clock time.
* If a user is a member of a role with a 'srchTimeEarliest' limit, or a role
  that inherits from other roles with 'srchTimeEarliest' limits, the Splunk
  platform applies the least restrictive time limit from the roles to the user.
  * For example, if a user is a member of role A (srchTimeEarliest = 86400),
    and inherits role B (srchTimeEarliest = 3600) and role C
    (srchTimeEarliest = -1 (default)), the user gets an effective earliest time
    limit of 1 day (86400 seconds) ago.
* When set to '-1', the role does not have a earliest time limit. This
  value can be overidden by the earliest time value of an inherited role.
* When set to '0' (infinite), the role does not have an earliest time limit.
  This value cannot be overidden by the earliest time limit value of an
  inherited role.
* This setting does not apply to real-time searches.
* Default: -1

srchDiskQuota = <integer>
* The maximum amount of disk space, in megabytes, that can be used by search
  jobs for a specific user with this role.
* In search head clustering environments, this setting takes effect on a
  per-member basis. There is no cluster-wide accounting.
* The dispatch manager checks the quota at the dispatch time of a search.
  Additionally, the search process checks the quota at intervals that are defined
  in the 'disk_usage_update_period' setting in limits.conf as long as the
  search is active.
* A user can occasionally exceed the quota because the search process does
  not constantly check the quota.
* Exceeding this quota causes the search to be auto-finalized immediately,
  even if there are results that have not yet been returned.
* When set to 0, this setting does not limit the amount of disk space that
  search jobs for a user with the role can use.
* Default: 100

srchJobsQuota = <integer>

```
* The maximum number of concurrently running historical searches that a user
  with this role can have.
* When set to 0, this setting does not limit the number of historical search
  jobs that can run concurrently for a user with this role.
* When 'enable_cumulative_quota = true' in limits.conf, the
  'cumulativeSrchJobsQuota' setting overrides this setting.
  * For example, under this condition, if you have a role named 'foo' for which
    'cumulativeSrchJobsQuota = 350' while 'srchJobsQuota = 100' and you have 4
    users with the 'foo' role, those users can only run 350 searches
    concurrently. If you set 'enable_cumulative_quota = false' those users can
    run 400 searches concurrently.
* This setting excludes real-time searches. See the 'rtSrchJobsQuota' setting.
* Default: 3

rtSrchJobsQuota = <integer>
* The maximum number of concurrently running real-time searches that a user
  with this role can have.
* When set to 0, this setting does not limit the number of real-time search
  jobs that can run concurrently for a user with this role.
* When 'enable_cumulative_quota = true' in limits.conf, the
  'cumulativeRTSrchJobsQuota' setting overrides this setting.
  * For example, under this condition, if you have a role named 'foo' for which
    'cumulativeRTSrchJobsQuota = 350' while 'rtSrchJobsQuota = 100' and you
    have 4 users with the 'foo' role, those users can only run 350 searches
    concurrently. If you set 'enable_cumulative_quota = false' those users can
    run 400 searches concurrently.
* Default: 6

srchMaxTime = <integer><unit>
* The maximum amount of time that search jobs from specific users with this role are
  allowed to run.
* After a search runs for this amount of time, it auto-finalizes.
* If the role inherits from other roles, the value of the 'srchMaxTime' setting is
  specified in the included roles.
* This maximum value does not apply to real-time searches.
* Examples: 1h, 10m, 2hours, 2h, 2hrs, 100s
* Default: 100days

srchIndexesDefault = <semicolon-separated list>
* A list of indexes to search when no index is specified.
* These indexes can be wild-carded ("*"), with the exception that "*" does not
  match internal indexes.
* To match internal indexes, start with an underscore ("_"). All internal indexes are
  represented by "_*".
* The wildcard character "*" is limited to match either all the non-internal
  indexes or all the internal indexes, but not both at once.
* No default.

srchIndexesAllowed = <semicolon-separated list>
* A list of indexes that this role is allowed to search.
* Follows the same wildcarding semantics as the 'srchIndexesDefault' setting.
* No default.

srchIndexesDisallowed = <semicolon-separated list>
* A list of indexes that this role does not have permission to search on or delete.
* 'srchIndexesDisallowed' takes precedence over 'srchIndexesAllowed', 'srchIndexesDefault'
  and 'deleteIndexesAllowed'. If you specify indexes in both this setting and the
  other settings, users will be unable to search on or delete those indexes.
* Follows the same wildcarding semantics as the 'srchIndexesDefault' setting.
* If you make any changes in the "Indexes" Settings panel for a role in Splunk Web,
  those values take precedence, and any wildcards you specify in this setting are lost.
* All search heads and search peers must be running Splunk Enterprise version
```

```
  8.1.0 or higher.
* No default.

deleteIndexesAllowed = <semicolon-separated list>
* A list of indexes that this role is allowed to delete.
* This setting must be used in conjunction with the 'delete_by_keyword' capability.
* Follows the same wildcarding semantics as the 'srchIndexesDefault' setting.
* No default.

cumulativeSrchJobsQuota = <integer>
* The maximum total number of concurrently running historical searches
  across all members of this role.
* For this setting to take effect, you must set the 'enable_cumulative_quota'
  setting to "true" in limits.conf.
* If a user belongs to multiple roles, the user's searches count against
  the role with the largest cumulative search quota. Once the quota for
  that role is consumed, the user's searches count against the role with
  the next largest quota, and so on.
* In search head clustering environments, this setting takes effect on a
  per-member basis. There is no cluster-wide accounting.
* When set to 0, this setting does not limit the number of real-time search
  jobs that can run concurrently across all users with this role.
* Default: 0

cumulativeRTSrchJobsQuota = <integer>
* The maximum total number of concurrently running real-time searches
  across all members of this role.
* For this setting to take effect, you must set the 'enable_cumulative_quota'
  setting to "true" in limits.conf.
* If a user belongs to multiple roles, the user's searches count against
  the role with the largest cumulative search quota. Once the quota for
  that role is consumed, the user's searches count against the role with
  the next largest quota, and so on.
* In search head clustering environments, this setting takes effect
  on a per-member basis. There is no cluster-wide accounting.
* When set to 0, this setting does not limit the number of historical search
  jobs that can run concurrently across all users with this role.
* Default: 0

kvstore_create.deny_list = <semicolon-separated list>
* A list of collections that this role doesn't have permission to perform create operations on.
* This setting can't be inherited from imported roles.
* No Default

kvstore_create.implicit_deny_list = <semicolon-separated list>
* A list of collections that this role doesn't have permission to perform create operations on.
* This setting can be inherited from imported roles.
* No Default

kvstore_update.deny_list = <semicolon-separated list>
* A list of collections that this role doesn't have permission to perform update operations on.
* This setting can't be inherited from imported roles.
* No Default

kvstore_update.implicit_deny_list = <semicolon-separated list>
* A list of collections that this role doesn't have permission to perform update operations on.
* This setting can be inherited from imported roles.
* No Default

kvstore_delete.deny_list = <semicolon-separated list>
* A list of collections that this role doesn't have permission to perform delete operations on.
* This setting can't be inherited from imported roles.
```

```
* No Default

kvstore_delete.implicit_deny_list = <semicolon-separated list>
* A list of collections that this role doesn't have permission to perform delete operations on.
* This setting can be inherited from imported roles.
* No Default

####
# Descriptions of Splunk system capabilities.
# Capabilities are added to roles to which users are then assigned.
# When a user is assigned a role, they acquire the capabilities added to that role.
####
```

[tokens_auth]

```
* Settings for token authorization.

disabled = <boolean>
* Whether or not Splunk token authorization is active.
* A value of "true" disables token authentication, and a value of
  "false" enables it.
* Default: false

expiration = <relative-time-modifier>|never
* The relative time when an authorization token expires.
* The syntax for using time modifiers is:
  * [+]<time_integer><time_unit>@<time_unit>
  * Where time_integer is an integer value and time_unit is relative
  * time unit in seconds (s), minutes (m), hours (h) or days (d) etc.
* The steps to specify a relative time modifier are:
  * Indicate the time offset from the current time.
  * Define the time amount, which is a number and a unit.
  * Specify a "snap to" time unit. The time unit indicates the nearest
    or latest time to which your time amount rounds down.
* For example, if you configure this setting to "+2h@h", the token expires at
  the top of the hour, two hours from the current time.
* For more information on relative time identifiers, see "Time Modifiers" in
  the Splunk Enterprise Search Reference Manual.
* The default value indicates that a token never expires. To set token
  expiration, you must set this value to a relative time value.
* Your account must hold the admin role to update this setting.
* This setting is optional.
* Default: +30d

ephemeralExpiration = <relative-time-modifier>
* The relative time when an ephemeral authorization token expires.
* An ephemeral token is identical to a standard authorization token, with the
  following exceptions:
  * The auth system does not keep the token in App Key Value Store.
    This means you cannot modify it after creating it.
  * Ephemeral tokens must always expire, meaning they cannot be given an
    expiration of "never".
  * Currently, ephemeral tokens can only be created using REST.
* The syntax for using time modifiers is:
  * [+]<time_integer><time_unit>@<time_unit>
  * Where time_integer is an integer value and time_unit is relative
  * time unit in seconds (s), minutes (m), hours (h) or days (d) etc.
* The steps to specify a relative time modifier are:
  * Indicate the time offset from the current time.
  * Define the time amount, which is a number and a unit.
  * Specify a "snap to" time unit. The time unit indicates the nearest
```

```
    or latest time to which your time amount rounds down.
* For example, if you configure this setting to "+2h@h", the token expires at
  the top of the hour, two hours from the current time.
* For more information on relative time identifiers, see "Time Modifiers" in
  the Splunk Enterprise Search Reference Manual.
* To set ephemeral token expiration, you must set this value to a relative time
  value.
* Your account must hold the admin role to update this setting.
* This setting is optional.
* Maximum: +6h
* Default: +1h
```

[capability::accelerate_datamodel]

```
* Lets a user enable or disable data model acceleration.
```

[capability::accelerate_search]

```
* Lets a user enable or disable acceleration for reports.
* The assigned role must also be granted the 'schedule_search' capability.
```

[capability::admin_all_objects]

```
* Lets a user access all objects in the system, such as user objects and
  knowledge objects.
* Lets a user bypass any Access Control List (ACL) restrictions, similar
  to the way root access in a *nix environment does.
* the Splunk platform checks this capability when accessing manager pages and objects.
```

[capability::edit_own_objects]

```
* Lets a user edit the knowledge objects or entities for configuration endpoints
  that they own.
```

[capability::list_all_objects]

```
* Lets a user list all configuration settings for the configuration endpoints.
* This capability prevents unauthorized access to configuration endpoints.
```

[capability::list_all_users]

```
* Lets a user list all users by accessing the /services/authentication/users
  REST endpoint.
* For full access to listing users, roles, and capabilities, the user must also
  have or assign the 'list_all_roles' capability.
```

[capability::list_all_roles]

```
* Lets a user list all roles and the capabilities that are assigned to
  those roles.
* For full access to listing users, roles, and capabilities, the user must also
```

```
  have or assign the 'list_all_users' capability.
```

[capability::edit_tokens_settings]

```
* Lets a user access all token auth settings in the system, such as turning the
  the feature on/off and system-wide expiration.
* Splunk checks this capability when accessing manager pages and objects.
```

[capability::change_authentication]

```
* Lets a user change authentication settings through the authentication endpoints.
* Lets the user reload authentication.
```

[capability::change_audit]

```
* Lets a user change audit settings through the audit endpoints.
* Lets a user reload audit settings.
```

[capability::change_own_password]

```
* Lets a user change their own password. You can remove this capability
  to control the password for a user.
```

[capability::list_tokens_scs]

```
* Lets a user retrieve a Splunk Cloud Services (SCS) token for an SCS service with which this
  Splunk Cloud deployment has been configured to communicate.
```

[capability::delete_by_keyword]

```
* Lets a user use the 'delete' command.
* NOTE: The 'delete' command does not actually delete the raw data on disk.
  Instead, it masks the data (via the index) from showing up in search results.
```

[capability::edit_messages]

```
* Lets a user create and delete system messages that appear in the Splunk Web navigation bar.
```

[capability::edit_log_alert_event]

```
* Lets a user log an event when an alert condition is met. Also lets the user
  select the "Log an event" option for an alert action in Splunk Web.
```

[capability::dispatch_rest_to_indexers]

```
* Lets a user dispatch the REST search command to indexers.
```

[capability::edit_authentication_extensions]

```
* Lets a user change the authentication extensions through the
  authentication endpoints.
```

[capability::edit_bookmarks_mc]

```
* Lets a user add bookmark URLs within the Monitoring Console.
```

[capability::edit_deployment_client]

```
* Lets a user edit the deployment client.
* Lets a user edit a deployment client admin endpoint.
```

[capability::edit_deployment_server]

```
* Lets a user edit the deployment server.
* Lets a user edit a deployment server admin endpoint.
* Lets a user change or create remote inputs that are pushed to the
  forwarders and other deployment clients.
```

[capability::list_dist_peer]

```
* Lets a user list/read peers for distributed search.
```

[capability::edit_dist_peer]

```
* Lets a user add and edit peers for distributed search.
* Supercedes list_dist_peer also allows list/read
```

[capability::edit_encryption_key_provider]

```
* Lets a user view and edit keyprovider properties when using
  the Server-Side Encryption (SSE) feature for a remote storage volume.
```

[capability::request_pstacks]

```
* Lets a user trigger pstacks generation of the main splunkd process
  using a REST endpoint.
```

[capability::edit_watchdog]

```
* Lets a user reconfigure watchdog settings using a REST endpoint.
```

[capability::edit_forwarders]

```
* Lets a user edit settings for forwarding data, including settings
  for SSL, backoff schemes, and so on.
* Also used by TCP and Syslog output admin handlers.
```

[capability::edit_health]

```
* Lets a user disable or enable health reporting for a feature in the splunkd
  health status tree through the server/health-config/{feature_name} endpoint.
```

[capability::edit_health_subset]

```
* Lets a user disable or enable health reporting for a feature in the
  "health_subset" view of the health status tree.
* Actions are performed through the server/health-config/{feature_name}
  endpoint.
```

[capability::edit_httpauths]

```
* Lets a user edit and end user sessions through the httpauth-tokens endpoint.
```

[capability::edit_indexer_cluster]

```
* Lets a user edit or manage indexer clusters.
```

[capability::edit_indexerdiscovery]

```
* Lets a user edit settings for indexer discovery, including settings
  for master_uri, pass4SymmKey, and so on.
* Also used by Indexer Discovery admin handlers.
```

[capability::edit_ingest_rulesets]

```
* Lets a user add, edit, and delete ingest action rule sets
  through the data/ingest/rulesets endpoint.
```

[capability::edit_input_defaults]

```
* Lets a user change the default hostname for input data through the server
  settings endpoint.
```

[capability::edit_local_apps]

```
* Lets a user edit apps on the local Splunk instance through the
  local apps endpoint.
* For full access to app management, also add the 'install_apps'
  capability to the role.
```

```
* To enable enforcement of the "install_apps" capability, see the
  "enable_install_apps" setting in limits.conf.
```

[capability::edit_monitor]

```
* Lets a user add inputs and edit settings for monitoring files.
* Also used by the standard inputs endpoint as well as the oneshot input
  endpoint.
```

[capability::edit_modinput_journald]

```
* Lets the user add and edit journald inputs.
* This input is not available on Windows.
```

[capability::edit_modinput_winhostmon]

```
* Lets a user add and edit inputs for monitoring Windows host data.
```

[capability::edit_modinput_winnetmon]

```
* Lets a user add and edit inputs for monitoring Windows network data.
```

[capability::edit_modinput_winprintmon]

```
* Lets a user add and edit inputs for monitoring Windows printer data.
```

[capability::edit_modinput_perfmon]

```
* Lets a user add and edit inputs for monitoring Windows performance.
```

[capability::edit_modinput_admon]

```
* Lets a user add and edit inputs for monitoring Active Directory (AD).
```

[capability::edit_roles]

```
* Lets a user edit roles.
* Lets a user change the mappings from users to roles.
* Used by both user and role endpoints.
```

[capability::edit_roles_grantable]

```
* Lets a user edit roles and change user-to-role mappings for a limited
  set of roles.
* To limit this ability, also assign the 'edit_roles_grantable' capability
  and configure the 'grantableRoles' setting in authorize.conf.
    * For example:
        grantableRoles = role1;role2;role3
```

```
        This configuration lets a user create roles using the subset of
        capabilities that the user has in their 'grantable_roles' setting.
```

[capability::edit_scripted]

```
* Lets a user create and edit scripted inputs.
```

[capability::edit_search_head_clustering]

```
* Lets a user edit and manage search head clustering.
```

[capability::edit_search_concurrency_all]

```
* Lets a user edit settings related to maximum concurrency of searches.
```

[capability::edit_search_concurrency_scheduled]

```
* Lets a user edit settings related to concurrency of scheduled searches.
```

[capability::edit_search_scheduler]

```
* Lets a user disable and enable the search scheduler.
```

[capability::edit_search_schedule_priority]

```
* Lets a user assign a search a higher-than-normal schedule priority.
```

[capability::edit_search_schedule_window]

```
* Lets a user edit a search schedule window.
* Requires the 'schedule_search' capability.
* For more about the search scheduler, see the Knowledge Manager Manual.
```

[capability::edit_search_server]

```
* Lets a user edit general distributed search settings like timeouts,
  heartbeats, and deny lists.
```

[capability::edit_server]

```
* Lets a user edit general server and introspection settings, such
  as the server name, log levels, and so on.
* This capability also inherits the ability to read general server
  and introspection settings.
```

[capability::edit_server_crl]

```
* Lets a user reload Certificate Revocation Lists (CRLs) within Splunk.
* A CRL is a list of digital certificates that have been revoked by the
  issuing certificate authority (CA) before their scheduled expiration
  date and should no longer be trusted.
```

[capability::edit_sourcetypes]

```
* Lets a user create and edit sourcetypes.
```

[capability::edit_splunktcp]

```
* Lets a user change settings for receiving TCP input from another Splunk
  instance.
```

[capability::edit_splunktcp_ssl]

```
* Lets a user view and edit SSL-specific settings for Splunk TCP input.
```

[capability::edit_user_seed]

```
* Lets a user view and edit the user-seed.conf file used for initial username
  and password configration.
```

[capability::edit_splunktcp_token]

```
* Lets a user view or edit splunktcptokens. The tokens can be used on a
  receiving system to only accept data from forwarders that have been
  configured with the same token.
```

[capability::edit_storage_passwords]

```
* Lets a user read from (GET) and write to (POST) the /storage/passwords endpoint.
```

[capability::edit_tcp]

```
* Lets a user change settings for receiving general TCP inputs.
```

[capability::edit_telemetry_settings]

```
* Lets a user change settings for opting in and sending telemetry data.
```

[capability::edit_token_http]

```
* Lets a user create, edit, display, and remove settings for HTTP token input.
```

```
* Enables the HTTP Events Collector feature, which is a way to send data to
  Splunk Enterprise and Splunk Cloud.
```

[capability::edit_tokens_all]

```
* Lets a user issue tokens to all users.
```

[capability::edit_tokens_own]

```
* Lets a user issue tokens to themself.
```

[capability::edit_udp]

```
* Lets a user change settings for UDP inputs.
```

[capability::edit_user]

```
* Lets a user create, edit, or remove other users.
* Also lets a user manage certificates for distributed search.
* To edit the roles of a user, you must hold roles whose combined capabilities
  either match or exceed the capabilities of the roles that you want to edit
  for the user.
* To let users grant additional roles, assign the
  'edit_roles_grantable' capability and configure the
  'grantableRoles' setting in authorize.conf.
    * Example: grantableRoles = role1;role2;role3
```

[capability::edit_view_html]

```
* Lets a user create, edit, or otherwise modify HTML-based views.
```

[capability::edit_web_settings]

```
* Lets a user change the settings for web.conf through the system settings
  endpoint.
```

[capability::export_results_is_visible]

```
* Lets a user show or hide the Export button in Splunk Web.
* Disable this setting to hide the Export button and prevent users with
  this role from exporting search results.
```

[capability::get_diag]

```
* Lets the user generate a diag on a remote instance through the
  /streams/diag endpoint.
```

[capability::get_metadata]

```
* Lets a user use the metadata search processor.
```

[capability::get_typeahead]

```
* Enables typeahead for a user, both the typeahead endpoint and the
  'typeahead' search processor.
```

[capability::indexes_edit]

```
* Lets a user change any index settings such as file size and memory limits.
```

[capability::input_file]

```
* Lets a user add a file as an input through the inputcsv command (except for
  dispatch=t mode) and the inputlookup command.
```

[capability::install_apps]

```
* Lets a user install, uninstall, create, and update apps on the local
  Splunk platform instance through the apps/local endpoint.
* For full access to app management, also add the 'edit_local_apps'
  capability to the role.
* To enable enforcement of the "install_apps" capability, see the
  "enable_install_apps" setting in limits.conf.
```

[capability::license_tab]

```
* DEPRECATED.
* Lets a user access and change the license.
* Replaced with the 'license_edit' capability.
```

[capability::license_edit]

```
* Users with this capability can access and change license attributes and related information.
```

[capability::license_read]

```
* Users with this capability can access license attributes and related information.
```

[capability::license_view_warnings]

```
* Lets a user see if they are exceeding limits or reaching the expiration
  date of their license.
* License warnings are displayed on the system banner.
```

[capability::list_accelerate_search]

```
* This capability is a subset of the 'accelerate_search' capability.
* This capability grants access to the summaries that are required to run accelerated reports.
* Users with this capability, but without the 'accelerate_search' capability, can run,
  but not create, accelerated reports.
```

[capability::list_deployment_client]

```
* Lets a user list the deployment clients.
```

[capability::list_deployment_server]

```
* Lets a user list the deployment servers.
```

[capability::list_pipeline_sets]

```
* Lets a user list information about pipeline sets.
```

[capability::list_forwarders]

```
* Lets a user list settings for data forwarding.
* Used by TCP and Syslog output admin handlers.
```

[capability::list_health]

```
* Lets a user monitor the health of various Splunk features
  (such as inputs, outputs, clustering, and so on) through REST endpoints.
```

[capability::list_health_subset]

```
* Lets a user monitor the health of a subset of Splunk features (such as search
  scheduler) through REST endpoints.
* These features are more oriented towards the end user, rather than the Splunk
  administrator.
```

[capability::list_httpauths]

```
* Lets a user list user sessions through the httpauth-tokens endpoint.
```

[capability::list_indexer_cluster]

```
* Lets a user list indexer cluster objects such as buckets, peers, and so on.
```

[capability::list_indexerdiscovery]

```
* Lets a user view settings for indexer discovery.
* Used by indexer discovery handlers.
```

[capability::list_ingest_rulesets]

```
* Lets a user view the list of ingest action rule sets
  through the data/ingest/rulesets endpoint.
```

[capability::list_inputs]

```
* Lets a user view the list of inputs including files, TCP, UDP, scripts, and so on.
```

[capability::list_introspection]

```
* Lets a user read introspection settings and statistics for indexers, search,
  processors, queues, and so on.
```

[capability::list_search_head_clustering]

```
* Lets a user list search head clustering objects such as artifacts, delegated
  jobs, members, captain, and so on.
```

[capability::list_search_scheduler]

```
* Lets a user list search scheduler settings.
```

[capability::list_settings]

```
* Lets a user list general server and introspection settings such as the server
  name and log levels.
```

[capability::list_metrics_catalog]

```
* Lets a user list metrics catalog information such as the metric names,
  dimensions, and dimension values.
```

[capability::edit_metrics_rollup]

```
* Lets a user create/edit metrics rollup defined on metric indexes.
```

[capability::list_storage_passwords]

```
* Lets a user read from (GET) the /storage/passwords endpoint.
* You must add the 'edit_storage_passwords' capability to the role for the user to
```

```
  perform POST operations to the /storage/passwords endpoint.
```

[capability::list_token_http]

```
* Lets a user display settings for HTTP token input.
```

[capability::list_tokens_all]

```
* Lets a user view all tokens.
```

[capability::list_tokens_own]

```
* Lets a user view their own tokens.
```

[capability::never_lockout]

```
* Allows a user's account to never lockout.
```

[capability::never_expire]

```
* Allows a user's account to never expire.
```

[capability::output_file]

```
* Lets a user create file outputs, including the 'outputcsv' command (except for
  dispatch=t mode) and the 'outputlookup' command.
```

[capability::pattern_detect]

```
* Controls ability to see and use the Patterns tab in the Search view.
```

[capability::request_remote_tok]

```
* Lets a user get a remote authentication token.
* Used for distributing search to old 4.0.x Splunk instances.
* Also used for some distributed peer management and bundle replication.
```

[capability::rest_apps_management]

```
* Lets a user edit settings for entries and categories in the Python remote
  apps handler.
* See restmap.conf.spec for more information.
```

[capability::rest_apps_view]

```
* Lets a user list various properties in the Python remote apps handler.
```

* See restmap.conf.spec for more info

[capability::rest_properties_get]

* Lets a user get information from the services/properties endpoint.

[capability::rest_properties_set]

* Lets a user edit the services/properties endpoint.

[capability::restart_splunkd]

* Lets a user restart the Splunk platform through the server control handler.

[capability::rtsearch]

* Lets a user run real-time searches.

[capability::run_collect]

* Lets a user run the 'collect' command.

[capability::run_dump]

* Lets a user run the 'dump' command.

[capability::run_custom_command]

* Lets a user run custom search commands.

[capability::run_mcollect]

* Lets a user run the 'mcollect' and 'meventcollect' commands.

[capability::run_msearch]

* Lets a user run the 'mpreview' and 'msearch' commands.

[capability::rest_access_server_endpoints]

* Lets a user run the 'rest' command and access 'services/server/' endpoints.

[capability::run_sendalert]

```
* Lets a user run the 'sendalert' command.
```

[capability::run_debug_commands]

```
* Lets a user run debugging commands, for example 'summarize'.
```

[capability::run_walklex]

```
* Lets a user run the 'walklex' command even if they have a role with a search filter.
```

[capability::run_commands_ignoring_field_filter]

```
* Lets a user run commands that return index information even when a
  'fieldFilter' is configured for that user's role.
* Some commands can return sensitive index information to which a role
  with a 'fieldFilter' should not have access.
* The following commands require this capability for roles configured with a
  'fieldFilter':  walklex, typeahead, tstats, mstats, mpreview.
```

[capability::schedule_rtsearch]

```
* Lets a user schedule real-time saved searches.
* You must enable the 'scheduled_search' and 'rtsearch' capabilities for the role.
```

[capability::schedule_search]

```
* Lets a user schedule saved searches, create and update alerts, and
  review triggered alert information.
```

[capability::metric_alerts]

```
* Lets a user create and update the new metric alerts.
```

[capability::search]

```
* Lets a user run a search.
```

[capability::search_process_config_refresh]

```
* Lets a user manually flush idle search processes through the
  'refresh search-process-config' CLI command.
```

[capability::use_file_operator]

```
* Lets a user use the 'file' command.
* The 'file' command is DEPRECATED.
```

[capability::upload_lookup_files]

```
* Lets a user upload files which can be used in conjunction with lookup definitions.
```

[capability::upload_mmdb_files]

```
* Lets a user upload mmdb files, which are used for iplocation searches.
```

[capability::web_debug]

```
* Lets a user access /_bump and /debug/** web debug endpoints.
```

[capability::edit_field_filter]

```
* Lets a user use an API to update role-based 'fieldFilter' configurations.
```

[capability::edit_statsd_transforms]

```
* Lets a user define regular expressions to extract manipulated dimensions out of
  metric_name fields in statsd metric data using the
  services/data/transforms/statsdextractions endpoint.
* For example, dimensions can be mashed inside a metric_name field like
  "dimension1.metric_name1.dimension2" and you can use regular expressions to extract it.
```

[capability::edit_metric_schema]

```
* Lets a user define the schema of the log data that must be converted
  to metric format using the services/data/metric-transforms/schema endpoint.
```

[capability::list_workload_pools]

```
* Lets a user list and view workload pool and workload status information through
  the workloads endpoint.
```

[capability::edit_workload_pools]

```
* Lets a user create and edit workload pool and workload configuration information
  (except workload rule) through the workloads endpoint.
```

[capability::select_workload_pools]

```
* Lets a user select a workload pool for a scheduled or ad-hoc search.
```

[capability::list_workload_rules]

```
* Lets a user list and view workload rule information from the workloads/rules
  endpoint.
```

[capability::edit_workload_rules]

```
* Lets a user create and edit workload rules through the workloads/rules endpoint.
```

[capability::list_workload_policy]

```
* Lets a user view workload_policy.conf file settings through the workloads/policy endpoint.
* For now, it is used to view 'admission_rules_enabled' setting under
  stanza [search_admission_control].
* admission_rules_enabled = 1 means the admission rules are enabled in
  [[/manager/system/workload_management|Admission Rules UI]]
```

[capability::edit_workload_policy]

```
* Lets a user edit workload_policy.conf file settings through the workloads/policy endpoint.
* For now, it used to change 'admission_rules_enabled' setting under
  stanza [search_admission_control].
* admission_rules_enabled = 1 means the admission rules are enabled in
  [[/manager/system/workload_management|Admission Rules UI]]
```

[capability::apps_restore]

```
* Lets a user restore configurations from a backup archive through
  the apps/restore endpoint.
```

[capability::edit_global_banner]

```
* Lets a user enable/edit a global banner visible to all users on every page.
```

[capability::edit_kvstore]

```
* Lets a user execute KV Store administrative commands through the KV Store REST endpoints.
```

[capability::list_cascading_plans]

```
* Lets a user view the generated knowledge bundle replication plans if the chosen replication
  policy in distsearch.conf is set to 'cascading'.
```

[capability::list_remote_output_queue]

```
* Lets a user view the configuration details of a configured remote output queue for Splunk Cloud
  and Splunk Cloud Services(SCS) instances.
```

[capability::list_remote_input_queue]

```
* Lets a user view the configuration details of a configured remote input queue for Splunk Cloud
  and Splunk Cloud Services(SCS) instances.
```

[capability::edit_manager_xml]

```
* Lets a user create and edit XML views using the /data/ui/manager REST endpoint.
```

[capability::merge_buckets]

```
* Lets a user merge buckets using cluster-merge-buckets CLI for clustered environments
```

[capability::read_internal_libraries_settings]

```
* Lets a user read the 'quarantined/status' REST endpoint and also view
  the Internal Libraries Settings page in Splunk Web.
```

[capability::edit_web_features]

```
* Lets a user write to the '/web-features' REST endpoint.
```

authorize.conf.example

```
#   Version 9.2.2
#
# This is an example authorize.conf.  Use this file to configure roles and
# capabilities.
#
# To use one or more of these configurations, copy the configuration block
# into authorize.conf in $SPLUNK_HOME/etc/system/local/.  You must reload
# auth or restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[role_ninja]
rtsearch = enabled
importRoles = user
srchFilter = host=foo
srchIndexesAllowed = *
srchIndexesDefault = mail;main
```

```
srchJobsQuota   = 8
rtSrchJobsQuota = 8
srchDiskQuota   = 500
srchTimeWin = 86400
srchTimeEarliest = 2592000

# This creates the role 'ninja', which inherits capabilities from the 'user'
# role.  ninja has almost the same capabilities as power, except cannot
# schedule searches.
#
# The search filter limits ninja to searching on host=foo.
#
# ninja is allowed to search all public indexes (those that do not start
# with underscore), and will search the indexes mail and main if no index is
# specified in the search.
#
# ninja is allowed to run 8 search jobs and 8 real time search jobs
# concurrently (these counts are independent).
#
# ninja is allowed to take up 500 megabytes total on disk for all their jobs.
#
# ninja is allowed to run searches that span a maximum of one day
#
# ninja is allowed to run searches on data that is newer than 30 days ago
```

bookmarks.conf

The following are the spec and example files for `bookmarks.conf`.

bookmarks.conf.spec

```
#   Version 9.2.2
#
# This file contains possible settings and values for configuring various
# "bookmark" entries to be stored within a Splunk instance.
#
# To add custom bookmarks, place a bookmarks.conf file in
# $SPLUNK_HOME/etc/system/local/ on the Splunk instance.
# configuration content is deployed to a
# given deployment client in serverclass.conf.  Refer to
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[bookmarks_mc:]*

```
url = <string>
* A bookmark URL that redirects logged-in administrators to other Monitoring
  Console instances that may be within their purview. Set this up if you have
  administrators who are responsible for the performance and uptime of multiple
  Splunk deployments.
* The bookmark appears in the left pane of the Monitoring Console.
* The URL must begin with http:// or https:// and contain 'splunk_monitoring_console'.
* Default: not set
```

bookmarks.conf.example

```
#   Version 9.2.2


# Example: Monitoring console
# User is administrator of 3 Splunk deployments: US security, Global Security,
# and US Applications, and wants convenient access to the monitoring console
# for each.

[bookmarks_mc:US Security]
url =
https://us-security.testcorp.example:8000/en-US/app/splunk_monitoring_console/monitoringconsole_overview

[bookmarks_mc:Global Security]
url =
https://global-security.testcorp.example:8000/en-US/app/splunk_monitoring_console/monitoringconsole_overview

[bookmarks_mc:US Applications]
url =
http://us-applications.testcorp.example:8000/en-US/app/splunk_monitoring_console/monitoringconsole_overview
```

checklist.conf

The following are the spec and example files for `checklist.conf`.

checklist.conf.spec

```
#   Version 9.2.2
#
# This file contains the set of attributes and values you can use to
# configure checklist.conf to run health checks in Monitoring Console.
# Any health checks you add manually should be stored in your app's local directory.
#
```

[<uniq-check-item-name>]

```
* A unique string for the name of this health check.

title = <ASCII string>
* (required) Displayed title for this health check.

category = <ASCII string>
* (required) Category for overarching goups of health check items.

tags = <ASCII string>
* (optional) Comma separated list of tags that apply to this health check.
* If omitted user will not be able to run this health check as part of a subset of health checks.

description = <ASCII string>
* (optional) A description of what this health check is checking.
* If omitted no description will be displayed.

failure_text = <ASCII string>
* If this health check did not pass, the text that you specify in this setting can
```

```
  explain what went wrong.
* While this setting is optional, if you do not specify a value for this
  setting, this health check does not display any text that helps
  identify why it did not pass.

suggested_action = <ASCII string>
* (optional) Suggested actions for diagnosing and fixing your Splunk installation
  so this health check is no longer failing.
* If omitted no suggested actions for fixing this health check will be displayed.

doc_link = <ASCII string>
* (optional) Location string for help documentation for this health check.
* If omitted no help link will be displayed to help the user fix this health check.
* Can be a comma separated list if more than one documentation link is needed.

doc_title = <ASCII string>
* (optional) Title string for help documentation link for this health check.
* Must be included if doc_link exists.
* Will be inserted in the text for the help documentation link like so: "Learn more about $doc_title$"
* If doc_link is a comma separated list,
*   then doc_title must also be a comma separated list with one title per item corresponding to doc_link.

applicable_to_groups = <ASCII string>
* (optional) Comma separated list of applicable groups that this check should be run against.
* If omitted this check item can be applied to all groups.

environments_to_exclude = <ASCII string>
* (optional) Comma separated list of environments that the health check should not run in.
*   Possible environments are 'standalone' and 'distributed'
* If omitted this check can be applied to all groups.

disabled = <boolean>
* Disable this check item by setting to 1.
* Default: 0

search = <ASCII string>
* (required) Search string to be run to perform the health check.
* Please separate lines by "\" if the search string has multiple lines.
*
* In single-instance mode, this search will be used to generate the final result.
* In multi-instance mode, this search will generate one row per instance in the result table.
*
* THE SEARCH RESULT NEEDS TO BE IN THE FOLLOWING FORMAT:
* |-------------------------------------------------------------
* |     instance    |            metric         | severity_level |
* |-------------------------------------------------------------
* | <instance name> | <metric number or string> | <level number> |
* |-------------------------------------------------------------
* |       ...       |              ...          |      ...       |
* |-------------------------------------------------------------
*
* <instance name> (required, unique) is either the "host" field of events or the
  "splunk_server" field of "| rest" search.
*   In order to generate this field, please do things like:
*     ... | rename host as instance
*   or
*     ... | rename splunk_server as instance
*
* <metric number or string> (optional) one ore more columns to "show your work"
*   This should be the data that severity_level is determined from.
*   The user should be able to look at this field to get some idea of what made the instance fail this
check.
```

```
*
* <level number> (required) could be one of the following:
*   - -1 (N/A) means: "Not Applicable"
*   - 0 (ok) means: "all good"
*   - 1 (info) means: "just ignore it if you don't understand"
*   - 2 (warning) means: "well, you'd better take a look"
*   - 3 (error) means: "FIRE!"
*
* Please also note that the search string must contain either of the following
  token to properly scope to either a single instance or a group of instances,
  depending on the settings of checklistsettings.conf.
    $rest_scope$ - used for "|rest" search
    $hist_scope$ - used for historical search

drilldown = <ASCII string>
* (optional) Link to a search or Monitoring Console dashboard for additional information.
* Please note that the drilldown string must contain a $ delimited string.
  * This string must match one of the fields output by the search.
  * Most dashboards will need the name of the instance, eg $instance$
```

checklist.conf.example

```
No example
```

collections.conf

The following are the spec and example files for `collections.conf`.

collections.conf.spec

```
#   Version 9.2.2
#
# This file configures the KV Store collections for a given app in Splunk.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[<collection-name>]

```
enforceTypes = <boolean>
* Indicates whether to enforce data types when inserting data into the
  collection.
* When set to true, invalid insert operations fail.
* When set to false, invalid insert operations drop only the invalid field.
* Default: false

field.<name> = number|bool|string|time
* Field type for a field called <name>.
* If the data type is not provided, the data type is inferred from the provided JSON
  data type.

accelerated_fields.<name> = <json>
* Acceleration definition for an acceleration called <name>.
* Must be a valid JSON document. Invalid JSON is ignored.
* Example: 'acceleration.foo={"a":1, "b":-1}' is a compound acceleration
```

```
  that first sorts 'a' in ascending order and then 'b' in descending order.
* There are restrictions in compound acceleration. A compound acceleration
  must not have more than one field in an array. If it does, KV store does
  not start or work correctly.
* Duplicating fields in KV store acceleration definitions might cause KV store to fail.
* If multiple accelerations with the same definition are in the same
  collection, the duplicates are skipped.
* If the data within a field is too large for acceleration, you see a
  warning when you try to create an accelerated field and the acceleration
  is not created.
* An acceleration is always created on the _key.
* The order of accelerations is important. For example, an acceleration of
  { "a":1, "b":1 } speeds queries on "a" and "a" + "b", but not on "b"
  alone.
* Multiple separate accelerations also speed up queries. For example,
  separate accelerations { "a": 1 } and { "b": 1 } speed up queries on
  "a" + "b", but not as well as a combined acceleration { "a":1, "b":1 }.
* Default: nothing (no acceleration)

profilingEnabled = <boolean>
* Indicates whether to enable logging of slow-running operations, as defined
  in 'profilingThresholdMs'.
* Default: false

profilingThresholdMs = <zero or positive integer>
* The threshold for logging a slow-running operation, in milliseconds.
* When set to 0, all operations are logged.
* This setting is used only when 'profilingEnabled' is "true".
* This setting affects the performance of the collection.
* Default: 1000

replicate = <boolean>
* Indicates whether to replicate this collection on indexers. When false,
  this collection is not replicated on indexers, and lookups that depend on
  this collection are not available (although if you run a lookup command
  with 'local=true', local lookups are available). When true,
  this collection is replicated on indexers.
* Default: false

replication_dump_strategy = one_file|auto
* Indicates how to store dump files. When set to one_file, dump files are
  stored in a single file. When set to auto, dump files are stored in
  multiple files when the size of the collection exceeds the value of
  'replication_dump_maximum_file_size'.
* Default: auto

replication_dump_maximum_file_size = <unsigned integer>
* Specifies the maximum file size (in KB) for each dump file when
  'replication_dump_strategy=auto'.
* If this value is larger than the value of 'concerningReplicatedFileSize'
  in distsearch.conf, the value of 'concerningReplicatedFileSize' is
  used instead.
* KV Store does not pre-calculate the size of the records to be written
  to disk, so the size of the resulting files can be affected by the
  'max_rows_in_memory_per_dump' setting from limits.conf.
* Default: 10240

type = internal_cache|undefined
* For internal use only.
* Indicates the type of data that this collection holds.
* When set to internal_cache, changing the configuration of the current
  instance between search head cluster, search head pool, or standalone
```

```
  erases the data in the collection.
* Default: undefined
```

collections.conf.example

```
#   Version 9.2.2
#
# The following is an example collections.conf configuration.
#
# To use one or more of these configurations, copy the configuration block
# into collections.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
# Note this example uses a compound acceleration. Please check collections.conf.spec
# for restrictions on compound acceleration.

[mycollection]

field.foo = number
field.bar = string
accelerated_fields.myacceleration = {"foo": 1, "bar": -1}
```

commands.conf

The following are the spec and example files for `commands.conf`.

commands.conf.spec

```
#   Version 9.2.2
```

OVERVIEW

```
# This file contains descriptions for the setting/value pairs that you can
# use for creating search commands for custom search scripts.
#
# If you add your custom search script to the $SPLUNK_HOME/etc/apps/MY_APP/bin/
# path, put a custom commands.conf file in the
# $SPLUNK_HOME/etc/apps/MY_APP/default/ directory.
#
# There is a commands.conf in $SPLUNK_HOME/etc/system/default/.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name commands.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see commands.conf.example.  You must restart the Splunk platform
# to enable configurations.
#
# To learn more about configuration files (including file precedence) see the
```

```
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file wins.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

[<STANZA_NAME>]

```
* Each stanza represents a search command. The command name is the stanza name.
* The stanza name invokes the command in the search language.
* Specify the following settings/values for the command.  Otherwise, the
  default values are used.
* If the 'filename' setting is not specified, an external program is searched for
  by appending extensions (e.g. ".py", ".pl") to the stanza name.
* If the `chunked` setting is set to "true", in addition to the extensions ".py"
  and ".pl" as above, the extensions ".exe", ".bat", ".cmd", ".sh", ".js", as
  well as no extension (to find binaries without extensions), are searched for.
* See the 'filename' setting for more information about how external programs
  are searched for.

type = <string>
* The type of script. Valid values are python and perl.
* Default: python

python.version = {default|python|python2|python3}
* For Python scripts only, specifies which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.

filename = <string>
* Optionally specify the program to run when the custom search command is used.
* The 'filename' is looked for in the `bin` directory for the app.
* The 'filename' setting cannot reference any file outside of the `bin` directory
  for the app.
* If the 'filename' ends in ".py", the python interpreter is used
  to invoke the external script.
* If the 'chunked' setting is set to "true", the 'filename' is looked for first in the
  $SPLUNK_HOME/etc/apps/MY_APP/<PLATFORM>/bin directory before searching the
  $SPLUNK_HOME/etc/apps/MY_APP/bin directory. The <PLATFORM> is one of the following:
  "linux_x86_64"
  "linux_x86"
  "windows_x86_64"
  "windows_x86"
  "darwin_x86_64"
  Depending on the platform that the Splunk software is running on.
* If the 'chunked' setting is set to "true" and if a path pointer file (*.path)
```

```
  is specified, the contents of the path pointer file are read and the result is
  used as the command to run. Environment variables in the path pointer
  file are substituted. You can use path pointer files to reference
  system binaries. For example: /usr/bin/python.

command.arg.<N> = <string>
* Additional command-line arguments to use when invoking this
  program. Environment variables, such as $SPLUNK_HOME, are substituted.
* Only available if the `chunked` setting is "true".

local = <boolean>
* If set to "true", specifies that the command should be run on the search head only.
* Default: false

perf_warn_limit = <integer>
* Issue a performance warning message if more than the value specified for input events are
  passed to this external command (0 = never)
* Default: 0 (disabled)

streaming = <boolean>
* Whether or not the command is streamable.
* Default: false

maxinputs = <integer>
* The maximum number of events that can be passed to the command for each
  invocation.
* This limit cannot exceed the value of the 'maxresultrows' setting in limits.conf file.
* Specify 0 for no limit.
* Default: 50000

passauth = <boolean>
* Whether or not the Splunk platform passes authentication-related facts
  at the start of input, as part of the header.
* See the 'enableheader' setting for additional information on headers.
* If set to "true", splunkd passes several authentication-related facts
  at the start of input, as part of the header.
* The Splunk platform passes the following headers:
  * authString: A pseudo-xml string that resembles
      <auth><userId>username</userId><username>username</username><authToken>auth_token< /authToken></auth>
    where the username is passed twice, and the authToken can be used
    to contact splunkd during the script run.
  * sessionKey: the session key again
  * owner: the user portion of the search context
  * namespace: the app portion of the search context
* Requires "enableheader = true". If "enableheader = false", the Splunk platform
  also treats this setting as "false".
* If "chunked = true", the Splunk platform ignores this setting. It always passes
  an authentication token to commands using the chunked custom search
  command protocol.
* Default: false

run_in_preview = <boolean>
* Determines whether to run a custom search command when it is generating
  results just for preview rather than for final output.
* A setting of 'false' means that the custom search command does not run during
  preview.
* This setting defaults to 'false' for commands that use 'chunked=true'. Custom
  search commands that run with 'chunked=true' can have performance issues when
  they also run in preview.
* There is no global default for this setting that would apply to all search
  commands.
  * If you have a custom search command that must deviate from the default
```

```
    behavior described here, set this setting for that command.
* Default: 'false' when 'chunked=true', 'true' otherwise.

enableheader = <boolean>
* Whether or not your script expects header information.
* If set to "true" it will expect as input a head section + '\n' then the CSV input.
* NOTE: Should be set to "true" if you use splunk.Intersplunk
* Default: true

retainsevents = <boolean>
* Whether or not the command retains events, the way that the sort/dedup/cluster
  commands do, or whether the command transforms events, the way that the stats
  command does.
* Default: false

generating = <boolean>
* Whether or not your command generates new events. If no events are passed to
  the command, will it generate events?
* Default: false

generates_timeorder = <boolean>
* If "generating = true", does the command generate events in descending time order,
  with the latest event first.
* Default: false

overrides_timeorder = <boolean>
* If "generating = false" and "streaming = true", does the command change the order of
  events with respect to time?
* Default: false

requires_preop = <boolean>
* Whether or not the command sequence specified by the 'streaming_preop' setting
  is required for proper execution or is it an optimization only.
* Default: false (streaming_preop not required)

streaming_preop = <string>
* A string that denotes the requested pre-streaming search string.

required_fields = <string>
* A comma-separated list of fields that this command can use.
* Informs previous commands that they should retain/extract these fields if
  possible.  No error is generated if a field specified is missing.
  The default is all fields.
* Default: '*'

supports_multivalues = <boolean>
* Whether or not the command supports multiple values.
* If set to "true", multivalues are treated as python lists of strings, instead of a
  flat string (when using Intersplunk to interpret stdin/stdout).
* If the list only contains one element, the value of that element is
  returned, rather than a list. For example:
    isinstance(val, basestring) == True

supports_getinfo = <boolean>
* Whether or not the command supports dynamic probing for settings
  (first argument invoked == __GETINFO__ or __EXECUTE__).

supports_rawargs = <boolean>
* If set to "true", specifies that the command supports raw arguments being passed to it.
* If set to "false", specifies that the command prefers parsed arguments,
  where quotes are stripped.
* Default: false
```

```
undo_scheduler_escaping = <boolean>
* Whether or not or not the raw arguments of a command should have any
  previously-applied escaping removed.
* This setting applies in particular to commands that the scheduler invokes,
  and only if the commands support raw arguments, where the 'supports_rawargs'
  setting for the command is "true".
* Default: false

requires_srinfo = <boolean>
* Specifies if the command requires information stored in SearchResultsInfo.
* If set to "true", requires that 'enableheader' is set to "true", and the full
  pathname of the info file (a csv file) will be emitted in the header under
  the key 'infoPath'.
* Default: false

needs_empty_results = <boolean>
* Whether or not this custom search command needs to be called with
  intermediate empty search results.
* Default: true

changes_colorder = <boolean>
* Whether or not the script output should be used to change the column
  ordering of the fields.
* Default: true

outputheader = <boolean>
* If set to "true", output of script should be a header section + blank
  line + csv output.
* If set to "false", the script output should be pure comma separated values only.
* Default: false

clear_required_fields = <boolean>
* If set to "true", 'required_fields' represents the *only* fields required.
* If set to "false", 'required_fields' are additive to any fields that might be
  required by subsequent commands.
* In most cases, "false" is appropriate for streaming commands and "true" for
  transforming commands.
* Default: false

stderr_dest = [log|message|none]
*  Specifies what do to with the stderr output from the script.
* 'log' means to write the output to the job search.log file.
* 'message' means to write each line as a search info message. The message
  level can be set to adding that level (in ALL CAPS) to the start of the
  line.For example, "WARN my warning message."
* 'none' means to discard the stderr output.
* Default: log

is_order_sensitive = <boolean>
* Set to "true" if the command requires the input to be in order.
* Default: false

is_risky = <boolean>
* Searches using Splunk Web are flagged to warn users when they
  unknowingly run a search that contains commands that might be a
  security risk. This warning appears when users click a link or type
  a URL that loads a search that contains risky commands. This warning
  does not appear when users create ad hoc searches.
* This flag is used to determine whether the command is risky.
* NOTE: Specific commands that ship with the product have their own
  default setting for 'is_risky'.
```

```
* Default: false

chunked = <boolean>
* Whether or not the search command supports the new "chunked" custom search
  command protocol.
* If set to "true", this command supports the new "chunked" custom
  search command protocol, and only the following commands.conf settings are valid:
  * 'is_risky'
  * 'maxwait'
  * 'maxchunksize'
  * 'filename'
  * 'command.arg.<N>'
  * 'python.version', and
  * 'run_in_preview'.
* If set to "false", this command uses the legacy custom search command
  protocol supported by Intersplunk.py.
* Default: false

pass_timezone = <boolean>
* Specify whether or not splunkd passes the serialized timezone information
  of the user to the script as part of the header. The serialized timezone
  information can be used to convert time to match the user's timezone.
* If set to "true", when an alert action generates a PDF file, the user's
  timezone is used when rendering the charts in the PDF.
* Valid only when 'enableheader' is set to "true". If 'enableheader' is set to "false",
  'pass_timezone' is set "false" as well.
* Default: false

maxwait = <integer>
* The maximum amount of time, in seconds, that the custom search command can
  pause before producing output.
* Only available if "chunked = true".
* Not supported on Windows.
* If set to "0", the command can pause forever.
* Default: 0

maxchunksize = <integer>
* The maximum chunk size, including the size of metadata plus the size of body,
  that the external command can produce. If the command
  tries to produce a larger chunk, the command is terminated.
* Only available if "chunked = true".
* If set to "0", the command can send any size chunk.
* Default: 0
```

commands.conf.example

```
#   Version 9.2.2
#
# This is an example commands.conf.  Use this file to configure settings
# for external search commands.
#
# To use one or more of these configurations, copy the configuration block
# into commands.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence)
# see the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

```
# Note: These are examples.  Replace the values with your own
# customizations.


##############
# defaults for all external commands, exceptions are below in
# individual stanzas

# type of script: 'python', 'perl'
TYPE = python
# default "filename" would be <stanza-name>.py for python,
# <stanza-name>.pl for perl, and
# <stanza-name> otherwise

# is command streamable?
streaming = false

# maximum data that can be passed to command (0 = no limit)
maxinputs = 50000

# end defaults
#####################

[createrss]
filename = createrss.py

[diff]
filename = diff.py

[runshellscript]
filename = runshellscript.py

[sendemail]
filename = sendemail.py

[uniq]
filename = uniq.py

[windbag]
filename = windbag.py
supports_multivalues = true

[xmlkv]
filename = xmlkv.py

[xmlunescape]
filename = xmlunescape.py
```

datamodels.conf

The following are the spec and example files for `datamodels.conf`.

datamodels.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attribute/value pairs for configuring
# data models.  To configure a datamodel for an app, put your custom
# datamodels.conf in $SPLUNK_HOME/etc/apps/MY_APP/local/
```

```
# For examples, see datamodels.conf.example.  You must restart Splunk to
# enable configurations.

# To learn more about configuration files (including precedence) see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top
#     of the file.
#   * Each conf file should have, at most, one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
#     file wins.
#   * If an attribute is defined at both the global level, and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

[<datamodel_name>]

```
* Each stanza represents a data model. The data model name is the stanza name.

acceleration = <boolean>
* Whether or not the Splunk platform automatically accelerates this data model.
* Automatic acceleration creates auxiliary column stores for the fields
  and values in the events for this data model on a per-bucket basis.
* These column stores take additional space on disk, so be sure you have the
  proper amount of disk space. Additional space required depends on the
  number of events, fields, and distinct field values in the data.
* Set to 'true' to enable automatic acceleration of this data model.
* The Splunk platform creates and maintains these column stores on a schedule
  you can specify with 'acceleration.cron_schedule'. You can search them with
  the 'tstats' command.
* Default: false


acceleration.earliest_time = <relative time string>
* Specifies how far back in time the Splunk platform keeps the column stores
  for an accelerated data model.
  * Also specifies when the Splunk platform should create the column stores,
    when you do not have a setting for acceleration.backfill_time.
* Specified by a relative time string. For example, "-7d" means "accelerate
  data within the last 7 days".
* Default: empty string.
  * An empty string for this setting means "keep these stores for all time".

acceleration.backfill_time = <relative time string>
* Specifies how far back in time the Splunk platform creates its
  column stores.
* This is an advanced setting.
* Only set this parameter if you want to backfill less data than the
  retention period set by 'acceleration.earliest_time'. You might want to use
  this parameter to limit your time window for column store creation in a large
  environment where initial creation of a large set of column stores is an
  expensive operation.
```

```
* CAUTION: Do not set 'acceleration.backfill_time' to a narrow time window. If
  one of your indexers is down for a period longer than this backfill time, you
  may miss accelerating a window of your incoming data.
* This setting MUST be set to a time that is more recent than
  'acceleration.earliest_time'. For example, if you set
  'acceleration.earliest_time' to "-1y" to retain your column stores for a one
  year window, you can set 'acceleration.backfill_time' to "-20d" to create
  column stores that cover only the last 20 days. However, you should not set
  'acceleration.backfill_time' to "-2y", because that setting goes farther back
  in time than the 'acceleration.earliest_time' setting of "-1y".
* Default: empty string.
  * When 'acceleration.backfill_time' is unset, the Splunk platform backfills
    fully to 'acceleration.earliest_time'.

acceleration.max_time = <unsigned integer>
* The maximum amount of time, in seconds, that the column store creation search
  can run.
* NOTE: This is an approximate time.
* An 'acceleration.max_time' setting of "0" indicates that there is no time
  limit.
* Default: 3600

acceleration.poll_buckets_until_maxtime = <boolean>
* In a distributed environment consisting of machines with varying amounts of
  free storage capacity and processing speed, summarizations might complete
  sooner on machines with less data and faster resources. After the
  summarization search is finished with all of the buckets, it is complete. The
  overall search runtime is determined by the slowest machine in the
  environment.
* When this setting is set to "true", all of the machines run for "max_time"
  (approximately). The Splunk platform repeatedly polls the buckets for new
  data to summarize.
* Set 'poll_buckets_until_maxtime' to "true" if your data model is sensitive to
  summarization latency delays.
* When 'poll_buckets_until_maxtime' is set to "true", the Splunk platform
  counts the summarization search against the number of concurrent searches you
  can run until "max_time" is reached.
* Default: false

acceleration.cron_schedule = <cron-string>
* This setting provides the cron schedule that the Splunk platform follows when
  it probes or generates the column stores of this data model.
* Default: */5 * * * *

acceleration.manual_rebuilds = <boolean>
* Whether or not the Splunk platform is prohibited from automatically rebuilding
  outdated summaries using the 'summarize' command.
* This is an advanced setting.
* Normally, during the creation phase, the 'summarize' command automatically
  rebuilds summaries that are considered to be out-of-date, such as when the
  configuration backing the data model changes.
* The Splunk platform considers a summary to be outdated when either of these
  conditions are present:
  * The data model search stored in its metadata no longer matches its current
    data model search.
  * The data model search stored in its metadata cannot be parsed.
* When set to "true", the Splunk platform does not rebuild outdated summaries
  using the 'summarize' command.
* NOTE: If the Splunk platform finds a partial summary to be outdated, it always
  rebuilds that summary so that a bucket summary only has results corresponding
  to one data model search.
* Default: false
```

```
acceleration.max_concurrent = <unsigned integer>
* The maximum number of concurrent acceleration instances for this data
  model that the scheduler is allowed to run.
* Default: 3

acceleration.allow_skew = <percentage>|<duration-specifier>
* Allows the search scheduler to randomly distribute scheduled searches more
  evenly over their periods.
* When set to non-zero for searches with the following cron_schedule values,
  the search scheduler randomly "skews" the second, minute, and hour that the
  search actually runs on:
    * * * * *     Every minute.
    */M * * * *   Every M minutes (M > 0).
    0 * * * *     Every hour.
    0 */H * * *   Every H hours (H > 0).
    0 0 * * *     Every day (at midnight).
* When set to non-zero for a search that has any other cron_schedule setting,
  the search scheduler can only randomly "skew" the second that the search runs
  on.
* The amount of skew for a specific search remains constant between edits of
  the search.
* An integer value followed by '%' (percent) specifies the maximum amount of
  time to skew as a percentage of the scheduled search period.
* Otherwise, use <integer><unit> to specify a maximum duration. Relevant units
  are: m, min, minute, mins, minutes, h, hr, hour, hrs, hours, d, day, days.
  The <unit> may be omitted only when the <integer> is 0.
* Examples:
    100% (for an every-5-minute search) = 5 minutes maximum
    50% (for an every-minute search) = 30 seconds maximum
    5m = 5 minutes maximum
    1h = 1 hour maximum
* A value of 0 disallows skew.
* Default: 0

acceleration.schedule_priority = default | higher | highest
* Raises the scheduling priority of a search:
  * "default": No scheduling priority increase.
  * "higher": Scheduling priority is higher than other data model searches.
  * "highest": Scheduling priority is higher than other searches regardless of
    scheduling tier except real-time-scheduled searches with priority = highest
    always have priority over all other searches.
  * Hence, the high-to-low order (where RTSS = real-time-scheduled search, CSS
    = continuous-scheduled search, DMAS = data-model-accelerated search, d =
    default, h = higher, H = highest) is:
      RTSS(H) > DMAS(H) > CSS(H)
      > RTSS(h) > RTSS(d) > CSS(h) > CSS(d)
      > DMAS(h) > DMAS(d)
* The scheduler honors a non-default priority only when the search owner has
  the 'edit_search_schedule_priority' capability.
* CAUTION: Having too many searches with a non-default priority impedes the
  ability of the scheduler to minimize search starvation. Use this setting
  only for mission-critical searches.
* Default: default

acceleration.allow_old_summaries = <boolean>
* Sets the default value of 'allow_old_summaries' for this data model.
* Only applies to accelerated data models.
* When you use commands like 'datamodel', 'from', or 'tstats' to run a search
  on this data model, allow_old_summaries=false causes the Splunk platform to
  verify that the data model search in each bucket's summary metadata matches
  the scheduled search that currently populates the data model summary.
```

```
  Summaries that fail this check are considered "out of date" and are not used
  to deliver results for your events search.
* This setting helps with situations where the definition of an accelerated
  data model has changed, but the Splunk platform has not yet updated its
  summaries to reflect this change. When allow_old_summaries=false for a data
  model, an event search of that data model returns results only from bucket
  summaries that match the current definition of the data model.
* If you set allow_old_summaries=true, your search can deliver results from
  bucket summaries that are out of date with the current data model definition.
* Default: false

acceleration.source_guid = <string>
* Use this setting to enable this data model to use a summary on a remote
  search head (SH) or search head cluster (SHC). You can save space and cut
  back on the work of building and maintaining summaries by accelerating the
  same data model once across multiple SC and SHC instances.
* This setting specifies the GUID (globally unique identifier) of another SH or
  SHC.
  * If you are running a single instance you can find the GUID in
    etc/instance.cfg.
  * You can find the GUID for a SHC in the [shclustering] stanza in server.conf.
* Set this for your data model only if you understand what you are doing!
* After you set this setting:
  * Searches of this data model draw upon the summaries related to the provided
    GUID when possible. You cannot edit this data model in Splunk Web while a
    source GUID is specified for it.
  * The Splunk platform ignores 'acceleration.enabled' and similar acceleration
    settings for your data model.
  * Summaries for this data model cease to be created on the indexers of the
    local deployment even if the model is accelerated.
* All of the data models that use a particular summary should have definitions
  and acceleration time ranges that are very similar to each other, if not
  identical.
  * When you set this setting for this data model, its 'allow_old_summaries'
    setting defaults to 'true'. This happens because there may be a slight
    difference between the definitions of this data model and the data model at
    the remote SC or SHC, whose summary it will be using.
  * If the data model at the remote SC or SHC is changed, this data model could
    end up using mismatched data.
* Default: not set

acceleration.hunk.compression_codec = <string>
* The compression codec to be used for the accelerated orc/parquet files.
* Applicable only to Hunk data models.

acceleration.hunk.dfs_block_size = <unsigned integer>
* The block size, in bytes, for the compression files.
* Applicable only to Hunk data models.

acceleration.hunk.file_format = [orc|parquet]
* Applicable only to Hunk data models.

acceleration.workload_pool = <string>
* Sets the workload pool to be used by this search.
* There are multiple workload pools defined in workload_pools.conf.
  Each workload pool has resource limits associated with it. For example,
  CPU, Memory, etc.
* The specific workload_pool to use is defined in workload_pools.conf.
* The search process for this search runs in the specified workload_pool.
* If workload management is enabled and you have not specified a workload_pool,
  the Splunk platform puts the search into a proper pool as specified by the
  workload rules defined in workload_rules.conf. If you have not defined a rule
```

```
  for this search, the Splunk platform uses the default_pool defined in
  workload_pools.conf.
* Optional.


#******** Dataset-Related Attributes ******
# These attributes affect your interactions with datasets in Splunk Web and
# should not be changed under normal conditions. Do not modify them unless you
# are sure you know what you are doing.

dataset.description = <string>
* User-entered description of the dataset entity.

dataset.type = [datamodel|table]
* The type of dataset:
  * "datamodel": An individual data model dataset.
  * "table": A special root data model dataset with a search where the dataset
    is defined by the dataset.commands attribute.
* Default: datamodel

dataset.commands = [<object>(, <object>)*]
* When the dataset.type = "table" this stringified JSON payload is created by
  the table editor and defines the dataset.

dataset.fields = [<string>(, <string>)*]
* Automatically generated JSON payload when dataset.type = "table" and the
  search for the root data model dataset has been updated.

dataset.display.diversity = [latest|random|diverse|rare]
* The user-selected diversity for previewing events contained by the dataset:
  * "latest": search a subset of the latest events
  * "random": search a random sampling of events
  * "diverse": search a diverse sampling of events
  * "rare": search a rare sampling of events based on clustering
* Default: latest

dataset.display.sample_ratio = <integer>
* The integer value used to calculate the sample ratio for the dataset
  diversity. The formula is 1 / <integer>.
* The sample ratio specifies the likelihood of any event being included in the
  sample.
* For example, if sample_ratio = 500, each event has a 1/500 chance of being
  included in the sample result set.
* Default: 1

dataset.display.limiting = <integer>
* The limit of events to search over when previewing the dataset.
* Default: 100000

dataset.display.currentCommand = <integer>
* The currently selected command the user is on while editing the dataset.

dataset.display.mode = [table|datasummary]
* The type of preview to use when editing the dataset:
  * "table": show individual events/results as rows.
  * "datasummary": show field values as columns.
* Default: table

dataset.display.datasummary.earliestTime = <time-string>
* The earliest time used for the search that powers the datasummary view of
  the dataset.
```

```
dataset.display.datasummary.latestTime = <time-string>
* The latest time used for the search that powers the datasummary view of
  the dataset.

strict_fields = <boolean>
* The default value for the 'strict_fields' argument when you use
  '| datamodel' in a search.
  * When you set 'strict_fields' to 'true', the search returns only the fields
    specified in the constraints for the data model.
  * When you set 'strict_fields' to 'false', the search returns all fields,
    including fields inherited from parent datasets and fields derived through
    search-time processes such as field extraction, eval-based field
    calculation, and lookup matching.
* You can override this setting by specifying the 'strict_fields' argument for
  a '| datamodel' search.
* This setting also applies to the 'from' command. When you use '| from' to
  search a data model that has 'strict_fields=true', the search returns only
  those fields that are defined in the constraints for the data model.
* Default: true

tags_whitelist = <comma-separated list>
* A comma-separated list of tag fields that the data model requires
  for its search result sets.
* This is a search performance setting. Apply it only to data models that use a
  significant number of tag field attributes in their definitions. Data models
  without tag fields cannot use this setting. This setting does not recognize
  tags used in constraint searches.
* Only the tag fields identified in this allow list (and the event types tagged
  by them) are loaded when you perform searches with this data model.
* When you update this setting for an accelerated data model, the Splunk
  software rebuilds the data model unless you have enabled
  accleration.manual_rebuild for it.
* If this setting is not set, the Splunk platform attempts to optimize out
  unnecessary tag fields when you perform searches with this data model.
* Default: empty (not set)
```

datamodels.conf.example

```
#   Version 9.2.2
#
# Configuration for example datamodels
#

# An example of accelerating data for the 'mymodel' datamodel for the
# past five days, generating and checking the column stores every 10 minutes
[mymodel]
acceleration = true
acceleration.earliest_time = -5d
acceleration.poll_buckets_until_maxtime = true
acceleration.cron_schedule = */10 * * * *
acceleration.hunk.compression_codec = snappy
acceleration.hunk.dfs_block_size = 134217728
acceleration.hunk.file_format = orc
```

datatypesbnf.conf

The following are the spec and example files for `datatypesbnf.conf`.

datatypesbnf.conf.spec

```
#   Version 9.2.2
#
# This file effects how the search assistant (typeahead) shows the syntax for
# search commands.
```

[<syntax-type>]

```
* The name of the syntax type you are configuring.
* Follow this field name with one syntax= definition.
* Syntax type can only contain a-z, and -, but cannot begin with -

syntax = <string>
* The syntax for your syntax type.
* Should correspond to a regular expression describing the term.
* Can also be a <field> or other similar value.
```

datatypesbnf.conf.example

```
No example
```

default.meta.conf

The following are the spec and example files for `default.meta.conf`.

default.meta.conf.spec

```
#   Version 9.2.2
#
#
# *.meta files contain ownership information, access controls, and export
# settings for Splunk objects like saved searches, event types, and views.
# Each app has its own default.meta file.

# Interaction of ACLs across app-level, category level, and specific object
# configuration:
* To access/use an object, users must have read access to:
  * the app containing the object
  * the generic category within the app (for example, [views])
  * the object itself
* If any layer does not permit read access, the object will not be accessible.

* To update/modify an object, such as to edit a saved search, users must have:
  * read and write access to the object
  * read access to the app, to locate the object
  * read access to the generic category within the app (for example, [savedsearches])
* If object does not permit write access to the user, the object will not be
  modifiable.
```

```
* If any layer does not permit read access to the user, the object will not be
  accessible in order to modify

* In order to add or remove objects from an app, users must have:
  * write access to the app
* If users do not have write access to the app, an attempt to add or remove an
  object will fail.

* By default, objects are only visible within the app in which they were created.
  To make an object available to all apps, set the object's 'export' setting to
  "system".
  * export = system

* Objects that are exported to other apps, or to system context, have no change
  to their accessibility rules.  Users must still have read access to the
  containing app, category, and object, despite the export.

# Set access controls on the app containing this metadata file.
[]
access = read : [ * ], write : [ admin, power ]
* Allow all users to read this app's contents. Unless overridden by other
  metadata, allow only admin and power users to share objects into this app.

# Set access controls on this app's views.
```

[views]

```
access = read : [ * ], write : [ admin ]
* Allow all users to read this app's views. Allow only admin users to create,
  remove, share, or unshare views in this app.

# Set access controls on a specific view in this app.
```

[views/index_status]

```
access = read : [ admin ], write : [ admin ]
* Allow only admin users to read or modify this view.

# Make this view available in all apps.
export = system
* To make this view available only in this app, set 'export = none' instead.
owner = admin
* Set admin as the owner of this view.
```

default.meta.conf.example

```
#   Version 9.2.2
#
# This file contains example patterns for the metadata files default.meta and
# local.meta
#

# This example would make all of the objects in an app globally accessible to
# all apps
[]
export=system
```

default-mode.conf

The following are the spec and example files for `default-mode.conf`.

default-mode.conf.spec

```
#   Version 9.2.2
#
# This file documents the syntax of default-mode.conf for comprehension and
# troubleshooting purposes.

# default-mode.conf is a file that exists primarily for Splunk Support and
# Services to configure the Splunk platform.

# CAVEATS:

# DO NOT make changes to default-mode.conf without coordinating with Splunk
# Support or Services.  End-user changes to default-mode.conf are not
# supported.
#
# default-mode.conf *will* be removed in a future version of the Splunk platform,
# along with the entire configuration scheme that it affects. Any settings present
# in default-mode.conf files will be completely ignored at this point.
#
# Settings in default-mode.conf affect how pieces of code communicate.
# Configuration changes in default-mode.conf might fail to work,
# behave unexpectedly, or harm your deployment. Any changes must be made
# only under the guidance of Splunk Support or Services staff for
# use in a specific deployment of Splunk Enterprise.

# INFORMATION:

# The main value of this spec file is to assist in reading these files for
# troubleshooting purposes.  default-mode.conf was originally intended to
# provide a way to describe the alternate setups used by the Splunk Light
# Forwarder and Splunk Universal Forwarder.

# The only reasonable action is to re-enable input pipelines that are
# disabled by default in those forwarder configurations.  However, keep the
# prior caveats in mind.  Any future means of enabling inputs will have a
# different form when this mechanism is removed.

# SYNTAX:
```

[pipeline:<string>]

```
disabled = <boolean>
disabled_processors = <string>
```

[pipeline:<string>]

```
* Refers to a particular Splunkd pipeline.
* The set of named pipelines is a splunk-internal design. That does not
  mean that the Splunk design is a secret, but it means it is not external
  for the purposes of configuration.
```

```
* Useful information on the data processing system of splunk can be found
  in the external documentation, for example
  http://docs.splunk.com/Documentation/Splunk/latest/Deploy/Datapipeline


disabled = <boolean>
* Whether or not the Splunk platform loads the specified pipeline.
* If set to true on a specific pipeline, the pipeline will not be loaded in
  the system.

disabled_processors = <processor1>, <processor2>
* Processors which normally would be loaded in this pipeline are not loaded
  if they appear in this list.
* The set of named processors is again a Splunk-internal design component.
```

default-mode.conf.example

```
No example
```

deployment.conf

The following are the spec and example files for `deployment.conf`.

deployment.conf.spec

```
#   Version 9.2.2
#
# *** REMOVED; NO LONGER USED ***
#
#
# This configuration file  has been replaced by:
# 1.) deploymentclient.conf - for configuring Deployment Clients.
# 2.) serverclass.conf - for Deployment Server server class configuration.
#
#
# Compatibility:
# Splunk 4.x Deployment Server is NOT compatible with Splunk 3.x Deployment Clients.
#
```

deployment.conf.example

```
No example
```

deploymentclient.conf

The following are the spec and example files for `deploymentclient.conf`.

deploymentclient.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# customize the way a deployment client behaves.
#
# Each stanza controls different search commands settings.
#
# There is a deploymentclient.conf file in the
# $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name
# deploymentclient.conf in the $SPLUNK_HOME/etc/system/local/ directory.
# Then add the specific settings that you want to customize to the local
# configuration file. For examples, see deploymentclient.conf.example.
# You must restart the Splunk instance to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
#*******************************************************************************
# Configure a Splunk deployment client
#
# Note: At minimum, the [deployment-client] stanza must be in
# deploymentclient.conf to enable a deployment client.
#*******************************************************************************
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top
#     of the file.
#   * Each .conf file should have only one default stanza. If there are
#     multiple default stanzas, their settings combine. When there are
#     multiple definitions of the same setting, the last definition in the
#     file takes precedence.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

[deployment-client]

```
disabled = <boolean>
* Whether or not a deployment client is disabled.
* Default: false

clientName = deploymentClient
* A name that the deployment server can filter on.
* This setting takes precedence over DNS names.
* You can use 'clientName' to filter with, or independently from
  the client IP or DNS name.
* Default: deploymentClient
```

```
workingDir = $SPLUNK_HOME/var/run
* The temporary folder that the deploymentClient uses to download apps and
  configuration content.

repositoryLocation = $SPLUNK_HOME/etc/apps
* The location where content installs when downloaded from a deployment server.
* For the Splunk platform instance on the deployment client to recognize an app
  or configuration content, install the app or content in the default location:
  $SPLUNK_HOME/etc/apps.
    * NOTE: Apps and configuration content for deployment can be in other
      locations on the deployment server. Set both 'repositoryLocation' and
      'serverRepositoryLocationPolicy' explicitly to ensure that the content
      installs on the deployment client in the correct location, which is
      $SPLUNK_HOME/etc/apps.
    * The deployment client uses the following 'serverRepositoryLocationPolicy'
      to determine the value of 'repositoryLocation'.

serverRepositoryLocationPolicy = [acceptSplunkHome|acceptAlways|rejectAlways]
* The value of 'repositoryLocation' for the deployment client to use.
* This setting accepts only the following values:
  * "acceptSplunkHome": Only accept the value of 'repositoryLocation' the
    deployment server supplies if it begins with $SPLUNK_HOME.
  * "acceptAlways": Always accept the 'repositoryLocation' that the deployment server
    supplies.
  * "rejectAlways": Always reject the 'repositoryLocation' that the deployment server
    supplies, and instead use the 'repositoryLocation' that the local
    deploymentclient.conf file specifies.
* Default: acceptSplunkHome

endpoint=$deploymentServerUri$/services/streams/deployment?name=$serverClassName$:$appName$
* Specifies the HTTP endpoint from which to download content.
* The deployment server can specify different endpoints from which to download
  different sets of content, such as individual apps.
* The deployment client uses the following 'serverEndpointPolicy' to determine
  which value to use:
* $deploymentServerUri$ resolves to "targetUri" defined in the following
  'target-broker'stanza.
* $serverClassName$ and $appName$ name the server class and the app,
  respectively.

serverEndpointPolicy = [acceptAlways|rejectAlways]

* acceptAlways: Always accept the endpoint supplied by the server.
* rejectAlways: Reject the endpoint supplied by the server. Always use the
  preceding endpoint definition.
* Default: acceptAlways

phoneHomeIntervalInSecs = <decimal>
* How frequently, in seconds, this deployment client should
  check for new content.
* Fractional seconds are allowed.
* Default: 60.

handshakeRetryIntervalInSecs = <integer>
* The handshake retry frequency, in seconds.
* Could be used to tune the initial connection rate on a new server.
* Default: The value of 'phoneHomeIntervalInSecs' / 5

handshakeReplySubscriptionRetry = <integer>
* If the Splunk platform is unable to complete the handshake, it will retry subscribing to
  the handshake channel after this many handshake attempts.
* Default: 10
```

```
appEventsResyncIntervalInSecs = <number in seconds>
* This sets the interval at which the client reports back its app state
  to the server.
* Fractional seconds are allowed.
* Default: 10 * the value of 'phoneHomeIntervalInSecs'

reloadDSOnAppInstall = <boolean>
* Whether or not the deployment server on this instance reloads after an app
  is installed by this deployment client.
* Setting this flag to true causes the deploymentServer on this Splunk
  platform instance to be reloaded whenever an app is installed by this
  deploymentClient.
* This is an advanced configuration. Only use it when you have a hierarchical
  deployment server installation, and have a Splunk instance that behaves
  as both a deployment client and a deployment server.
* Do not use a hierarchical deployment server unless you have no other
  alternative. Splunk has seen problems in the field that have not yet
  been resolved with this kind of configuration.
* Default: false

sslVersions = <versions_list>
* Comma-separated list of SSL versions to connect to the specified
  Deployment Server
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list but
  does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Default: The 'sslVersions' value in the server.conf file [sslConfig] stanza

sslVerifyServerCert = <boolean>
* If this is set to true, Splunk verifies that the Deployment Server
  (specified in 'targetUri')
  being connected to is a valid one (authenticated).  Both the common
  name and the alternate name of the server are then checked for a
  match if they are specified in 'sslCommonNameToCheck' and 'sslAltNameToCheck'.
  A certificate is considered verified if either is matched.
* Default: The 'sslVerifyServerCert' value in the server.conf file
  [sslConfig] stanza

sslVerifyServerName = <boolean>
* Whether or not a deployment client (DC) performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
* Specifically, the validation check forces the DC to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that the DC performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
```

```
  If that check fails, the DC terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that the DC does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

caCertFile = <path>
* Specifies a full path to a Certificate Authority (ca) certificate(s) PEM
  format file.
* The <path> must refer to a PEM format file containing one or more root CA
  certificates concatenated together.
* Used for validating the SSL certificate from the deployment server
* Default: The 'caCertFile' value in the server.conf file [sslConfig] stanza

sslCommonNameToCheck = <commonName1>, <commonName2>, ...
* If this value is set, and 'sslVerifyServerCert' is set to true,
  splunkd reads the Deployment Servers from 'targetUri', gets the common
  names from these servers' certificates, and checks if they are in this
  list of common names.
* Default: The 'sslCommonNameToCheck' value in the server.conf file
  [sslConfig] stanza.

sslAltNameToCheck =  <alternateName1>, <alternateName2>, ...
* If this value is set, and 'sslVerifyServerCert' is set to true,
  splunkd checks the alternate name(s) of the certificate presented by
  the Deployment Server (specified in 'targetUri') against this list of
  subject alternate names.
* Default: The 'sslAltNameToCheck' value in the server.conf file [sslConfig] stanza

cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for making outbound HTTPS connection.
* No default.

ecdhCurves = <comma separated list of ec curves>
* Defines Elliptic Curve-Diffie Hellman curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only support named curves specified by
  their SHORT names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* For example: ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: empty string

connect_timeout = <positive integer>
* The amount of time, in seconds, that a deployment client can take to connect
  to a deployment server before the server connection times out.
* Default: 60

send_timeout = <positive integer>
* The amount of time, in seconds, that a deployment client can take to send or
  write data to a deployment server before the server connection times out.
* Default: 60

recv_timeout = <positive integer>
* The amount of time, in seconds, that a deployment client can take to receive
  or read data from a deployment server before the server connection times out.
* Default: 60
```

```
# The following stanza specifies deployment server connection information
```

[target-broker:deploymentServer]

```
targetUri= <string>
* The target URI of the deployment server.
* An example of <uri>: <scheme>://<deploymentServer>:<mgmtPort>

connect_timeout = <positive integer>
* See 'connect_timeout' in the "[deployment-client]" stanza for
  information on this setting.

send_timeout = <positive integer>
* See 'send_timeout' in the "[deployment-client]" stanza for
  information on this setting.

recv_timeout = <positive integer>
* See 'recv_timeout' in the "[deployment-client]" stanza for
  information on this setting.
```

deploymentclient.conf.example

```
#   Version 9.2.2
#
# Example 1
# Deployment client receives apps and places them into the same
# repositoryLocation (locally, relative to $SPLUNK_HOME) as it picked them
# up from. This is typically $SPLUNK_HOME/etc/apps.  There
# is nothing in [deployment-client] because the deployment client is not
# overriding the value set on the deployment server side.

[deployment-client]

[target-broker:deploymentServer]
targetUri= https://deploymentserver.splunk.mycompany.com:8089


# Example 2
# Deployment server keeps apps to be deployed in a non-standard location on
# the server side (perhaps for organization purposes).
# Deployment client receives apps and places them in the standard location.
# Note: Apps deployed to any location other than
# $SPLUNK_HOME/etc/apps on the deployment client side will
# not be recognized and run.
# This configuration rejects any location specified by the deployment server
# and replaces it with the standard client-side location.

[deployment-client]
serverRepositoryLocationPolicy = rejectAlways
repositoryLocation = $SPLUNK_HOME/etc/apps

[target-broker:deploymentServer]
targetUri= https://deploymentserver.splunk.mycompany.com:8089


# Example 3
# Deployment client should get apps from an HTTP server that is different
# from the one specified by the deployment server.
```

```
[deployment-client]
serverEndpointPolicy = rejectAlways
endpoint = http://apache.mycompany.server:8080/$serverClassName$/$appName$.tar

[target-broker:deploymentServer]
targetUri= https://deploymentserver.splunk.mycompany.com:8089


# Example 4
# Deployment client should get apps from a location on the file system and
# not from a location specified by the deployment server

[deployment-client]
serverEndpointPolicy = rejectAlways
endpoint = file:/<some_mount_point>/$serverClassName$/$appName$.tar
handshakeRetryIntervalInSecs=20

[target-broker:deploymentServer]
targetUri= https://deploymentserver.splunk.mycompany.com:8089

# Example 5
# Deployment client should phonehome server for app updates quicker
# Deployment client should only send back appEvents once a day

[deployment-client]
phoneHomeIntervalInSecs=30
appEventsResyncIntervalInSecs=86400

[target-broker:deploymentServer]
targetUri= https://deploymentserver.splunk.mycompany.com:8089


# Example 6
# Sets the deployment client connection/transaction timeouts to 1 minute.
# Deployment clients terminate connections if deployment server does not reply.

[deployment-client]
connect_timeout=60
send_timeout=60
recv_timeout=60
```

distsearch.conf

The following are the spec and example files for `distsearch.conf`.

distsearch.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attributes and values you can use to configure
# distributed search.
#
# To set custom configurations, place a distsearch.conf in
# $SPLUNK_HOME/etc/system/local/.  For examples, see distsearch.conf.example.
# You must restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
```

```
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# These attributes are all configured on the search head, with the exception of
# the optional attributes listed under the SEARCH HEAD BUNDLE MOUNTING OPTIONS
# heading, which are configured on the search peers.
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
#     file wins.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

[distributedSearch]
* Set distributed search configuration options under this stanza name.
* Follow this stanza name with any number of the following attribute/value
  pairs.
* If you do not set any attribute, the Splunk platform uses the default value
  (if there is one listed).

disabled = <boolean>
* Whether or not distributed search is disabled.
* To turn distributed search off, set to "true". To turn on, set to "false".
* Default: false (distributed search is enabled by default)

heartbeatMcastAddr = <IP address>
* DEPRECATED.

heartbeatPort = <port>
* DEPRECATED.

ttl = <integer>
* DEPRECATED.

heartbeatFrequency = <integer>
* DEPRECATED.

statusTimeout = <integer>
* The connection timeout when gathering a search peer's basic
  info using the /services/server/info REST endpoint.
* Increasing this value on the Distributed Monitoring Console (DMC) can result
  in fewer peers showing up as "Down" in /services/search/distributed/peers/.
* NOTE: Read/write timeouts are automatically set to twice this value.
* Default: 10

removedTimedOutServers = <boolean>
* This setting is no longer supported, and will be ignored.

checkTimedOutServersFrequency = <integer>
* This setting is no longer supported, and will be ignored.

autoAddServers = <boolean>
* DEPRECATED.
```

```
bestEffortSearch = <boolean>
* This setting determines whether a search peer that's missing the
  knowledge bundle participates in the search.
* If set to "true", the peer participates in the search even if it
  doesn't have the knowledge bundle. The peers that don't have any
  common bundles are simply not searched.
* Default: false

skipOurselves = <boolean>
* DEPRECATED.

servers = <comma-separated list>
* An initial list of servers.
* Each member of this list must be a valid URI in the format of
  scheme://hostname:port

disabled_servers = <comma-separated list>
* A list of disabled search peers. Peers in this list are not monitored
  or searched.
* Each member of this list must be a valid URI in the format of
  scheme://hostname:port

quarantined_servers = <comma-separated list>
* A list of quarantined search peers.
* Each member of this list must be a valid URI in the format of
  scheme://hostname:port
* The admin might quarantine peers that seem unhealthy and are degrading search
  performance of the whole deployment.
* Quarantined peers are monitored but not searched by default.
* A user might use the splunk_server arguments to target a search
  to quarantined peers at the risk of slowing the search.
* When you quarantine a peer, any real-time searches that are running are NOT
  restarted. Currently running real-time searches continue to return results
  from the quarantined peers. Any real-time searches started after the peer
  has been quarantined will not contact the peer.
* Whenever a quarantined peer is excluded from search, appropriate warnings
  are displayed in the search.log and in the Job Inspector.

useDisabledListAsBlacklist = <boolean>
* Whether or not the search head treats the 'disabled_servers' setting as
  a deny list.
* If set to "true", search peers that appear in both the 'servers'
  and 'disabled_servers' lists are disabled and do not participate in search.
* If set to "false", search peers that appear in both lists are enabled
  and participate in search.
* Default: false

useSHPBundleReplication =[true|false|always]
* Whether the search heads in the pool compete with each other to decide which
  one handles the bundle replication (every time bundle replication needs
  to happen), or whether each of them individually replicates the bundles.
* This setting is only relevant in search head pooling environments.
* When set to "always" and you have configured mounted bundles, use the
  search head pool GUID rather than each individual server name to identify
  bundles (and search heads to the remote peers).
* Default: true

trySSLFirst = <boolean>
* This setting is no longer supported, and will be ignored.

peerResolutionThreads = <integer>
* This setting is no longer supported, and will be ignored.
```

```
defaultUriScheme = [http|https]
* The default URI scheme to use if you add a new peer without specifying
  a scheme for the URI to its management port.
* Default: https

serverTimeout = <integer>
* This setting is no longer supported, and will be ignored.
* It has been replaced by the following settings:
  'connectionTimeout', 'sendTimeout', 'receiveTimeout'.

connectionTimeout = <integer>
* The maximum amount of time to wait, in seconds, when the search head
  is attempting to establish a connection to the search peer.
* Default: 10

sendTimeout = <integer>
* The maximum amount of time to wait, in seconds, when the search head
  is attempting to write or send data to a search peer.
* Default: 30

receiveTimeout = <integer>
* The maximum amount of time to wait, in seconds, when the search head
  is attempting to read or receive data from a search peer.
* Default: 600

authTokenConnectionTimeout = <integer>
* The maximum amount of time to wait, in seconds, for the search head
  to connect to a remote search peer when reading its authentication token.
* Fractional seconds are allowed (for example, 10.5 seconds).
* Default: 5

authTokenSendTimeout = <integer>
* The maximum amount of time to wait, in seconds, for the search head
  to send a request to a remote peer when getting its authentication token.
* Fractional seconds are allowed (for example, 10.5 seconds).
* Default: 10

authTokenReceiveTimeout = <integer>
* The maximum amount of time to wait, in seconds, for the search head to
  receive a response from a remote peer when getting its authentication token.
* Fractional seconds are allowed (for example, 10.5 seconds).
* Default: 10

bcs = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* A string that represents the URL for the Bucket Catalog Service.
* Optional.
* There is no default.

bcsPath = <path>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* Default: /bcs/v1/buckets

parallelReduceBackwardCompatibility = [cloud|enterprise]
* This setting determines the conditions under which the Splunk software avoids
  search ID (SID) duplication during parallel reduce search processing by
  appending search head server names to remoteSIDs.
* The conditions under which this behavior is applied differ depending on
```

```
  whether this is a Splunk Cloud Platform or Splunk Enterprise instance.
* A setting of 'cloud' means that this is a Splunk Cloud Platform instance.
  * The Splunk software appends search head server names to remote SIDs as long
    as all of the search heads and indexes in the Splunk Cloud Platform
    deployment share the same version.
  * If the search heads and indexes in this Splunk Cloud Platform deployment do
    not all share the same version, the Splunk software does not change the
    remoteSIDs.
    * In this case the search processor falls back to classic search processing
      methods.
* A setting of 'enterprise' means that this is a Splunk Enterprise instance.
  * The Splunk software appends search head server names to remoteSIDs when all
    of the search heads and index peers in this Splunk Enterprise instance all
    have a version higher than 8.3.0.
  * If this is not the case, the Splunk software does not change the
    remoteSIDs, and in some cases might fall back to classic search processing
    methods.
* Default: enterprise

searchableIndexMapping = enabled|disabled
* Determines whether the search head maintains information on how
  searchable indexes map to search peers.  If enabled, the search
  head periodically requests, from its search peers, a list of
  the searchable indexes that each peer holds.
* Do not change this setting unless directed to do so by
  Splunk Support.
* Default: enabled.
```

DISTRIBUTED SEARCH KEY PAIR GENERATION OPTIONS

```
[tokenExchKeys]

certDir = <directory>
* This directory contains the local Splunk Enterprise instance's distributed
  search key pair.
* This directory also contains the public keys of servers that distribute
  searches to this Splunk Enterprise instance.
* Default: $SPLUNK_HOME/etc/auth/distServerKeys

publicKey = <string>
* The name of the public key file for this Splunk Enterprise instance.
* Default: trusted.pem

privateKey = <string>
* The name of private key file for this Splunk Enterprise instance.
* Default: private.pem

genKeyScript = <string>
* The command used to generate the two files above.
* Default: $SPLUNK_HOME/bin/splunk, createssl, audit-keys

minKeyLength = <integer>
* The minimum key length, in bits, that this Splunk platform instance accepts
  when you configure it as a search peer.
* Typical key lengths are 1024 or 2048, but the 'genKeyScript' can be configured
  to generate 3072- and 4096-bit keys.
* Example: 2048
* Optional.
```

```
* No default.

legacyKeyLengthAuthPolicy = [ warn | reject ]
* This setting applies to existing search heads that were added prior to
  the configuration of a 'minKeyLength' value on this search peer.
* When set to 'warn', this search peer fulfills an authentication token request
  from a search head that supplies a key that is shorter than 'minKeyLength'
  bits, after it first writes a warning message to splunkd.log.
* When set to 'reject', this search peer refuses an authentication token request
  from a search head that supplies a key whose length is too short. It writes
  an error message to splunkd.log about this rejection. This prevents search
  heads from running searches on this search peer when their key lengths
  are not long enough.
* Optional.
* No default.
```

REPLICATION SETTING OPTIONS

```
[replicationSettings]

replicationPolicy = [classic | cascading | rfs | mounted]
* The strategy used by the search head to replicate knowledge bundle across all
  search peers.
* When set to 'classic', the search head replicates bundle to all search peers.
* When set to 'cascading', the search head replicates bundle to a select few
  search peers who in turn replicate to other peers. For tuning parameters for
  cascading replication, refer to the `cascading_replication` stanza in
  server.conf.
* When set to 'rfs', the search head uploads the bundle to the configured remote
  file system like Amazon S3. Note that this policy is not supported for
  on-premise Splunk Enterprise deployments.
* When set to 'mounted', the search head assumes that all the search peers can
  access the correct bundles via shared storage and have configured the
  options listed under the "SEARCH HEAD BUNDLE MOUNTING OPTIONS" heading.
  The 'mounted' option replaces the 'shareBundles' setting, which is no longer
  available. The functionality remains unchanged.
* Default: classic
```

'classic' REPLICATION-SPECIFIC SETTINGS

```
connectionTimeout = <integer>
* The maximum amount of time to wait, in seconds, before a search head's initial
  connection to a peer times out.
* Default: 60

sendRcvTimeout = <integer>
* The maximum amount of time to wait, in seconds, when a search head is sending
  a full replication to a peer.
* Default: 60

replicationThreads = <positive integer>|auto
* The maximum number of threads to use when performing bundle replication
  to peers.
* If set to "auto", the peer auto-tunes the number of threads it uses for
```

```
  bundle replication.
    * If the peer has 3 or fewer CPUs, it allocates 2 threads.
    * If the peer has 4-7 CPUs, it allocates up to '# of CPUs - 2' threads.
    * If the peer has 8-15 CPUs, it allocates up to '# of CPUs - 3' threads.
    * If the peer has 16 or more CPUs, it allocates up to
      '# of CPUs - 4' threads.
* This setting is applicable only when replicationPolicy is set to 'classic'.
* Maximum accepted value for this setting is 16.
* Default: auto

maxMemoryBundleSize = <integer>
* UNSUPPORTED: This setting is no longer supported

maxBundleSize = <integer>
* The maximum bundle size, in megabytes, for which replication can occur.
* If a bundle is larger than this value, bundle replication does not occur and
  the Splunk platform logs an error message.
* The maximum value is 102400 (100 GB).
* If the bundle exceeds 'maxBundleSize', you must increase this value or remove
  files from the bundle to resume normal system operation.
* This value must be larger than the current bundle size. Do not decrease
  it to a value less than the most recent bundle size.
* Bundles reside in the $SPLUNK_HOME/var/run directory on the search head.
  Check the size of the most recent full bundle in that directory.
* If the value for this setting is greater than the value of
  'server.conf:[HttpServer]/max_content_length' on indexers, bundle
  replication failures can occur.
* Default: 2048 (2GB)

warnMaxBundleSizePerc = <integer>
* The search head sends warnings when the knowledge bundle size exceeds this setting's
  percentage of maxBundleSize.
* For example, if maxBundleSize is 2GB and this setting is 50, the search head sends
  warnings when the bundle size exceeds 1GB (2GB * 50%).
* Supported values range from 1 to 100.
* Default: 75

concerningReplicatedFileSize = <integer>
* The maximum allowable file size, in megabytes, within a bundle.
* Any individual file within a bundle that is larger than this value
  triggers a splunkd.log message.
* If excludeReplicatedLookupSize is enabled with a value less than or equal to
  concerningReplicatedFileSize, no warning message will be displayed.
* Where possible, avoid replicating such files by customizing your deny lists.
* Default: 500

excludeReplicatedLookupSize = <integer>
* The maximum allowable lookup file size, in megabytes, during knowledge
  bundle replication.
* Any lookup file larger than this value is excluded from the knowledge bundle
  that the search head replicates to its search peers.
* When this value is set to "0", this feature is disabled. All file sizes
  are included.
* Default: 0

allowStreamUpload = [auto|true|false]
* UNSUPPORTED: This setting is no longer supported

allowSkipEncoding = <boolean>
* UNSUPPORTED: This setting is no longer supported

allowDeltaUpload = <boolean>
```

```
* Whether to enable delta-based bundle replication.
* Delta-based replication keeps the bundle compact, with the search head only
  replicating the changed portion of the bundle to its search peers.
* Default: true

preCompressKnowledgeBundlesClassicMode = <boolean>
* Whether or not this search head cluster member compresses the
  knowledge bundles before replicating them to search peers.
* When set to "true", the search head compresses the bundles
  before replicating them to search peers.
  This helps reduce network bandwidth consumption during replications.
* Default: true

preCompressKnowledgeBundlesCascadeMode = <boolean>
* Whether or not this search head cluster member compresses the
  knowledge bundles before replicating them to search peers.
* When set to "true", the search head compresses the bundles
  before replicating them to search peers.
  This helps reduce network bandwidth consumption during replications.
* This flag applies to cascade mode replication only
* Default: false

sanitizeMetaFiles = <boolean>
* Whether to sanitize or filter *.meta files before replication.
* Use this setting to avoid unnecessary replications triggered by
  writes to *.meta files that have no real effect on search behavior.
* The types of stanzas that "survive" filtering are configured via the
  replicationSettings:refineConf stanza.
* The filtering process removes comments and cosmetic white space.
* Default: true

statusQueueSize = <integer>
* The maximum number of knowledge bundle replication cycle status values that the
  search head maintains in memory. These status values remain accessible by queries.
* Default: 5

allowDeltaIndexing = <boolean>
* Specifies whether to enable delta indexing for knowledge bundle replication.
* Delta indexing causes the indexer to index only those lookup files that have
  changed since the previous bundle, thus reducing the time and resources needed
  to create a new bundle.
* Delta indexing also keeps the bundle compact by using hard links for files that
  have not changed since the previous bundle, instead of copying those files to the
  new bundle.
* Do not change this setting unless instructed to do so by Splunk Support.
* Default: true
```

CASCADING BUNDLE REPLICATION-SPECIFIC SETTINGS

```
cascade_replication_status_interval = <interval>
* The interval at which the cascading replication status thread runs
  to update the cascading replication status for all peers.
* The maximum and recommended value for this setting is 60s.
* The minimum accepted value is 1s.
* Do not change this setting without consulting Splunk Support.
* Default: 60s

cascade_replication_status_unchanged_threshold = <integer>
* The maximum number of intervals (interval length being determined
```

```
  by the "cascade_replication_status_interval" setting) that a peer's
  status can remain unchanged while stuck in an in-progress state.
* Once this limit is reached, the replication is resent to this peer.
* The maximum accepted value for this setting is 20.
* The minimum accepted value for this setting is 1.
* Default: 5

cascade_plan_replication_retry_fast = <boolean>
* Determines whether a cascading bundle replication plan is retried
  if the number of replication failures exceed the threshold
  specified by 'cascade_plan_replication_threshold_failures'.
* Default: true

cascade_plan_replication_threshold_failures = <integer>
* The number of search peers that can fail during a cascading bundle replication
  without triggering a retry of the bundle replication.
* The default value of 0 auto-configures the threshold to
  5% of the peers participating in the bundle replication.
  For example, if there are 80 search peers, auto-configuration
  means that the threshold is 4 peers.
* Do not change this setting without consulting
  Splunk Support.
* Valid only when 'cascade_plan_replication_retry_fast'
  is set to "true".
* Default: 0 (auto configure).
```

RFS (AKA S3/REMOTE FILE SYSTEM) REPLICATION-SPECIFIC SETTINGS

```
enableRFSMonitoring = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* If set to "true", remote file system bundle monitoring is enabled.
* Search peers periodically monitor the configured remote file system
  and download any bundles that they do not have on disk.
* Required on search peers.
* Default: false

rfsMonitoringPeriod = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The amount of time, in seconds, that a search peer waits between polling
  attempts. You must also configure this setting on search heads, whether or
  not the 'enableRFSMonitoring' setting is enabled on them.
* For search heads when the 'rfsSyncReplicationTimeout' setting is set to
  "auto", this setting automatically adapts the 'rfsSyncReplicationTimeout'
  setting to the monitoring frequency of the search peers.
* If you set this value to less than "60", it automatically defaults to 60.
* Default: 60

rfsSyncReplicationTimeout = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The amount of time, in seconds, that a search head waits for synchronous
  replication to complete. Only applies to RFS bundle replication.
* The default value is computed from the 'rfsMonitoringPeriod' setting.
  For example, (rfsMonitoringPeriod + 60) * 5, where 60 is the non-configurable
  polling interval from search heads to search peers, and 5 is an
  arbitrary multiplier.
```

```
* If you do not modify the 'rfsMonitoringPeriod' setting, the default
  value is 600.
* Default: auto

activeServerTimeout = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The amount of time, in seconds, that must elapse before a search peer
  considers the search head to be inactive and no longer attempts to
  download knowledge bundles from that search head from S3/RFS.
* Only applies to RFS bundle replication.
* Default: 360

path = <path>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The remote storage location where bundles reside.
* Required.
* The format for this attribute is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string
    for identifying a location inside the storage system.
* The following external systems are supported:
  * Object stores that support AWS's S3 protocol. These use the scheme "s3".
    Example: "path=s3://mybucket/some/path"
  * POSIX file system, potentially a remote file system mounted over NFS.
    These use the scheme "file".
    Example: "path=file:///mnt/cheap-storage/some/path"

remote.s3.url_version = v1|v2
* Specifies which url version to use, both for parsing the endpoint/path, and
* for communicating with the remote storage. This value only needs to be
* specified when running on non-AWS S3-compatible storage that has been configured
* to use v2 urls.
* In v1 the bucket is the first element of the path.
* Example: mydomain.com/bucketname/rest/of/path
* In v2 the bucket is the outermost subdomain in the endpoint.
* Exmaple: bucketname.mydomain.com/rest/of/path
* Default: v1

remote.s3.endpoint = <URL>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote storage system supporting the S3 API.
* The protocol, http or https, can be used to enable or disable SSL
  connectivity with the endpoint.
* If not specified and the indexer is running on EC2, the endpoint is
  constructed automatically based on the EC2 region of the instance where
  the indexer is running, as follows: https://s3-<region>.amazonaws.com
* Example: https://s3-us-west-2.amazonaws.com

remote.s3.bucket_name = <string>
* Specifies the S3 bucket to use when endpoint isn't set.
* Example
  path = s3://path/example
  remote.s3.bucket_name = mybucket
* Used for constructing the amazonaws.com hostname, as shown above.
* If neither endpoint nor bucket_name is specified, the bucket is assumed
  to be the first path element.
* Optional.

remote.s3.encryption = [sse-s3|none]
```

```
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the schema to use for Server-Side Encryption (SSE) for data at rest.
* sse-s3: See:
  http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html
* none: Server-side encryption is disabled. Data is stored unencrypted on the
  remote storage.
* Optional.
* Default: none

remote.s3.supports_versioning = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies whether the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage. While versioning is not used by
  RFS bundle replication, this much match the configuration of the S3 bucket
  for bundle reaping to work correctly.
* This setting determines how splunkd removes data from remote storage.
  If set to true, splunkd will delete all versions of objects at
  time of data removal. Otherwise, if set to false, splunkd will use a simple DELETE
  (See https://docs.aws.amazon.com/AmazonS3/latest/dev/DeletingObjectVersions.html).
* Optional.
* Default: true
```

SEARCH HEAD BUNDLE MOUNTING OPTIONS

```
# Configure these settings on the search peers only, and only if you also
# configure replicationPolicy=mounted in the [replicationSettings] stanza on the search
# head. Use these settings to access bundles that are not replicated. The search
# peers use a shared
# storage mount point to access the search head bundles ($SPLUNK_HOME/etc).
#*******************************************************************************

[searchhead:<searchhead-splunk-server-name>]
* <searchhead-splunk-server-name> is the name of the related search head
  installation.
* The server name is located in server.conf: serverName = <name>

mounted_bundles = <boolean>
* Determines whether the bundles belonging to the search head specified in the
  stanza name are mounted.
* You must set this value to "true" to use mounted bundles.
* Default: false

bundles_location = <path>
* The path to where the search head's bundles are mounted.
* This path must be the mount point on the search peer, not on the search head.
* The path should point to a directory that is equivalent to $SPLUNK_HOME/etc/.
* The path must contain at least the following subdirectories: system, apps,
  users

[replicationSettings:refineConf]

replicate.<conf_file_name> = <boolean>
* Whether or not the Splunk platform replicates a particular type of
  *.conf file, along with any associated permissions in *.meta files.
* These settings on their own do not cause files to be replicated. You must
  still allow list a file (via the 'replicationAllowlist' setting) in order for
```

```
  it to be eligible for inclusion via these settings.
* In a sense, these settings constitute another level of filtering that applies
  specifically to *.conf files and stanzas with *.meta files.
* Default: false
```

REPLICATION ALLOW LIST OPTIONS

```
[replicationWhitelist]

<name> = <string>
* DEPRECATED; use 'replicationAllowlist' instead.

[replicationAllowlist]

<name> = <string>
* Controls the Splunk platform search-time configuration replication from
  search heads to search peers.
* Only files that match an allow list entry are replicated.
* Conversely, files that do not match an allow list entry are not replicated.
* Only files located under $SPLUNK_HOME/etc will ever be replicated in this way.
  * The regex is matched against the file name, relative to $SPLUNK_HOME/etc.
    Example: For a file "$SPLUNK_HOME/etc/apps/fancy_app/default/inputs.conf",
             this allow list should match "apps/fancy_app/default/inputs.conf"
  * Similarly, the etc/system files are available as system/...
    User-specific files are available as users/username/appname/...
* The 'name' element is generally descriptive, with one exception:
  If <name> begins with "refine.", files allow listed by the given pattern will
  also go through another level of filtering configured in the
  [replicationSettings:refineConf] stanza.
* The allow list pattern is the Splunk style pattern matching, which is
  primarily regex-based with special local behavior for '...' and '*'.
  * '...' matches anything, while '*' matches anything besides
    directory separators. See props.conf.spec for more detail on these.
  * Note: '.' will match a literal dot, not any character.
* These lists are applied globally across all configuration data, not to any
  particular application, regardless of where they are defined. Be careful to
  pull in only your intended files.
```

REPLICATION DENY LIST OPTIONS

```
[replicationBlacklist]

<name> = <string>
* DEPRECATED; use 'replicationDenylist' instead.

[replicationDenylist]

<name> = <string>
* All comments from the replication allow list notes above also apply here.
* Replication deny list takes precedence over the allow list, meaning that a
  file that matches both the allow list and the deny list is NOT replicated.
* Use this setting to prevent unwanted bundle replication in two common
  scenarios:
```

```
    * Very large files which part of an application might not want to be
      replicated, especially if they are not needed on search nodes.
    * Frequently updated files (for example, some lookups) will trigger
      retransmission of all search head data.
* These lists are applied globally across all configuration data. Especially
  for deny listing, be sure to constrain your deny list to match only data
  that your application does not need.
```

BUNDLE ENFORCER ALLOW LIST OPTIONS

```
[bundleEnforcerWhitelist]

<name> = <string>
* DEPRECATED; use 'bundleEnforcerAllowlist' instead.

[bundleEnforcerAllowlist]

<name> = <string>
* Peers use this setting to make sure knowledge bundles sent by search heads and
  masters do not contain alien files.
* If this stanza is empty, the receiver accepts the bundle unless it contains
  files matching the rules specified in the [bundleEnforcerDenylist] stanza.
  Hence, if both [bundleEnforcerAllowlist] and [bundleEnforcerDenylist] are
  empty (which is the default), then the receiver accepts all bundles.
* If this stanza is not empty, the receiver accepts the bundle only if it
  contains only files that match the rules specified here but not those in the
  [bundleEnforcerDenylist] stanza.
* All rules are regular expressions.
* No default.
```

BUNDLE ENFORCER DENY LIST OPTIONS

```
[bundleEnforcerBlacklist]

<name> = <string>
* DEPRECATED; use 'bundleEnforcerDenylist' instead.

[bundleEnforcerDenylist]

<name> = <string>
* Peers use this setting to make sure knowledge bundle sent by search heads and
  masters do not contain alien files.
* This list overrides the [bundleEnforceAllowlist] stanza above. This means that
  the receiver removes the bundle if it contains any file that matches the
  rules specified here even if that file is allowed by [bundleEnforcerAllowlist].
* If this stanza is empty, then only [bundleEnforcerAllowlist] matters.
* No default.
```

DISTRIBUTED SEARCH GROUP DEFINITIONS

```
# These settings are the definitions of the distributed search groups. A search
# group is a set of search peers as identified by thier host:management-port. A
# search can be directed to a search group using the splunk_server_group argument.
# The search is dispatched to only the members of the group.
#*******************************************************************************

[distributedSearch:<splunk-server-group-name>]
* <splunk-server-group-name> is the name of the Splunk server group that is
  defined in this stanza

servers = <comma-separated list>
* A list of search peers that are members of this group.
* The list must use peer identifiers (i.e. hostname:port).

default = <boolean>
* Specifies whether this distributed search group is the default distributed
  search group.
* A setting of 'true' means that any search that does not explicitly specify a
  distributed search group runs against this default distributed search group
  of peers.
* You can set 'Default=true' for only one distributed search group at any
  given time.
* If you do not specify a distributed search group in your search, the full set
  of search peers in the '[distributedSearch]' stanza is searched under the
  following circumstances:
  * You do not set any of your distributed search groups to 'default=true'.
  * You set 'default=true' for a distributed search group, but you do not
    define a 'servers' list for that distributed search group.
* Default: false
```

distsearch.conf.example

```
#   Version 9.2.2
#
# These are example configurations for distsearch.conf. Use this file to
# configure distributed search.  For all available attribute/value pairs, see
# distsearch.conf.spec.
#
# There is NO DEFAULT distsearch.conf.
#
# To use one or more of these configurations, copy the configuration block into
# distsearch.conf in $SPLUNK_HOME/etc/system/local/.  You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[distributedSearch]
servers = https://192.168.1.1:8059,https://192.168.1.2:8059

# This entry distributes searches to 192.168.1.1:8059,192.168.1.2:8059.
# These machines will be contacted on port 8059 using https
# Attributes not set here will use the defaults listed in distsearch.conf.spec.
```

```
# this stanza controls the timing settings for connecting to a remote peer and
# the send timeout
[replicationSettings]
connectionTimeout = 10
sendRcvTimeout = 60

# this stanza controls what files are replicated to the other peer each is a
# regex
[replicationAllowlist]
allConf = *.conf

# Mounted bundles example.
# This example shows two distsearch.conf configurations, one for the search
# head and another for each of the search head's search peers. It shows only
# the attributes necessary to implement mounted bundles.

# On a search head whose Splunk server name is "searcher01":
[replicationSettings]
...
replicationPolicy = mounted

# On each search peer:
[searchhead:searcher01]
mounted_bundles = true
bundles_location = /opt/shared_bundles/searcher01
```

eventdiscoverer.conf

The following are the spec and example files for `eventdiscoverer.conf`.

eventdiscoverer.conf.spec

```
#   Version 9.2.2

# This file contains possible settings and values you can use to configure
# event discovery through the search command "typelearner."
#
# There is an eventdiscoverer.conf in $SPLUNK_HOME/etc/system/default/.  To set
# custom configurations, place an eventdiscoverer.conf in
# $SPLUNK_HOME/etc/system/local/.  For examples, see
# eventdiscoverer.conf.example. You must restart Splunk to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
```

```
#     file wins.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

ignored_keywords = <comma-separated list of terms>
* If you find that event types have terms you do not want considered (for
  example, "mylaptopname"), add that term to this list.
* Terms in this list are never considered for defining an event type.
* For more details, see $SPLUNK_HOME/etc/system/default/eventdiscoverer.conf).
* Default = "sun, mon, tue,..."

ignored_fields = <comma-separated list of fields>
* Similar to ignored_keywords, except these are fields as defined in Splunk
  instead of terms.
* Defaults include time-related fields that would not be useful for defining an
  event type.

important_keywords = <comma-separated list of terms>
* When there are multiple possible phrases for generating an eventtype search,
  those phrases with important_keyword terms are favored.  For example,
  "fatal error" would be preferred over "last message repeated", as "fatal" is
  an important keyword.
* Default = "abort, abstract, accept,..."
* For the full default setting, see $SPLUNK_HOME/etc/system/default/eventdiscoverer.conf.
```

eventdiscoverer.conf.example

```
#   Version 9.2.2
#
# This is an example eventdiscoverer.conf.  These settings are used to control
# the discovery of common eventtypes used by the typelearner search command.
#
# To use one or more of these configurations, copy the configuration block into
# eventdiscoverer.conf in $SPLUNK_HOME/etc/system/local/.  You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# Terms in this list are never considered for defining an eventtype.
ignored_keywords = foo, bar, application, kate, charlie

# Fields in this list are never considered for defining an eventtype.
ignored_fields = pid, others, directory
```

event_renderers.conf

The following are the spec and example files for `event_renderers.conf`.

event_renderers.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attribute/value pairs for configuring event rendering properties.
```

```
#
# Beginning with version 6.0, Splunk Enterprise does not support the
# customization of event displays using event renderers.
#
# There is an event_renderers.conf in $SPLUNK_HOME/etc/system/default/.  To set custom configurations,
# place an event_renderers.conf in $SPLUNK_HOME/etc/system/local/, or your own custom app directory.
#
# To learn more about configuration files (including precedence) please see the documentation
# located at http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#     * You can also define global settings outside of any stanza, at the top of the file.
#     * Each conf file should have at most one default stanza. If there are multiple default
#       stanzas, attributes are combined. In the case of multiple definitions of the same
#       attribute, the last definition in the file wins.
#     * If an attribute is defined at both the global level and in a specific stanza, the
#       value in the specific stanza takes precedence.
```

[<name>]

```
* Stanza name. This name must be unique.

eventtype = <event type>
* Specify event type name from eventtypes.conf.

priority = <positive integer>
* Highest number wins!!

template = <valid Mako template>
* Any template from the $APP/appserver/event_renderers directory.

css_class = <css class name suffix to apply to the parent event element class attribute>
* This can be any valid css class value.
* The value is appended to a standard suffix string of "splEvent-". A css_class value of foo would
result in the parent element of the event having an html attribute class with a value of splEvent-foo
(for example, class="splEvent-foo"). You can externalize your css style rules for this in
$APP/appserver/static/application.css. For example, to make the text red you would add to
application.css:.splEvent-foo { color:red; }
```

event_renderers.conf.example

```
#   Version 9.2.2
# DO NOT EDIT THIS FILE!
# Please make all changes to files in $SPLUNK_HOME/etc/system/local.
# To make changes, copy the section/stanza you want to change from $SPLUNK_HOME/etc/system/default
# into ../local and edit there.
#
# This file contains mappings between Splunk eventtypes and event renderers.
#
# Beginning with version 6.0, Splunk Enterprise does not support the
# customization of event displays using event renderers.
#
```

```
[event_renderer_1]
eventtype = hawaiian_type
priority = 1
css_class = EventRenderer1

[event_renderer_2]
eventtype = french_food_type
priority = 1
template = event_renderer2.html
css_class = EventRenderer2

[event_renderer_3]
eventtype = japan_type
priority = 1
css_class = EventRenderer3
```

eventtypes.conf

The following are the spec and example files for `eventtypes.conf`.

eventtypes.conf.spec

```
#   Version 9.2.2
#
# This file contains descriptions of the settings that you can use to
# configure event types and their properties.
#
# Each stanza controls different settings.
#
# There is an eventtypes.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name eventtypes.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see eventtypes.conf.example.
#
# Any event types that you create through Splunk Web are automatically added to
# the user's $SPLUNK_HOME/etc/users/$user/$app/local/eventtypes.conf file.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top
#    of the file.
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, attributes are combined. In the case of
#    multiple definitions of the same attribute, the last definition in the
#    file wins.
```

```
#  * If an attribute is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.
```

[<$EVENTTYPE>]

```
* Header for the event type
* $EVENTTYPE is the name of your event type.
* You can have any number of event types, each represented by a stanza and
  any number of the following attribute/value pairs.
* NOTE: If the name of the event type includes field names surrounded by the
  percent character (for example "%$FIELD%") then the value of $FIELD is
  substituted into the event type name for that event.  For example, an
  event type with the header [cisco-%code%] that has "code=432" becomes
  labeled "cisco-432".

disabled = [1|0]
* Toggle event type on or off.
* Set to 1 to disable.

search = <string>
* Search terms for this event type.
* For example: error OR warn.
* NOTE: You cannot base an event type on:
* A search that includes a pipe operator (a "|" character).
* A subsearch (a search pipeline enclosed in square brackets).
* A search referencing a report. This is a best practice. Any report that is referenced by an
  event type can later be updated in a way that makes it invalid as an event type. For example,
  a report that is updated to include transforming commands cannot be used as the definition for
  an event type. You have more control over your event type if you define it with the same search
  string as the report.

priority = <integer, 1 through 10>
* Value used to determine the order in which the matching eventtypes of an
  event are displayed.
* 1 is the highest priority and 10 is the lowest priority.

description = <string>
* Optional human-readable description of this saved search.

tags = <string>
* DEPRECATED - see tags.conf.spec

color = <string>
* color for this event type.
* Supported colors: none, et_blue, et_green, et_magenta, et_orange,
  et_purple, et_red, et_sky, et_teal, et_yellow
```

eventtypes.conf.example

```
#   Version 9.2.2
#
# This file contains an example eventtypes.conf.  Use this file to configure custom eventtypes.
#
# To use one or more of these configurations, copy the configuration block into eventtypes.conf
# in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the documentation
```

```
# located at http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#

# The following example makes an eventtype called "error" based on the search "error OR fatal."

[error]
search = error OR fatal



# The following example makes an eventtype template because it includes a field name
# surrounded by the percent character (in this case "%code%").
# The value of "%code%" is substituted into the event type name for that event.
# For example, if the following example event type is instantiated on an event that has a
# "code=432," it becomes "cisco-432".

[cisco-%code%]
search = cisco
```

federated.conf

The following are the spec and example files for `federated.conf`.

federated.conf.spec

```
#   Version 9.2.2
#
# This file contains possible setting and value pairs for federated provider entries
# for use when the federated search functionality is enabled.
#
# A federated search allows authorized users to run searches across multiple federated
# providers. Only Splunk deployments are supported as federated providers. Information
# on the Splunk deployment (i.e. the federated provider) is added in the federated
# provider stanza of the federated.conf file. A federated search deployment can have
# multiple federated search datasets. The settings for federated search dataset stanzas
# are located in savedsearches.conf.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#

#
# Federated Provider Stanza
#
```

[provider]

```
* Each federated provider definition must have a separate stanza.
* <provider> must follow the following syntax:
  provider://<unique-federated-provider-name>
* <unique-federated-provider-name> can contain only alphanumeric characters and
  underscores.

type = [splunk]
* Specifies the type of the federated provider.
* A setting of 'splunk' means that the federated provider is a Splunk
  deployment.
```

```
* Default: splunk

hostPort = <Host_Name_or_IP_Address>:<service_port>
* Specifies the protocols required to connect to a federated provider.
* You can provide a host name or an IP address.
* The <service_port> can be any legitimate port number.
* No default.

serviceAccount = <user_name>
* Specifies the user name for a service account that has been set up on the
  federated provider for the purpose of enabling secure federated search.
* This service account allows the federated search head on your local Splunk
  platform deployment to query datasets on the federated provider in a secure
  manner.
* No default.

password = <password>
* Specifies the service account password for the user specified in the
  'serviceAccount' setting.
* No default.

appContext = <application_short_name>
* Specifies the Splunk application context for the federated searches that are
  run with this federated provider definition.
* NOTE: Applicable only to federated providers that have 'type = splunk' and
 'mode = standard'.
  * Federated providers with 'type = splunk' and 'mode = transparent' ignore
    the 'appContext' property. Such providers instead apply the application
    context of the federated search that is run from the local search head to
    the remote portion of the federated search that is run on the remote
    search head.
* Provision of an application context ensures that federated searches which use
  the federated provider are limited to the knowledge objects that are
  associated with the named application. Application context can also affect
  search job quota and resource allocation parameters.
* '<application_short_name>' must be the short name of a Splunk application
  currently installed on the federated provider. For example, the short name of
  Splunk IT Service Intelligence is 'itsi'.
  * Find the short names of apps installed on a Splunk deployment by going to
    'Apps > Manage Apps' and reviewing the values in the 'Folder name' column.
* You can create multiple federated provider definitions with 'type = splunk'
  and 'mode = standard' for the same remote search head that differ only by
  name and application context.
* Default: search

useFSHKnowledgeObjects = <boolean>
* Determines whether federated searches with this provider use knowledge
  objects from the federated provider (the remote search head) or from the
  federated search head (the local search head).
* When set to 'true' federated searches with this provider use knowledge
  objects from the federated search head.
* NOTE: This setting can be set to "true" only when the federated provider is in
  transparent mode. If this setting is set to "true" on a standard mode
  provider, the Splunk software considers the provider to be misconfigured and
  ignores this setting when you run searches on it. So Splunk software always
  uses knowledge objects from the federated provider in standard mode.
* Default: false

mode = [ standard | transparent ]
* Specifies whether a federated provider is in standard or transparent mode.
* A setting of 'transparent' means that searches with the federated provider
  can use only knowledge objects from the federated search head. In other
```

```
  words, the value for 'useFSHKnowledgeObjects' is always interpreted by the
  transparent mode federated provider as 'true'.
* A setting of 'standard' means that the federated provider respects the
  setting of 'useFSHKnowledgeObjects'. In other words, searches with the
  federated provider can use knowledge objects from the remote search head or
  the federated search head.
* Default: standard


#
# General Federated Search Stanza
#
```

[general]

```
* This stanza is for settings that are applicable to the overall logic for
  search federation. They are typically applicable to all federated providers
  and all search head cluster members.

needs_consent = <boolean>
* A setting of 'true' causes a checkbox to appear in the federated provider
  definition UI. This checkbox requires that users legally acknowledge that
  federated providers can be set up in a manner detrimental to regulatory
  compliance.
* Default: true

heartbeatEnabled = <boolean>
* Specifies whether the federated search heartbeat mechanism is running.
* A setting of 'true' means the heartbeat mechanism is running on an interval
  specified by 'heartbeatInterval'.
* The heartbeat mechanism monitors the remote federated providers for this
  Splunk platform instance. When you run federated searches and the heartbeat
  mechanism has detected problems with the federated providers, it can tell you
  what is wrong and take actions.
  * If a federated provider is found to be unreachable a consecutive number of
    times set by 'connectivityFailuresThreshold', the heartbeat mechanism sets
    the federated provider to an invalid state, meaning it ignores the
    unreachable provider in federated searches.
      * When the heartbeat mechanism reconnects to the provider, it resets the
        provider to a valid state.
  * If two transparent mode federated providers are found to point to the same
    server ID, the heartbeat mechanism randomly chooses one provider to run the
    search over.
    * On Splunk Enterprise deployments, this functionality is extended so that
      it also detects when two transparent mode federated providers share the
      same cluster ID. For this extension to work, the service accounts for the
      transparent mode federated providers must have the
      list_search_head_clustering capability.
* A setting of 'false' means the heartbeat mechanism does not take actions when
  it detects problems with providers.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: true

heartbeatInterval = <integer>
* The interval, in seconds, of the federated search heartbeat mechanism.
  It's value should be greater than 5 seconds.
* When 'heartbeatEnabled = true' the federated search heartbeat mechanism
  performs its federated provider monitoring activities on this interval.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
```

```
* Default: 60

connectivityFailuresThreshold = <integer>
* When the federated search heartbeat mechanism detects this number of
  consecutive connectivity failures for a specific remote provider, the
  heartbeat mechanism sets the remote provider to an invalid state.
* When the heartbeat mechanism successfully reconnects to an invalid state
  federated provider, it resets the federated provider to a valid state.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: 3

controlCommandsMaxThreads = <int>
* The maximum number of threads that can run a federated search action, such as
  a search pause or search cancellation, from a local federated search head on
  the federated providers.
* Change this setting only when directed to do so by Splunk Support.
* Default: 5

controlCommandsMaxTimeThreshold = <int>
* The maximum number of seconds that a federated search action, such as
  a search pause or search cancellation, from a local federated search head waits
  for the federated providers to finish the same command.
* Change this setting only when directed to do so by Splunk Support.
* Default: 5

controlCommandsFeatureEnabled = <boolean>
* Specifies whether a federated search head can send a federated search action,
  such as a search pause or search cancellation, to federated providers.
* Change this setting only when directed to do so by Splunk Support.
* Default: true

proxyBundlesTTL = <int>
* Specifies the time to live in seconds of a proxy bundle on the remote search
  head after the last time it was used for a search.
* Change this setting only when directed to do so by Splunk Support.
* Default: 172800

remoteEventsDownloadRetryCountMax = <integer>
* When you run a verbose-mode federated search, the federated search head
  downloads events from the federated provider.
* If this event download fails, the federated search head retries the download.
* This setting sets the maximum number of event download retries that the
  federated search head can make before it reports a failure.
* See 'remoteEventsDownloadRetryTimeoutMs' for the interval between retries.
* Change this setting only when directed to do so by Splunk Support.
* Default: 20

remoteEventsDownloadRetryTimeoutMs = <int>
* Specifies the interval, in milliseconds, between retries of a failed event
  download from a federated provider.
* See 'remoteEventsDownloadRetryCountMax' for the total number of event
  download retries a federated search head can make before it must report a
  failure.
* Change this setting only when directed to do so by Splunk Support.
* Default: 1000

verbose_mode = <boolean>
* Specifies whether federated searches can be run in verbose mode.
* A setting of 'false' restricts the ability of federated searches to run in
  verbose mode, while allowing federated searches to run in fast and smart
  mode.
```

```
* In Transparent Mode, a setting of 'false' means that Splunk software runs
  only the local portion of a verbose mode federated search.
* In Standard Mode, a setting of 'false' terminates verbose mode federated
  searches without displaying their results.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true

max_preview_generation_duration = <unsigned integer>
* The maximum amount of time, in seconds, that the search head can spend to
  generate search result previews.
* NOTE: This setting applies only to federated searches.
* This limit does not stop federated searches from completing and returning
  final result sets.
* When this limit is reached by a federated search, preview generation is
  halted, but the search continues gathering results until it completes and
  displays the final result set.
* Change the value of this setting to a number above zero if you find that your
  federated searches are being terminated because their preview generation
  duration exceeds a timeout set by another component in your network, such as
  an elastic load balancer (ELB).
  * For example, if you have an ELB that times out at 60 seconds, you might set
    the 'max_preview_generation_duration' to '55'.
* A setting of '0' means that the preview generation duration of federated
  searches is unlimited.
* Default: 0


############################################################################
# Configs for blocking unsupported commands in Federated Search
############################################################################

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_standard_mode_unsupported_command:metadata]

```
* This stanza controls whether the metadata command is blocked for
  Federated Search for Splunk on standard mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'metadata' command for standard mode
  federated search.
  * A value of "true" means that the 'metadata' command is not blocked for
    standard mode federated search.
  * A value of "false" means that the 'metadata' command is blocked for
    standard mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_standard_mode_unsupported_command:metasearch]

```
* This stanza controls whether the metasearch command is blocked for
  Federated Search for Splunk on standard mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'metasearch' command for standard mode
  federated search.
```

```
  * A value of "true" means that the 'metasearch' command is not blocked for
    standard mode federated search.
  * A value of "false" means that the 'metasearch' command is blocked for
    standard mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:makeresults]

```
* This stanza controls whether Splunk software blocks the 'makeresults' command
  on transparent mode federated providers for Federated Search for Splunk.

active = <boolean>
* Controls whether Splunk software blocks the 'makeresults' command for
  transparent mode federated search.
  * A value of "true" means that Splunk software does not block the
    'makeresults' command for transparent mode federated search.
  * A value of "false" means that Splunk software blocks the 'makeresults'
    command for transparent mode federated search. The 'makeresults' command
    still runs on your local search head.
* Even when 'active=false', you can run a 'makeresults' search over a
  transparent mode federated provider when the following things are true:
  * The 'allow_target' setting is set to 'true' and you use the 'splunk_server'
    or 'splunk_server_group' arguments in conjunction with the 'makeresults'
    command.
  * The 'splunk_server' or 'splunk_server_group' arguments point to a server or
    server group that exists on the transparent mode federated provider.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

allow_target = <boolean>
* Determines whether you can run the 'makeresults' command over transparent
  mode federated providers with the 'splunk_server' or 'splunk_server_group'
  arguments even when 'active = false'.
  * A value of "true" means that you can run the specified command over
    transparent mode federated providers when you use the 'splunk_server' or
    'splunk_server_group' argument in conjunction with the command.
    * If you do not specify a server or server group that exists on the the
      transparent mode federated provider, Splunk software blocks 'makeresults'
      for transparent mode federated search, and runs only on your local search
      head.
  * A value of "false" means that you cannot run 'makeresults' over transparent
    mode federated providers even when you use the 'splunk_server' or
   'splunk_server_group' arguments to specify servers or server groups that
    exist on the transparent mode provider.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: true

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:delete]

```
* This stanza controls whether the delete command is blocked for
  Federated Search for Splunk on transparent mode federated providers.
```

```
active = <boolean>
* Whether Splunk software blocks the 'delete' command for transparent mode
  federated search.
  * A value of "true" means that the 'delete' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'delete' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:dump]

```
* This stanza controls whether the dump command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'dump' command for transparent mode
  federated search.
  * A value of "true" means that the 'dump' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'dump' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:map]

```
* This stanza controls whether the map command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'map' command for transparent mode
  federated search.
  * A value of "true" means that the 'map' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'map' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:run]

```
* This stanza controls whether the run command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'run' command for transparent mode
```

```
  federated search.
  * A value of "true" means that the 'run' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'run' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:runshellscript]

```
* This stanza controls whether the runshellscript command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'runshellscript' command for transparent mode
  federated search.
  * A value of "true" means that the 'runshellscript' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'runshellscript' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:script]

```
* This stanza controls whether the script command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'script' command for transparent mode
  federated search.
  * A value of "true" means that the 'script' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'script' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:sendalert]

```
* This stanza controls whether the sendalert command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'sendalert' command for transparent mode
  federated search.
  * A value of "true" means that the 'sendalert' command is not blocked for
    transparent mode federated search.
```

```
  * A value of "false" means that the 'sendalert' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:sendemail]

```
* This stanza controls whether the sendemail command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'sendemail' command for transparent mode
  federated search.
  * A value of "true" means that the 'sendemail' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'sendemail' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change this setting only when instructed to do so by Splunk Support.
```

[s2s_transparent_mode_unsupported_command:rest]

```
* This stanza controls whether the rest command is blocked for
  Federated Search for Splunk on transparent mode federated providers.

active = <boolean>
* Whether Splunk software blocks the 'rest' command for transparent mode
  federated search.
  * A value of "true" means that the 'rest' command is not blocked for
    transparent mode federated search.
  * A value of "false" means that the 'rest' command is blocked for
    transparent mode federated search.
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
* Default: false

# Change the settings in this stanza only when Splunk Support instructs you to
# do so.
```

[s2s_transparent_mode_unsupported_command:summarize]

```
* This stanza controls whether Splunk software blocks the 'summarize' command
  on transparent mode federated providers for Federated Search for Splunk.
* Note: The 'summarize' command is an internal command. Use it only under the
  direction of Splunk Support.

active = <boolean>
* Controls whether Splunk software blocks the 'summarize' command for
  transparent mode federated search.
  * A value of "true" means that Splunk software does not block the
    'summarize' command for transparent mode federated search.
```

```
  * A value of "false" means that Splunk software blocks the 'summarize'
    command for transparent mode federated search. The 'summarize' command
    still runs on your local search head.
* Transparent mode federated providers with lower versions encounter
  complications when they run the 'summarize' command. For those providers, the
  command must always be blocked. The 'rsh_min_version_cloud' and
  'rsh_version_onprem' settings ensure that 'summarize' is blocked for
  transparent mode federated providers that have versions lower than the
  versions those settings specify, even when 'active=true'.
* Default: true

rsh_min_version_cloud = <string>
* Specifies the minimal Splunk Cloud Platform version with full support for
  'summarize'.
* Affects only transparent mode federated providers.
* This setting blocks 'summarize' for any Splunk Cloud Platform transparent
  mode federated provider with a version lower than this setting.
* Default: 9.0.2303.100

rsh_min_version_onprem = <string>
* Specifies the minimal Splunk Enterprise version with full support for
  'summarize'.
* Affects only transparent mode federated providers.
* This setting blocks 'summarize' for any Splunk Enterprise transparent mode
  federated provider with a version lower than this setting.
* Default: 9.1.0

# Change the settings in this stanza only when Splunk Support instructs you to
# do so.
```

[s2s_transparent_mode_unsupported_command:tstats]

```
* This stanza controls whether Splunk software blocks the 'tstats' command
  on transparent mode federated providers for Federated Search for Splunk.

active = <boolean>
* Controls whether Splunk software blocks the 'tstats' command for
  transparent mode federated search.
  * A value of "true" means that Splunk software does not block the
    'tstats' command for transparent mode federated search.
  * A value of "false" means that Splunk software blocks the 'tstats'
    command for transparent mode federated search. The 'tstats' command
    still runs on your local search head.
* Under certain conditions, transparent mode federated providers with lower
  versions encounter complications when they run the 'tstats' command.
  * The 'rsh_min_version_cloud' and 'rsh_version_onprem' settings block
    'tstats' searches that inlude 'FROM' clauses for transparent mode
    federated providers that have versions lower than the versions the
    'rsh_min_version_cloud' and 'rsh_version_onprem' settings specify,
    even when 'active=true'.
  * However, if a 'tstats' search on a lower-version transparent mode federated
    provider does not include a 'FROM' clause, Splunk software ignores the
    'rsh_min_version_cloud' and 'rsh_version_onprem' settings and allows the
    'tstats' search to proceed.
* Default: true

rsh_min_version_cloud = <string>
* Specifies the minimal Splunk Cloud Platform version with full support for
  'tstats'.
* Affects only transparent mode federated providers.
* This setting blocks 'tstats' for any Splunk Cloud Platform transparent mode
```

```
  federated provider with a version lower than this setting, when the 'tstats'
  search includes a 'FROM' clause.
* Default: 9.0.2303.100

rsh_min_version_onprem = <string>
* Specifies the minimal Splunk Enterprise version with full support for
  'tstats'.
* Affects only transparent mode federated providers.
* This setting blocks 'tstats' for any Splunk Enterprise transparent mode
  federated provider with a version lower than this setting, when the 'tstats'
  search includes a 'FROM' clause.
* Default: 9.1.0
```

federated.conf.example

```
#   Version 9.2.2
#
# Here are some examples of stanzas in federated.conf
#
#
[provider://provider_1]
hostPort = remote_searchhead1:8090
password = secret1
serviceAccount = user1
type = splunk
appContext = search
useFSHKnowledgeObjects = 0
mode = standard

[provider://provider_2]
hostPort = remote_searchhead2:8090
password = secret2
serviceAccount = user2
type = splunk
appContext = search
useFSHKnowledgeObjects = 1
mode = transparent


[general]
# Limit preview for Federated Searches
max_preview_generation_duration = 0
```

fields.conf

The following are the spec and example files for `fields.conf`.

fields.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains possible attribute and value pairs for:
#  * Telling Splunk how to handle multi-value fields.
#  * Distinguishing indexed and extracted fields.
#  * Improving search performance by telling the search processor how to
#    handle field values.
#
# Each stanza controls different search commands settings.
#
# There is a fields.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name fields.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see fields.conf.example.
# You must restart the Splunk instance to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
#
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
#     file wins.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

[<field name>|sourcetype::<sourcetype>::<wildcard expression>]

```
* The name of the field that you are configuring. This can be a simple field name,
  or it can be a wildcard expression that is scoped to a source type.
* Field names can contain only "a-z", "A-Z", "0-9", "." , ":", and "_". They
  cannot begin with a number or "_".
  Field names cannot begin with a number "0-9" or an underscore "_".
* Wildcard expressions have the same limitations as field names, but they can
  also contain and/or start with a *.
* Do not create indexed fields with names that collide with names of fields
  that are extracted at search time.
* A source-type-scoped wildcard expression causes all indexed fields that match
  the wildcard expression to be scoped with the specified source type.
  * Apply source-type-scoped wildcard expressions to all fields associated with
    structured data source types, such as JSON-formatted data. Do not apply it
    to mixed datatypes that contain both structured and unstructured data.
  * When you apply this method to structured data fields, searches against
```

```
    those fields should complete faster.
  * Example: '[sourcetype::splunk_resource_usage::data*]' defines all fields
    starting with "data" as indexed fields for
    'sourcetype=splunk_resource_usage'.
  * The Splunk software processes source-type-scoped wildcard expressions
    before it processes source type aliases.
  * Source-type-scoped wildcard expressions require
  'indexed_fields_expansion = t' in limits.conf.
* Follow the stanza name with any number of the following attribute/value
  pairs.

# 'TOKENIZER' enables you to indicate that a field value is a smaller part of a
# token. For example, your raw event has a field with the value "abc123", but
# you need this field to to be a multivalue field with both "abc" and "123" as
# values.
TOKENIZER = <regular expression>
* A regular expression that indicates how the field can take on multiple values
  at the same time.
* Use this setting to configure multivalue fields. Refer to the online
  documentation for multivalue fields.
* If empty, the field can only take on a single value.
* Otherwise, the first group is taken from each match to form the set of
  values.
* This setting is used by the "search" and "where" commands, the summary and
  XML outputs of the asynchronous search API, and by the "top", "timeline", and
  "stats" commands.
* Tokenization of indexed fields is not supported. If "INDEXED = true",
  the tokenizer attribute will be ignored.
* No default.

INDEXED = <boolean>
* Indicates whether a field is created at index time or search time.
* Set to "true" if the field is created at index time.
* Set to "false" for fields extracted at search time. This accounts for the
  majority of fields.
* Default: false

INDEXED_VALUE = [true|false|<sed-cmd>|<simple-substitution-string>]
* Set to "true" if the value is in the raw text of the event.
* Set to "false" if the value is not in the raw text of the event.
* Setting this to "true" expands any search for "key=value"
  into a search for value AND key=value
  since value is indexed.
* For advanced customization, this setting supports sed style substitution.
  For example, 'INDEXED_VALUE=s/foo/bar/g'
  takes the value of the field, replaces all instances of 'foo' with 'bar,'
  and uses that new value as the value to search in the index.
* This setting also supports a simple substitution based on looking for the
  literal string '<VALUE>' (including the '<' and '>' characters).
  For example, 'INDEXED_VALUE=source::*<VALUE>*'
  takes a search for 'myfield=myvalue'
  and searches for 'source::*myvalue*'
  in the index as a single term.
* For both substitution constructs, if the resulting string starts with a '[',
  Splunk interprets the string as a Splunk LISPY expression.  For example,
  'INDEXED_VALUE=[OR <VALUE> source::*<VALUE>]'
  turns 'myfield=myvalue'
  into applying the LISPY expression '[OR myvalue source::*myvalue]'
  (meaning it matches either 'myvalue' or 'source::*myvalue' terms).
* NOTE: You only need to set 'indexed_value' if "indexed = false".
* Default: true
```

fields.conf.example

```
#   Version 9.2.2
#
# This file contains an example fields.conf.  Use this file to configure
# dynamic field extractions.
#
# To use one or more of these configurations, copy the configuration block into
# fields.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# These tokenizers result in the values of To, From and Cc treated as a list,
# where each list element is an email address found in the raw string of data.

[To]
TOKENIZER = (\w[\w\.\-]*@[\w\.\-]*\w)

[From]
TOKENIZER = (\w[\w\.\-]*@[\w\.\-]*\w)

[Cc]
TOKENIZER = (\w[\w\.\-]*@[\w\.\-]*\w)
```

global-banner.conf

The following are the spec and example files for `global-banner.conf`.

global-banner.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure a global banner at the top of every page in Splunk, above the Splunk bar.
#
# Each stanza controls different search commands settings.
#
# There is a global-banner.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name global-banner.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see global-banner.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
```

```
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[BANNER_MESSAGE_SINGLETON]

```
* IMPORTANT: It is only possible to declare one global banner. This is the only
  stanza that Splunk Web will read.

global_banner.visible = <bool>
* Default: false

global_banner.message = <string>
* Default: Sample banner notification text. Please replace with your own message.

global_banner.background_color = [green|blue|yellow|orange|red]
* Default: blue

global_banner.hyperlink = [http://<string>|https://<string>]
* Default: none

global_banner.hyperlink_text = <string>
* Default: none
```

global-banner.conf.example

```
#   Version 9.2.2
#
# The following are example global-banner.conf configurations. Configure properties for
# your custom application.
#
# To use one or more of these configurations, copy the configuration block into
# app.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[BANNER_MESSAGE_SINGLETON]
global_banner.visible = false
global_banner.message = Sample banner notification text. Please replace with your own message.
global_banner.background_color = blue
global_banner.hyperlink = https://www.splunk.com/
global_banner.hyperlink_text = Splunk
```

health.conf

The following are the spec and example files for `health.conf`.

health.conf.spec

```
#   Version 9.2.2
#
# This file sets the default thresholds for Splunk Enterprise's built
# in Health Report.
```

```
#
# Feature stanzas contain indicators, and each indicator has two thresholds:
# * Yellow: Indicates something is wrong and should be investigated.
# * Red: Means that the indicator is effectively not working.
#
# There is a health.conf in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name health.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[distributed_health_reporter]

```
disabled = <boolean>
* Whether or not this Splunk platform instance calls connected search peers to
  retrieve health report information.
* A value of 1 disables the distributed health report on this Splunk platform
  instance. When disabled, the instance does not call connected search peers
  to retrieve health report information.
* Default: 0 (enabled)
```

[health_reporter]

```
full_health_log_interval = <number>
* The amount of time, in seconds, that elapses between each 'PeriodicHealthReporter=INFO' log entry.
* Default: 30.

suppress_status_update_ms = <number>
* The minimum amount of time, in milliseconds, that must elapse between an
  indicator's health status changes.
* Changes that occur earlier will be suppressed.
* Default: 300.

latency_tracker_log_interval = <number>
* The amount of time, in seconds, that elapses between each latency tracker log entry.
* Default: 30.

aggregate_ingestion_latency_health = [0|1]
* A value of 0 disables the aggregation feature for ingestion latency health reporter.
* Default: 1 (enabled).

ingestion_latency_send_interval = <integer>
* The amount of time, in seconds, that splunkd waits before it sends ingestion
  latency data as part of a heartbeat message.
* splunkd determines the actual interval at which it sends this data by factoring
  the value for 'ingestion_latency_send_interval' with the value for 'heartbeatFrequency' in
  the [tcpout] stanza of the outputs.conf file. This is because splunkd uses the
  tcpout heartbeat to send ingestion latency data, and that it won't send ingestion latency
  data at a frequency of less than outputs.conf:[tcpout].'heartbeatFrequency' seconds.
  * If you set 'ingestion_latency_send_interval' to a value that is higher than
    'heartbeatFrequency', splunkd sends that data
    only when the number of 'heartbeatFrequency' seconds exceeds the number of
```

```
    'ingestion_latency_send_interval' seconds at each
    'ingestion_latency_send_interval'.
  * For example: if 'ingestion_latency_send_interval' has a value of 75 and
    'heartbeatFrequency' has a value of 60, splunkd sends the data every
    120 seconds, because it takes two periods of 'heartbeatFrequency'
    seconds before the 'heartbeatFrequency' is greater than the
    'ingestion_latency_send_interval'.
  * Conversely, if you set 'ingestion_latency_send_interval' to a value that is lower than
    'heartbeatFrequency', splunkd sends that data only when the number of
    'ingestiona_latency_send_interval' seconds has elapsed.
  * If, for example, 'ingestion_latency_send_interval' has a value of 30 and
    'heartbeatFrequency' has a value of 90, splunkd sends the data every
    90 seconds because of the value of 'heartbeatFrequency', even though you set a
    'ingestion_latency_send_interval' of 30.
* Default: 30

ingestion_latency_send_interval_max = <number>
* The maximum amount of time, in seconds, that elapses between ingestion latency sent as part of heart beat
message. Should be in range 0-86400
* Default: 86400.

alert.disabled = [0|1]
* A value of 1 disables the alerting feature for health reporter.
* If the value is set to 1, alerting for all features is disabled.
* Default: 0 (enabled)

alert.actions = <string>
* The alert actions that will run when an alert is fired.

alert.min_duration_sec = <integer>
* The minimum amount of time, in seconds, that the health status color must
  persist within threshold_color before triggering an alert.
* Default: 60.

alert.threshold_color = [yellow|red]
* The health status color that will trigger an alert.
* Default: red.

alert.suppress_period = <integer>[m|s|h|d]
* The minimum amount of time, in [minutes|seconds|hours|days], that must
  elapse between each fired alert.
* Alerts that occur earlier will be sent as a batch after this time period
  elapses.
* Default: 10m
```

[clustering]

```
health_report_period = <number>
* The amount of time, in seconds, that elapses between each Clustering
  health report run.
* Default: 20.

disabled = <boolean>
* Whether or not the clustering feature health check is disabled.
* A value of 1 disables the clustering feature health check.
* Default: 0 (enabled)
```

[tree_view:health_subset]

```
* Defines a tree view for health features.
* Users with 'list_health_subset' capability can view features belonging
  to this tree view.
* Users with 'edit_health_subset' capability can edit thresholds for features
  belonging to this tree view.
```

[feature:]*

```
suppress_status_update_ms = <number>
* The minimum amount of time, in milliseconds, that must elapse between an indicator's
  health status changes.
* Changes that occur earlier will be suppressed.
* Default: 300.

display_name = <string>
* A human readable name for the feature.

distributed_disabled = <boolean>
* Whether or not the distributed health report (DHR) tree view includes
  information about this feature.
* A value of "true" means that the DHR does not include this feature in
  its tree view, which means you won't see it when you open the Health Report
  in Splunk Web.
  * This value doesn't apply to the ability of the feature to
    generate alerts, as appropriate.
* A value of "false" means that the DHR includes this feature in
  its tree view.
* Default: 0

snooze_end_time = <number>
* Determines the snooze end time, in seconds since the epoch (Unix time), for this feature.
  Specifying a value for this setting enables a snooze period that suppresses color changes
  for a feature until the <snooze_end_time>.
* A value of 0 disables snoozing for this feature.
* Default = 0

alert.disabled = <boolean>
* Whether or not alerting is disabled for this feature.
* A value of 1 disables alerting for this feature.
* If alerting is disabled in the [health_reporter] stanza, alerting for this feature is disabled,
  regardless of the value set here.
* Otherwise, if the value is set to 1, alerting for all indicators is disabled.
* Default: 0 (enabled)

alert.min_duration_sec = <integer>
* The minimum amount of time, in seconds, that the health status color must
  persist within threshold_color before triggering an alert.

alert.threshold_color = [yellow|red]
* The health status color to trigger an alert.
* Default: red.

friendly_description = <string>
* A general description to help the user determine what functionality is monitored
  by the heath report indicator.

indicator:<indicator name>:friendly_description = <string>
```

```
* A general description of the technical behavior monitored by the indicator.
  Use common terminology that a user can search on to find documentation,
  details, or troubleshooting guidance.

indicator:<indicator name>:description = <string>
* Description of this indicator to help users to make basic decisions such as:
  Turning indicators on or off
  Adjusting the threshold of an indicator
  Turning on alerting for an indicator

indicator:<indicator name>:<indicator color> = <number>
* There are various indicator names. See your health.conf for the complete list.
* There are two valid colors: yellow and red.
* These settings should not be adjusted lightly. If the numbers are set too
  high, you might inadvertently mask serious errors that the Health Report is
  trying to bring to your attention.

alert:<indicator name>.disabled = [0|1]
* A value of 1 disables alerting for this indicator.
* Default: 0 (enabled)

alert:<indicator name>.min_duration_sec = <integer>
* The minimum amount of time, in seconds, that the health status color must
  persist within threshold_color before triggering an alert.

alert:<indicator name>.threshold_color = [yellow|red]
* The health status color to trigger an alert.

tree_view:health_subset = [enabled | disabled]
* Indicates that this feature belongs to the 'health_subset' tree view.
```

[alert_action:]*

```
disabled = [0|1]
* A value of 1 disables this alert action.
* Default: 0 (enabled)

action.<action parameter> = <string>
* There are various parameters for different alert actions.
* Each value defines one parameter for the alert action.

* NOTE: [feature:master_connectivity], [feature:slave_state]
*       feature:slave_version] stanzas are now DEPRECATED.
```

health.conf.example

```
#   Version 9.2.2
#
# This file contains an example health.conf.  Use this file to configure thresholds
# for Splunk Enterprise's built in Health Report.
#
# To use one or more of these configurations, copy the configuration block
# into health.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.

[health_reporter]
# Every 30 seconds a new 'PeriodicHealthReporter=INFO' log entry will be created.
full_health_log_interval = 30
```

```
# If an indicator's health status changes before 600 milliseconds elapses,
# the status change will be suppressed.
suppress_status_update_ms = 600
# Alerting for all features is enabled.
# You can disable alerting for each feature by setting 'alert.disabled' to 1.
alert.disabled = 0

# If you don't want to send alerts too frequently, you can define a minimum
# time period that must elapse before another alert is fired. Alerts triggered
# during the suppression period are sent after the period expires as a batch.
# The suppress_period value can be in seconds, minutes, hours, and days, and
# uses the format: 60s, 60m, 60h and 60d.
# Default is 10 minutes.
alert.suppress_period = 30m


[alert_action:email]
# Enable email alerts for the health report.
# Before you can send an email alert, you must configure the email notification
# settings on the email settings page.
# In the 'Search and Reporting' app home page, click Settings > Server settings
# > Email settings, and specify values for the settings.
# After you configure email settings, click Settings > Alert actions.
# Make sure that the 'Send email' option is enabled.
disabled = 0

# Define recipients when an email alert is triggered.
# You can define 'to', 'cc', and 'bcc' recipients.
# For multiple recipients in a list, separate email addresses with commas.
# If there is no recipient for a certain recipient type (e.g. bcc), leave the value blank.
action.to = admin_1@testcorp.example, admin_2@testcorp.example
action.cc = admin_3@testcorp.example, admin_4@testcorp.example
action.bcc =

[alert_action:pagerduty]
# Enable Pager Duty alerts for the health report.
# Before you can send an alert to PagerDuty, you must configure some settings
# on both the PagerDuty side and the Splunk Enterprise side.
# In PagerDuty, you must add a service to save your new integration.
# From the Integrations tab of the created service, copy the Integration Key
# string to the 'action.integration_url_override' below.
# On the Splunk side, you must install the PagerDuty Incidents app from
# Splunkbase.
# After you install the app, in Splunk Web, click Settings > Alert actions.
# Make sure that the PagerDuty app is enabled.
disabled = 0
action.integration_url_override = 12345678901234567890123456789ab

[alert_action:mobile]
# Enable Splunk Mobile alerts for the health report.
# You need to configure the 'alert_recipients' under this stanza in order to
# send health report alerts to the Splunk Mobile app on your phone.
#
# Steps to setup the health report mobile alert:
# * Download the Splunk Mobile App on your phone and open the app.
# * Download the Cloud Gateway App from Splunkbase to your splunk instance.
# * In Splunk Web, click Settings > Alert actions and make sure the Cloud
#   Gateway App is enabled.
# * In Splunk Web, click Cloud Gateway App > Configure and enable Splunk
#   Mobile.
# * In Splunk Web, click Cloud Gateway App > Register and copy the activation
#   code displayed in the Splunk Mobile App to register your device(phone).
```

```
# * In health.conf configure 'alert_recipients' under the [alert_action:mobile]
#   stanza, e.g. action.alert_recipients = admin
#
# Details of how to install and use the Cloud Gateway App please refer to
# https://docs.splunk.com/Documentation/Gateway
disabled = 0
action.alert_recipients = admin

[alert_action:victorops]
# Enable VictorOps alerts for the health report.
# Before you can send an alert to VictorOps, you must configure some settings
# on both the VictorOps side and the Splunk Enterprise side.
# In VictorOps, you must create an API key and can optionally create a routing key.
# On the Splunk side, you must install the VictorOps App from Splunkbase.
# After you install the app, in Splunk Web, click Settings > Alert actions.
# Make sure that the VictorOps app is enabled and the API key is properly configured.
disabled = 0
# alert message type in VictorOps.
# Valid alert message types in VictorOps:
#  * CRITICAL - Triggers an incident.
#  * WARNING - May trigger an incident, depending on your settings in VictorOps.
#  * ACKNOWLEDGEMENT - Acknowledges an incident. This value is unlikely to be useful.
#  * INFO - Creates a timeline event, but does not trigger an incident.
#  * RECOVERY - Resolves an incident. This value is unlikely to be useful.
action.message_type = CRITICAL
# ID of the incident in VictorOps.
* Optional.
action.entity_id =
# Use this field to choose one of the API keys configured in passwords.conf
# under victorops_app.
# Leave this field empty if you want to use the default API key.
* Optional.
action.record_id =
# Use this field to overwrite the default routing key.
* Optional.
action.routing_key_override =

[clustering]
# Clustering health report will run in every 20 seconds.
health_report_period = 20
# Enable the clustering feature health check.
disabled = 0

[feature:s2s_autolb]
# If more than 20% of forwarding destinations have failed, health status changes to yellow.
indicator:s2s_connections:yellow = 20
# If more than 70% of forwarding destinations have failed, health status changes to red.
indicator:s2s_connections:red = 70
# Alerting for all indicators is disabled.
alert.disabled = 1

[feature:batchreader]
# Enable alerts for feature:batchreader. If there is no 'alert.disabled' value
# specified in a feature stanza, then the alert is enabled for the feature by
# default.
# You can also enable/disable alerts at the indicator level, using the setting:
# 'alert:<indicator name>.disabled'.
alert.disabled = 0

# You can define which color triggers an alert.
# If the value is yellow, both yellow and red trigger an alert.
# If the value is red, only red triggers an alert.
```

```
# Default value is red.
# You can also define the threshold_color for each indicator using the setting:
# 'alert:<indicator name>.threshold_color'.
# Indicator level setting overrides the feature level threshold_color setting.
alert.threshold_color = red

# You can define the duration that an unhealthy status persists before the alert fires.
# Default value is 60 seconds.
# You can also define the min_duration_sec for each indicator using the setting:
# 'alert:<indicator name>.min_duration_sec'.
# Indicator level setting overrides feature level min_duration_sec setting.
alert.min_duration_sec = 30

# Suppresses color changes for this feature until March 25, 2021 8:00:00 PM GMT.
snooze_end_time = 1616702400
```

indexes.conf

The following are the spec and example files for `indexes.conf`.

indexes.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains all possible options for an indexes.conf file.  Use
# this file to configure Splunk's indexes and their properties.
#
# Each stanza controls different search commands settings.
#
# There is a indexes.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name indexes.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see indexes.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# Some settings changes might require a restart or reload. To determine when a
# restart or reload is required, refer to the "Managing Indexers and
# Clusters of Indexers" documentation:
# http://docs.splunk.com/Documentation/Splunk/latest/Indexer/Determinerestart
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# CAUTION: You can drastically affect your Splunk installation by changing
# these settings. Consult technical support
# (http://www.splunk.com/page/submit_issue) if you are not sure how to
# configure this file.
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top
#     of the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file wins.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

sync = <nonnegative integer>
* The index processor syncs events every 'sync' number of events.
* Set to 0 to disable.
* Highest legal value is 32767.
* Default: 0

defaultDatabase = <index name>
* If an index is not specified during search, Splunk software
  searches the default index.
* The specified index displays as the default in Splunk Manager settings.
* Default: main

bucketMerging = <boolean>
* This setting is supported on indexer clusters when 'storageType' is "remote" or "local".
  Standalone indexers support "local" only.
* The bucket merge task will evaluate and localize remote buckets before merging.
* Set to true to enable bucket merging service on all indexes
* You can override this value on a per-index basis.
* Default: false

bucketMerge.minMergeSizeMB = <unsigned integer>
* This setting is supported on indexer clusters when 'storageType' is "remote" or "local".
  Standalone indexers support "local" only.
* Minimum cumulative bucket sizes to merge.
* You can override this value on a per-index basis.
* Default: 750

bucketMerge.maxMergeSizeMB = <unsigned integer>
* This setting is supported on indexer clusters when 'storageType' is "remote" or "local".
  Standalone indexers support "local" only.
* Maximum cumulative bucket sizes to merge.
* You can override this value on a per-index basis.
* Default: 1000

bucketMerge.maxMergeTimeSpanSecs = <unsigned integer>
* This setting is supported on indexer clusters when 'storageType' is "remote" or "local".
  Standalone indexers support "local" only.
* Maximum allowed time span, in seconds, between buckets about to be merged.
* You can override this value on a per-index basis.
* Default: 7776000 (90 days)

bucketMerge.minMergeCount = <unsigned integer>
* This setting is supported on indexer clusters when 'storageType' is "remote" or "local".
  Standalone indexers support "local" only.
* Minimum number of buckets to merge.
* You can override this value on a per-index basis.
* Default: 2
```

```
bucketMerge.maxMergeCount = <unsigned integer>
* This setting is supported on indexer clusters when 'storageType' is "remote" or "local".
  Standalone indexers support "local" only.
* Maximum number of buckets to merge.
* You can override this value on a per-index basis.
* Default: 24

queryLanguageDefinition = <path to file>
* DO NOT EDIT THIS SETTING. SERIOUSLY.
* The path to the search language definition file.
* Default: $SPLUNK_HOME/etc/searchLanguage.xml.

lastChanceIndex = <index name>
* An index that receives events that are otherwise not associated
  with a valid index.
* If you do not specify a valid index with this setting, such events are
  dropped entirely.
* Routes the following kinds of events to the specified index:
  * events with a non-existent index specified at an input layer, like an
    invalid "index" setting in inputs.conf
  * events with a non-existent index computed at index-time, like an invalid
    _MetaData:Index value set from a "FORMAT" setting in transforms.conf
* You must set 'lastChanceIndex' to an existing, enabled index.
  Splunk software cannot start otherwise.
* If set to "default", then the default index specified by the
  'defaultDatabase' setting is used as a last chance index.
* Default: empty string

malformedEventIndex = <index name>
* Currently not supported. This setting is related to a feature that is
  still under development.
* An index to receive malformed events.
* If you do not specify a valid index with this setting, or Splunk software
  cannot use the index specified in the 'defaultDatabase' setting,
  such events are dropped entirely.
* Routes the following kinds of events to the specified index:
    * events destined for read-only indexes
    * log events destined for datatype=metric indexes
    * log events with invalid raw data values, like all-whitespace raw
    * metric events destined for datatype=event indexes
    * metric events with invalid metric values, like non-numeric values
    * metric events lacking required attributes, like metric name
* Malformed events can be modified in order to make them suitable for
  indexing, as well as to aid in debugging.
* A high volume of malformed events can affect search performance against
  the specified index; for example, malformed metric events can lead to an
  excessive number of Strings.data entries
* <index name> must refer to an existing, enabled index. Splunk software
  does not start if this is not the case.
* If set to "default", the indexer places malformed events in the index
  specified by the 'defaultDatabase' setting
* Default: empty string

memPoolMB = <positive integer>|auto
* Determines how much memory is given to the indexer memory pool. This
  restricts the number of outstanding events in the indexer at any given
  time.
* Must be greater than 0; maximum value is 1048576 (which corresponds to 1 TB)
* Setting this too high can cause splunkd memory usage to increase
  significantly.
* Setting this too low can degrade splunkd indexing performance.
* Setting this to "auto" or an invalid value causes splunkd to autotune
```

```
  the value as follows:
    * System Memory Available less than ... | 'memPoolMB'
                   1 GB                     |    64  MB
                   2 GB                     |   128 MB
                   8 GB                     |   128 MB
                  16 GB                     |   256 MB
                  32 GB                     |     1 GB
                  64 GB                     |     2 GB
                  64 GB or higher           |     4 GB
* Only set this value if you are an expert user or have been advised to by
  Splunk Support.
* CAUTION: CARELESSNESS IN SETTING THIS CAN LEAD TO LOSS OF JOB.
* Default: auto

indexThreads = <nonnegative integer>|auto
* Determines the number of threads to use for indexing.
* Must be at least 1 and no more than 16.
* This value should not be set higher than the number of processor cores in
  the machine.
* If splunkd is also doing parsing and aggregation, the number should be set
  lower than the total number of processors minus two.
* Setting this to "auto" or an invalid value will cause Splunk to autotune
  this setting.
* Only set this value if you are an expert user or have been advised to by
  Splunk Support.
* CAUTION: CARELESSNESS IN SETTING THIS CAN LEAD TO LOSS OF JOB.
* Default: auto

rtRouterThreads = 0|1
* Set to "1" if you expect to use non-indexed real time searches regularly. Index
  throughput drops rapidly if there are a handful of these running concurrently
  on the system.
* If you are not sure what "indexed vs non-indexed" real time searches are, see
  README of indexed_realtime* settings in limits.conf
* NOTE: This is not a boolean value. Acceptable values are "0" and "1" ONLY.
  At the present time, you can only create a single real-time thread per
  pipeline set.

rtRouterQueueSize = <positive integer>
* This setting is only valid if 'rtRouterThreads' != 0
* This queue sits between the indexer pipeline set thread (producer) and the
  'rtRouterThread'
* Changing the size of this queue can impact real-time search performance.
* Default: 10000

selfStorageThreads = <positive integer>
* Specifies the number of threads used to transfer data to customer-owned remote
  storage.
* The threads are created on demand when any index is configured with
  self storage options.
* Default: 2

assureUTF8 = <boolean>
* Verifies that all data retrieved from the index is proper by validating
  all the byte strings.
  * This does not ensure all data will be emitted, but can be a workaround
    if an index is corrupted in such a way that the text inside it is no
    longer valid utf8.
* Will degrade indexing performance when enabled (set to true).
* Can only be set globally, by specifying in the [default] stanza.
* Default: false
```

```
enableRealtimeSearch = <boolean>
* Enables real-time searches.
* Default: true

suppressBannerList = <comma-separated list of strings>
* suppresses index missing warning banner messages for specified indexes
* Default: empty string

maxRunningProcessGroups = <positive integer>
* splunkd runs helper child processes like "splunk-optimize",
  "recover-metadata", etc. This setting limits how many child processes
  can run at any given time.
* This maximum applies to all of splunkd, not per index. If you have N
  indexes, there will be at most 'maxRunningProcessGroups' child processes,
  not N * 'maxRunningProcessGroups' processes.
* Must maintain maxRunningProcessGroupsLowPriority < maxRunningProcessGroups
* This is an advanced setting; do NOT set unless instructed by Splunk
  Support.
* Highest legal value is 4294967295.
* Default: 8

maxRunningProcessGroupsLowPriority = <positive integer>
* Of the 'maxRunningProcessGroups' helper child processes, at most
  'maxRunningProcessGroupsLowPriority' may be low-priority
  (for example, "fsck") ones.
* This maximum applies to all of splunkd, not per index. If you have N
  indexes, there will be at most 'maxRunningProcessGroupsLowPriority'
  low-priority child processes, not N * 'maxRunningProcessGroupsLowPriority'
  processes.
* There must always be fewer 'maxRunningProcessGroupsLowPriority' child
  processes than there are 'maxRunningProcessGroups' child processes.
* This is an advanced setting; do NOT set unless instructed by Splunk
  Support.
* Highest legal value is 4294967295.
* Default: 1

bucketRebuildMemoryHint = <positive integer>[KB|MB|GB]|auto
* A suggestion for the bucket rebuild process for the size, in bytes,
  of the tsidx file it will try to build.
* Larger files use more memory in a rebuild, but rebuilds fail if there is
  not enough memory.
* Smaller files make the rebuild take longer during the final optimize step.
* NOTE: This value is not a hard limit on either rebuild memory usage or
  tsidx size.
* This is an advanced setting, do NOT set this unless instructed by Splunk
  Support.
* If set to "auto", the bucket rebuild process tunes the setting based on
  the amount of physical RAM on the machine:
  *  less than 2GB RAM = 67108864 (64MB) tsidx
  *  2GB to 8GB RAM = 134217728 (128MB) tsidx
  *  more than 8GB RAM = 268435456 (256MB) tsidx
* If not set to "auto", then you must set this setting between 16MB and 1GB.
* A value can be specified using a size suffix: "16777216" or "16MB" are
  equivalent.
* Inappropriate use of this setting causes splunkd to not start if
  rebuild is required.
* Highest legal value (in bytes) is 4294967295.
* Default: auto

inPlaceUpdates = <boolean>
* Whether or not splunkd writes metadata updates to .data files in place.
* Intended for advanced debugging of metadata issues.
```

```
* If set to "true", metadata updates are written to the .data files directly.
* If set to "false", metadata updates are written to a temporary file and
  then moved into place.
* Configuring this setting to "false" (to use a temporary file) affects
  indexing performance, particularly with large numbers of hosts, sources,
  or sourcetypes (~1 million, across all indexes.)
* This is an advanced setting; do NOT set unless instructed by Splunk
  Support
* Default: true

serviceInactiveIndexesPeriod = <positive integer>
* How frequently, in seconds, inactive indexes are serviced.
* An inactive index is an index that has not been written to for a period
  greater than the value of 'serviceMetaPeriod'.  The inactive state is not
  affected by whether the index is being read from.
* The highest legal value is 4294967295.
* Default: 60

serviceOnlyAsNeeded = <boolean>
* DEPRECATED; use 'serviceInactiveIndexesPeriod' instead.
* Causes index service (housekeeping tasks) overhead to be incurred only
  after index activity.
* Indexer module problems might be easier to diagnose when this optimization
  is disabled (set to false).
* Default: true

serviceSubtaskTimingPeriod = <positive integer>
* Subtasks of indexer service task will be timed on every Nth execution,
  where N = value of this setting, in seconds.
* Smaller values give greater accuracy; larger values lessen timer
  overhead.
* Timer measurements are found in metrics.log, marked
  "group=subtask_seconds, task=indexer_service"
* Highest legal value is 4294967295
* Configure a value for this setting that divides evenly into the value for
  the 'rotatePeriodInSecs' setting where possible.
* Default: 30

processTrackerServiceInterval = <nonnegative integer>
* How often, in seconds, the indexer checks the status of the child OS
  processes it has launched to see if it can launch new processes for queued
  requests.
* If set to 0, the indexer checks child process status every second.
* Highest legal value is 4294967295.
* Default: 1

maxBucketSizeCacheEntries = <nonnegative integer>
* This value is no longer needed. Its value is ignored.

tsidxStatsHomePath = <string>
* An absolute path that specifies where the indexer creates namespace data
  with the 'tscollect' command.
* If the directory does not exist, the indexer attempts to create it.
* Optional.
* NOTE: The "$SPLUNK_DB" directory must be writable.
* Default: $SPLUNK_DB/tsidxstats

tsidxWritingLevel = [1|2|3|4]
* Enables various performance and space-saving improvements for tsidx files.
* Tsidx files written with a higher tsidxWritingLevel setting have limited backward
  compatibility when searched with lower versions of Splunk Enterprise.
* Setting tsidxWritingLevel globally is recommended. It can also be set per-index.
```

```
* For deployments that have multi-site index clustering, change the setting AFTER
  all your indexers in the cluster have been upgraded to the latest release.
* Default: 3

hotBucketTimeRefreshInterval = <positive integer>
* How often each index refreshes the available hot bucket times
  used by the 'indexes' REST endpoint.
* A refresh occurs every N times service is performed for each index.
  * For busy indexes, this is a multiple of seconds.
  * For idle indexes, this is a multiple of the second-long-periods in
    which data is received.
* This setting is only intended to relax the frequency of these refreshes in
  the unexpected case that it adversely affects performance in unusual
  production scenarios.
* This time is tracked on a per-index basis, and thus can be adjusted
  on a per-index basis if needed.
* If you want the index information to be refreshed with
  every service (and accept minor performance overhead), set to 1.
* Default: 10 (services)

fileSystemExecutorWorkers = <positive iinteger>
* Determines the number of threads to use for file system io operations.
* This maximum applies to all of splunkd, not per index. If you have N
  indexes, there will be at most 'fileSystemExecutorWorkers' workers,
  not N * 'fileSystemExecutorWorkers' workers.
* This is an advanced setting; do NOT set unless instructed by Splunk
  Support.
* Highest legal value is 4294967295.
* Default: 5

hotBucketStreaming.extraBucketBuildingCmdlineArgs = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Default: empty
```

PER INDEX OPTIONS

```
# These options can be set under an [<index>] entry.
#
# Index names must consist of only numbers, lowercase letters, underscores,
# and hyphens. They cannot begin with an underscore or hyphen, or contain
# the word "kvstore".
#*******************************************************************************

disabled = <boolean>
* Toggles your index entry off and on.
* Set to "true" to disable an index.
* CAUTION: Do not set this setting to "true" on remote storage enabled indexes.
* Default: false

deleted = true
* If present, means that this index has been marked for deletion: if splunkd
  is running, deletion is in progress; if splunkd is stopped, deletion
  re-commences on startup.
* Do NOT manually set, clear, or modify the value of this setting.
* CAUTION: Seriously: LEAVE THIS SETTING ALONE.
* No default.

deleteId = <nonnegative integer>
```

```
* If present, means that this index has been marked for deletion: if splunkd
  is running, deletion is in progress; if splunkd is stopped, deletion
  re-commences on startup.
* Do NOT manually set, clear, or modify the value of this setting.
* CAUTION: Seriously: LEAVE THIS SETTING ALONE.
* No default.

homePath = <string>
* An absolute path that contains the hot and warm buckets for the index.
* Best practice is to specify the path with the following syntax:
     homePath = $SPLUNK_DB/$_index_name/db
  At runtime, splunkd expands "$_index_name" to the name of the index. For example,
  if the index name is "newindex", homePath becomes
     "$SPLUNK_DB/newindex/db".
* Splunkd keeps a file handle open for warmdbs at all times.
* Can contain a volume reference (see volume section below) in place of $SPLUNK_DB.
* CAUTION: The parent path "$SPLUNK_DB/$_index_name/" must be writable.
* Required. Splunkd does not start if an index lacks a valid 'homePath'.
* You must restart splunkd after changing this setting for the changes to take effect.
* Avoid the use of other environment variables in index paths, aside from the possible
  exception of SPLUNK_DB.
  * As an exception, SPLUNK_DB is explicitly managed by the software,
    so most possible downsides here do not exist.
  * Environment variables can be different from launch to launch of the
    software, causing severe problems with management of indexed data,
    including:
    * Data in the prior location is not searchable.
    * The indexer might not be able to write to the new location, causing outages
      or data loss.
    * Writing to a new, unexpected location could lead to disk space exhaustion
      causing additional operational problems.
    * Recovery from such a scenario requires manual intervention and bucket
      renaming, especially difficult in an index cluster environment.
    * In all circumstances, Splunk Diag, the diagnostic tool that Splunk Support
      uses, has no way to determine the correct values for the environment
      variables, and cannot reliably operate. You might need to manually acquire
      information about your index buckets in troubleshooting scenarios.
  * Volumes provide a more appropriate way to control the
    storage location for indexes.
* No default.

coldPath = <string>
* An absolute path that contains the colddbs for the index.
* Best practice is to specify the path with the following syntax:
     coldPath = $SPLUNK_DB/$_index_name/colddb
  At runtime, splunkd expands "$_index_name" to the name of the index. For example,
  if the index name is "newindex", 'coldPath'
  becomes "$SPLUNK_DB/newindex/colddb".
* Cold databases are opened as needed when searching.
* Can contain a volume reference (see volume section below) in place of $SPLUNK_DB.
* Path must be writable.
* Required. Splunkd does not start if an index lacks a valid 'coldPath'.
* You must restart splunkd after changing this setting for the changes to
  take effect. Reloading the index configuration does not suffice.
* Avoid using environment variables in index paths, aside from the
  possible exception of $SPLUNK_DB. See 'homePath' for additional
  information as to why.
* Remote-storage-enabled indexes do not cycle buckets from homePath to coldPath.
  However, if buckets already reside in 'coldPath' for a
  non-remote-storage-enabled index, and that index is later enabled for remote
  storage, those buckets will be searchable and will have their life cycle
  managed.
```

```
thawedPath = <string>
* An absolute path that contains the thawed (resurrected) databases for the
  index.
* CANNOT contain a volume reference.
* Path must be writable.
* Required. Splunkd does not start if an index lacks a valid thawedPath.
* You must restart splunkd after changing this setting for the changes to
  take effect. Reloading the index configuration does not suffice.
* Avoid the use of environment variables in index paths, aside from the
  exception of SPLUNK_DB. See 'homePath' for additional information as
  to why.

bloomHomePath = <string>
* The location where the bloomfilter files for the index are stored.
* If specified, 'bloomHomePath' must be defined in terms of a volume definition
  (see volume section below).
* If 'bloomHomePath' is not specified, the indexer stores bloomfilter files
  for the index inline, inside index bucket directories.
* Path must be writable.
* You must restart splunkd after changing this setting for the
  changes to take effect. Reloading the index configuration does
  not suffice.
* Avoid the use of environment variables in index paths, aside from the
  exception of SPLUNK_DB.  See 'homePath' for additional information
  as to why.
* CAUTION: Do not set this setting on indexes that have been
  configured to use remote storage with the "remotePath" setting.

createBloomfilter = <boolean>
* Whether or not to create bloomfilter files for the index.
* If set to "true", the indexer creates bloomfilter files.
* If set to "false", the indexer does not create bloomfilter files.
* You must set to "true" for remote storage enabled indexes.
* CAUTION: Do not set this setting to "false" on indexes that have been
  configured to use remote storage with the "remotePath" setting.
* Default: true

summaryHomePath = <string>
* An absolute path where transparent summarization results for data in this
  index should be stored.
* This value must be different for each index and can be on any disk drive.
* Best practice is to specify the path with the following syntax:
     summaryHomePath = $SPLUNK_DB/$_index_name/summary
  At runtime, splunkd expands "$_index_name" to the name of the index.
  For example, if the index name is "newindex", summaryHomePath becomes
  "$SPLUNK_DB/newindex/summary".
* Can contain a volume reference (see volume section below) in place of $SPLUNK_DB.
* Volume reference must be used if you want to retain data based on data size.
* Path must be writable.
* If not specified, splunkd creates a directory 'summary' in the same
  location as 'homePath'.
  * For example, if 'homePath' is "/opt/splunk/var/lib/splunk/index1/db",
    then 'summaryHomePath' must be "/opt/splunk/var/lib/splunk/index1/summary".
* The parent path must be writable.
* You must not set this setting for remote storage enabled indexes.
* You must restart splunkd after changing this setting for the
  changes to take effect. Reloading the index configuration does
  not suffice.
* Avoid the use of environment variables in index paths, aside from the
  exception of SPLUNK_DB. See 'homePath' for additional
  information as to why.
```

```
* No default.

tstatsHomePath = <string>
* Location where data model acceleration TSIDX data for this index should be stored.
* Required.
* MUST be defined in terms of a volume definition (see volume section below)
* Path must be writable.
* You must not set this setting for remote storage enabled indexes.
* You must restart splunkd after changing this setting for the
  changes to take effect. Reloading the index configuration does
  not suffice.
* Default: volume:_splunk_summaries/$_index_name/datamodel_summary,
  where "$_index_name" is runtime-expanded to the name of the index

remotePath = <root path for remote volume, prefixed by a URI-like scheme>
* Optional.
* Presence of this setting means that this index uses remote storage, instead
  of the local file system, as the main repository for bucket storage. The
  index processor works with a cache manager to fetch buckets locally, as
  necessary, for searching and to evict them from local storage as space fills
  up and they are no longer needed for searching.
* This setting must be defined in terms of a storageType=remote volume
  definition. See the volume section below.
* The path portion that follows the volume reference is relative to the path
  specified for the volume. For example, if the path for a volume "v1" is
  "s3://bucket/path" and 'remotePath' is "volume:v1/idx1", then the fully
  qualified path is "s3://bucket/path/idx1". The rules for resolving the
  relative path with the absolute path specified in the volume can vary
  depending on the underlying storage type.
* If 'remotePath' is specified, the 'coldPath' and 'thawedPath' settings are
  ignored. However, you must still specify them.

maxBloomBackfillBucketAge = <nonnegative integer>[smhd]|infinite
* If a (warm or cold) bucket with no bloomfilter is older than this,
  splunkd does not create a bloomfilter for that bucket.
* When set to 0, splunkd never backfills bloomfilters.
* When set to "infinite", splunkd always backfills bloomfilters.
* NOTE: If 'createBloomfilter' is set to "false", bloomfilters are never
  backfilled regardless of the value of this setting.
* The highest legal value in computed seconds is 2 billion, or 2000000000, which
  is approximately 68 years.
* Default: 30d

hotlist_recency_secs = <unsigned integer>
* When a bucket is older than this value, it becomes eligible for eviction.
  Buckets younger than this value are evicted only if there are no older
  buckets eligible for eviction.
* Default: The global setting in the server.conf file [cachemanager] stanza

hotlist_bloom_filter_recency_hours = <unsigned integer>
* When a bucket's non-journal and non-tsidx files (such as bloomfilter files)
  are older than this value, those files become eligible for eviction. Bloomfilter
  and associated files younger than this value are evicted only if there are
  no older files eligible for eviction.
* Default: The global setting in the server.conf file [cachemanager] stanza

enableOnlineBucketRepair = <boolean>
* Controls asynchronous "online fsck" bucket repair, which runs concurrently
  with splunkd.
* When enabled, you do not have to wait until buckets are repaired, to start
  splunkd.
* When enabled, you might observe a slight degradation in performance.
```

```
* You must set to "true" for remote storage enabled indexes.
* Default: true

enableDataIntegrityControl = <boolean>
* Whether or not splunkd computes hashes on rawdata slices and stores the hashes
  for future data integrity checks.
* If set to "true", hashes are computed on the rawdata slices.
* If set to "false", no hashes are computed on the rawdata slices.
* Default: false

maxWarmDBCount = <nonnegative integer>
* The maximum number of warm buckets.
* Warm buckets are located in the 'homePath' for the index.
* If set to zero, splunkd does not retain any warm buckets
  It rolls the buckets to cold as soon as it is able.
* Splunkd ignores this setting on remote storage enabled indexes.
* Highest legal value is 4294967295.
* Default: 300

maxTotalDataSizeMB = <nonnegative integer>
* The maximum size of an index, in megabytes.
* If an index grows larger than the maximum size, splunkd freezes the oldest
  data in the index.
* This setting applies only to hot, warm, and cold buckets. It does
  not apply to thawed buckets.
* CAUTION: The 'maxTotalDataSizeMB' size limit can be reached before the time
  limit defined in 'frozenTimePeriodInSecs' due to the way bucket time spans
  are calculated. When the 'maxTotalDataSizeMB' limit is reached, the buckets
  are rolled to frozen. As the default policy for frozen data is deletion,
  unintended data loss could occur.
* Splunkd ignores this setting on remote storage enabled indexes.
* Highest legal value is 4294967295
* Default: 500000

maxGlobalRawDataSizeMB = <nonnegative integer>
* The maximum amount of cumulative raw data (in MB) allowed in a remote
  storage-enabled index.
* This setting is available for both standalone indexers and indexer clusters.
  In the case of indexer clusters, the raw data size is calculated as the total
  amount of raw data ingested for the index, across all peer nodes.
* When the amount of uncompressed raw data in an index exceeds the value of this
  setting, the bucket containing the oldest data is frozen.
* For example, assume that the setting is set to 500 and the indexer cluster
  has already ingested 400MB of raw data into the index, across all peer nodes.
  If the cluster ingests an additional amount of raw data greater than 100MB in
  size, the cluster freezes the oldest buckets, until the size of raw data
  reduces to less than or equal to 500MB.
* This value applies to warm and cold buckets. It does not
  apply to hot or thawed buckets.
* The maximum allowable value is 4294967295.
* Default: 0 (no limit to the amount of raw data in an index)

maxGlobalDataSizeMB = <nonnegative integer>
* The maximum size, in megabytes, for all warm buckets in a SmartStore
  index. If the index was migrated from non-SmartStore to SmartStore this
  size also includes the size of all migrated cold buckets.
* This setting includes the sum of the size of all buckets that reside
  on remote storage, along with any buckets that have recently rolled
  from hot to warm on a peer node and are awaiting upload to remote storage.
* If the total size of the warm buckets in an index exceeds
  'maxGlobalDataSizeMB', the oldest bucket in the index is frozen.
  * For example, assume that 'maxGlobalDataSizeMB' is set to 5000 for
```

```
    an index, and the index's warm buckets occupy 4800 MB. If a 750 MB
    hot bucket then rolls to warm, the index size now exceeds
    'maxGlobalDataSizeMB', which triggers bucket freezing. The cluster
    freezes the oldest buckets on the index, until the total warm bucket
    size falls below 'maxGlobalDataSizeMB'.
* The size calculation for this setting applies on a per-index basis.
* The calculation applies across all peers in the cluster.
* The calculation includes only one copy of each bucket. If a duplicate
  copy of a bucket exists on a peer node, the size calculation does
  not include it.
  * For example, if the bucket exists on both remote storage and on a peer
    node's local cache, the calculation ignores the copy on local cache.
* The calculation includes only the size of the buckets themselves.
  It does not include the size of any associated files, such as report
  acceleration or data model acceleration summaries.
* The highest legal value is 4294967295 (4.2 petabytes.)
* Default: 0 (No limit to the space that the warm buckets on an index can occupy.)

rotatePeriodInSecs = <positive integer>
* Controls the service period (in seconds): how often splunkd performs
  certain housekeeping tasks. Among these tasks are:
  * Check if a new hot DB needs to be created.
  * Check if there are any cold DBs that should be frozen.
  * Check whether buckets need to be moved out of hot and cold DBs, due to
    respective size constraints (i.e., homePath.maxDataSizeMB and
    coldPath.maxDataSizeMB)
* This value becomes the default value of the 'rotatePeriodInSecs' setting
  for all volumes (see 'rotatePeriodInSecs' in the Volumes section)
* The highest legal value is 4294967295.
* Default: 60

frozenTimePeriodInSecs = <nonnegative integer>
* The number of seconds after which indexed data rolls to frozen.
* If you do not specify a 'coldToFrozenScript', data is deleted when rolled to
  frozen.
* NOTE: Every event in a bucket must be older than 'frozenTimePeriodInSecs'
  seconds before the bucket rolls to frozen.
* The highest legal value is 4294967295.
* Default: 188697600 (6 years)

warmToColdScript = <script path>
* Specifies a script to run when moving data from warm to cold buckets.
* This setting is supported for backwards compatibility with versions
  older than 4.0. Migrating data across filesystems is now handled natively
  by splunkd.
* If you specify a script here, the script becomes responsible for moving
  the event data, and Splunk-native data migration is not used.
* The script must accept two arguments:
  * First: the warm directory (bucket) to be rolled to cold.
  * Second: the destination in the cold path.
* If the script you specify is a Python script, it uses the default system-wide
  Python interpreter. You cannot override this configuration with the
  'python.version' setting.
* Searches and other activities are paused while the script is running.
* Contact Splunk Support (http://www.splunk.com/page/submit_issue) if you
  need help configuring this setting.
* The script must be in $SPLUNK_HOME/bin or a subdirectory thereof.
* Splunkd ignores this setting for remote storage enabled indexes.
* Default: empty string

coldToFrozenScript = <path to script interpreter> <path to script>
* Specifies a script to run when data is to leave the splunk index system.
```

```
  * Essentially, this implements any archival tasks before the data is
    deleted out of its default location.
* Add "$DIR" (including quotes) to this setting on Windows (see below
  for details).
* Script Requirements:
  * The script must accept only one argument: An absolute path to the bucket directory
    that is to be archived. The bundle push from a Cluster Manager
    fails if you use more than one argument.
  * In the case of metrics indexes, the script must also accept the flag "--search-files-required",
    to prevent the script from archiving empty rawdata files. For more details, see the entry for the
    "metric.stubOutRawdataJournal" setting.
  * Your script should work reliably.
    * If your script returns success (0), Splunk completes deleting
      the directory from the managed index location.
    * If your script return failure (non-zero), Splunk leaves the bucket
      in the index, and tries calling your script again several minutes later.
    * If your script continues to return failure, this will eventually cause
      the index to grow to maximum configured size, or fill the disk.
  * Your script should complete in a reasonable amount of time.
    * If the script stalls indefinitely, it will occupy slots.
    * This script should not run for long as it would occupy
      resources which will affect indexing.
* If the string $DIR is present in this setting, it will be expanded to the
  absolute path to the directory.
* If $DIR is not present, the directory will be added to the end of the
  invocation line of the script.
  * This is important for Windows.
    * For historical reasons, the entire string is broken up by
      shell-pattern expansion rules.
    * Since Windows paths frequently include spaces, and the Windows shell
      breaks on space, the quotes are needed for the script to understand
      the directory.
* If your script can be run directly on your platform, you can specify just
  the script.
  * Examples of this are:
    * .bat and .cmd files on Windows
    * scripts set executable on UNIX with a #! shebang line pointing to a
      valid interpreter.
* You can also specify an explicit path to an interpreter and the script.
    * Example:  /path/to/my/installation/of/python.exe path/to/my/script.py
* Splunk software ships with an example archiving script in that you SHOULD
  NOT USE $SPLUNK_HOME/bin called coldToFrozenExample.py
  * DO NOT USE the example for production use, because:
    * 1 - It will be overwritten on upgrade.
    * 2 - You should be implementing whatever requirements you need in a
          script of your creation. If you have no such requirements, use
          'coldToFrozenDir'
* Example configuration:
  * If you create a script in bin/ called our_archival_script.py, you could use:
    UNIX:
        coldToFrozenScript = "$SPLUNK_HOME/bin/python" \
          "$SPLUNK_HOME/bin/our_archival_script.py"
    Windows:
        coldToFrozenScript = "$SPLUNK_HOME/bin/python" \
          "$SPLUNK_HOME/bin/our_archival_script.py" "$DIR"
* The example script handles data created by different versions of Splunk
  differently. Specifically, data from before version 4.2 and after version 4.2
  are handled differently. See "Freezing and Thawing" below:
* The script must be in $SPLUNK_HOME/bin or a subdirectory thereof.
* No default.

python.version = {default|python|python2|python3}
```

```
* For Python scripts only, selects which Python version to use.
* This setting is valid for 'coldToFrozenScript' only when the value starts
  with the canonical path to the Python interpreter, in other words,
  $SPLUNK_HOME/bin/python. If you use any other path, this setting is ignored.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.

coldToFrozenDir = <path to frozen archive>
* An alternative to a 'coldToFrozen' script - this setting lets you
  specify a destination path for the frozen archive.
* Splunk software automatically puts frozen buckets in this directory
* For information on how buckets created by different versions are
  handled, see "Freezing and Thawing" below.
* If both 'coldToFrozenDir' and 'coldToFrozenScript' are specified,
  'coldToFrozenDir' takes precedence
* You must restart splunkd after changing this setting. Reloading the
  configuration does not suffice.
* May NOT contain a volume reference.

# Freezing and Thawing (this should move to web docs
4.2 and later data:
  * To archive: remove files except for the rawdata directory, since rawdata
    contains all the facts in the bucket.
    CAUTION: if the bucket has a stubbed-out (empty) rawdata file, then
    all of the bucket files, not just the rawdata directory must be archived
    to allow for data recovery. To determine whether the rawdata file is stubbed-out,
    check whether the setting "metric.stubOutRawdataJournal" is set to "true"
    for the index that the bucket belongs to. In addition, a stubbed-out rawdata
    file has a very small size (around 16KB) compared with the size of a normal
    rawdata file.
  * To restore: run splunk rebuild <bucket_dir> on the archived bucket, then
    atomically move the bucket to thawed for that index
4.1 and earlier data:
  * To archive: gzip the .tsidx files, as they are highly compressible but
    cannot be recreated
  * To restore: unpack the tsidx files within the bucket, then atomically
    move the bucket to thawed for that index

compressRawdata = true|false
* This setting is ignored. The splunkd process always compresses raw data.

maxConcurrentOptimizes = <nonnegative integer>
* The number of concurrent optimize processes that can run against a hot
  bucket.
* This number should be increased if:
  * There are always many small tsidx files in the hot bucket.
  * After rolling, there are many tsidx files in warm or cold buckets.
* You must restart splunkd after changing this setting. Reloading the
  configuration does not suffice.
* The highest legal value is 4294967295.
* Default: 6

maxDataSize = <positive integer>|auto|auto_high_volume
* The maximum size, in megabytes, that a hot bucket can reach before splunkd
  triggers a roll to warm.
* Specifying "auto" or "auto_high_volume" will cause Splunk to autotune this
  setting (recommended).
* You should use "auto_high_volume" for high-volume indexes (such as the
  main index); otherwise, use "auto". A "high volume index" would typically
  be considered one that gets over 10GB of data per day.
```

```
* "auto_high_volume" sets the size to 10GB on 64-bit, and 1GB on 32-bit
  systems.
* Although the maximum value you can set this is 1048576 MB, which
  corresponds to 1 TB, a reasonable number ranges anywhere from 100 to
  50000. Before proceeding with any higher value, please seek approval of
  Splunk Support.
* If you specify an invalid number or string, maxDataSize will be auto
  tuned.
* NOTE: The maximum size of your warm buckets might slightly exceed
  'maxDataSize', due to post-processing and timing issues with the rolling
  policy.
* For remote storage enabled indexes, consider setting this value to "auto"
  (750MB) or lower.
* Default: "auto" (sets the size to 750 megabytes)

rawFileSizeBytes = <positive integer>
* Deprecated in version 4.2 and later. Splunkd ignores this value.
* Rawdata chunks are no longer stored in individual files.
* If you really need to optimize the new rawdata chunks (highly unlikely),
  configure the 'rawChunkSizeBytes' setting.

rawChunkSizeBytes = <positive integer>
* Target uncompressed size, in bytes, for individual raw slices in the rawdata
  journal of the index.
* This is an advanced setting. Do not change it unless a Splunk Support
  professional asks you to.
* If you specify "0", 'rawChunkSizeBytes' is set to the default value.
* NOTE: 'rawChunkSizeBytes' only specifies a target chunk size. The actual
  chunk size may be slightly larger by an amount proportional to an
  individual event size.
* You must restart splunkd after changing this setting. Reloading the
  configuration does not suffice.
* The highest legal value is 18446744073709551615
* Default: 131072 (128 kilobytes)

minRawFileSyncSecs = <nonnegative decimal>|disable
* How frequently splunkd forces a filesystem sync while compressing journal
  slices. During this interval, uncompressed slices are left on disk even
  after they are compressed. Splunkd then forces a filesystem sync of the
  compressed journal and remove the accumulated uncompressed files.
* If you specify "0", splunkd forces a filesystem sync after every slice
  completes compressing.
* If you specify "disable", syncing is disabled entirely; uncompressed
  slices are removed as soon as compression is complete.
* Some filesystems are very inefficient at performing sync operations, so
  only enable this if you are sure you need it.
* You must restart splunkd after changing this setting. Reloading the
  configuration does not suffice.
* No exponent may follow the decimal.
* The highest legal value is 18446744073709551615.
* Default: "disable"

maxMemMB = <nonnegative integer>
* The amount of memory, in megabytes, to allocate for indexing.
* This amount of memory will be allocated PER INDEX THREAD, or, if
  indexThreads is set to 0, once per index.
* CAUTION: Calculate this number carefully. splunkd crashes if you set
  this number to higher than the amount of memory available.
* The default is recommended for all environments.
* The highest legal value is 4294967295.
* Default: 5
```

```
maxHotSpanSecs = <positive integer>
* Upper bound of timespan of hot/warm buckets, in seconds.
* This is an advanced setting that should be set
  with care and understanding of the characteristics of your data.
* Splunkd applies this limit per ingestion pipeline. For more
  information about multiple ingestion pipelines, see
  'parallelIngestionPipelines' in the server.conf.spec file.
* With N parallel ingestion pipelines, each ingestion pipeline writes to
  and manages its own set of hot buckets, without taking into account the state
  of hot buckets managed by other ingestion pipelines. Each ingestion pipeline
  independently applies this setting only to its own set of hot buckets.
* If you set 'maxHotBuckets' to 1, splunkd attempts to send all
  events to the single hot bucket and does not enforce 'maxHotSpanSeconds'.
* If you set this setting to less than 3600, it will be automatically
  reset to 3600.
* NOTE: If you set this setting to too small a value, splunkd can generate
  a very large number of hot and warm buckets within a short period of time.
* The highest legal value is 4294967295.
* NOTE: the bucket timespan snapping behavior is removed from this setting.
  See the 6.5 spec file for details of this behavior.
* Default: 7776000 (90 days)

maxHotIdleSecs = <nonnegative integer>
* How long, in seconds, that a hot bucket can remain in hot status without
  receiving any data.
* If a hot bucket receives no data for more than 'maxHotIdleSecs' seconds,
  splunkd rolls the bucket to warm.
* This setting operates independently of 'maxHotBuckets', which can also cause
  hot buckets to roll.
* A value of 0 turns off the idle check (equivalent to infinite idle time).
* The highest legal value is 4294967295
* Default: 0

maxHotBuckets = <positive integer> | auto
* Maximum number of hot buckets that can exist per index.
* When 'maxHotBuckets' is exceeded, the indexer rolls the hot bucket
  containing the least recent data to warm.
* Both normal hot buckets and quarantined hot buckets count towards this
  total.
* This setting operates independently of maxHotIdleSecs, which can also
  cause hot buckets to roll.
* NOTE: the indexer applies this limit per ingestion pipeline. For more
  information about multiple ingestion pipelines, see
  'parallelIngestionPipelines' in the server.conf.spec file.
* With N parallel ingestion pipelines, the maximum number of hot buckets across
  all of the ingestion pipelines is N * 'maxHotBuckets', but only
  'maxHotBuckets' for each ingestion pipeline. Each ingestion pipeline
  independently writes to and manages up to 'maxHotBuckets' number of hot
  buckets. Consequently, when multiple ingestion pipelines are configured, there
  may be multiple hot buckets with events on overlapping time ranges.
* The highest legal value is 1024. However, do not set to a value greater
  than 11 without direction from Splunk Support. Higher values can degrade
  indexing performance.
* If you specify "auto", the indexer sets the value to 3.
* This setting applies only to event indexes.
* Default: "auto"

metric.maxHotBuckets = <positive integer> | auto
* Maximum number of hot buckets that can exist per metric index
* When 'metric.maxHotBuckets' is exceeded, the indexer rolls the hot bucket
  containing the least recent data to warm.
* Both normal hot buckets and quarantined hot buckets count towards this
```

```
  total.
* This setting operates independently of maxHotIdleSecs, which can also
  cause hot buckets to roll.
* NOTE: the indexer applies this limit per ingestion pipeline. For more
  information about multiple ingestion pipelines, see
  'parallelIngestionPipelines' in the server.conf.spec file.
* With N parallel ingestion pipelines, the maximum number of hot buckets across
  all of the ingestion pipelines is N * 'metric.maxHotBuckets', but only
  'metric.maxHotBuckets' for each ingestion pipeline. Each ingestion pipeline
  independently writes to and manages up to 'metric.maxHotBuckets' number of hot
  buckets. Consequently, when multiple ingestion pipelines are configured, there
  may be multiple hot buckets with events on overlapping time ranges.
* The highest legal value is 4294967295
* When set to "auto", this setting will take the defined value from
  "maxHotBuckets". If "maxHotBuckets" is also set to "auto", the functional
  value for "metric.maxHotBuckets" is 6.
* This setting applies only to metric indexes.
* Default: "auto"

minHotIdleSecsBeforeForceRoll = <nonnegative integer>|auto
* When there are no existing hot buckets that can fit new events because of
  their timestamps and the constraints on the index (refer to 'maxHotBuckets',
  'maxHotSpanSecs' and 'quarantinePastSecs'), if any hot bucket has been idle
  (not receiving any data) for 'minHotIdleSecsBeforeForceRoll' seconds,
  a new bucket is created to receive these new events and the
  idle bucket is rolled to warm.
* If no hot bucket has been idle for 'minHotIdleSecsBeforeForceRoll' seconds,
  or if 'minHotIdleSecsBeforeForceRoll' has been set to 0, then a best fit
  bucket is chosen for these new events from the existing set of hot buckets.
* This setting operates independently of 'maxHotIdleSecs', which causes hot
  buckets to roll after they have been idle for 'maxHotIdleSecs' seconds,
  regardless of whether new events can fit into the existing hot buckets or not
  due to an event timestamp. 'minHotIdleSecsBeforeForceRoll', on the other hand,
  controls a hot bucket roll only under the circumstances when the timestamp
  of a new event cannot fit into the existing hot buckets given the other
  setting constraints on the system (such as 'maxHotBuckets',
  'maxHotSpanSecs', and 'quarantinePastSecs').
* If you specify "auto", splunkd autotunes this setting.
  The value begins at 600 and automatically adjusts upwards for
  optimal performance. Specifically, the value increases when a hot bucket rolls
  due to idle time with a significantly smaller size than 'maxDataSize'. As a
  consequence, the outcome might be fewer buckets, though these buckets might
  span wider earliest-latest time ranges of events.
* If you specify a value of "0", splunkd turns off the idle check
  (this is equivalent to infinite idle time).
* Setting this to zero means that splunkd never rolls a hot bucket as a
  consequence of an event not fitting into an existing hot bucket due to the
  constraints of other settings. Instead, it finds a best fitting
  bucket to accommodate that event.
* The highest legal value is 4294967295.
* NOTE: If you configure this setting, there is a chance that this could
  lead to frequent hot bucket rolls to warm, depending on the value. If your
  index contains a large number of buckets whose size on disk falls
  considerably short of the size specified in 'maxDataSize', and if the reason
  for the roll of these buckets is due to "caller=lru", then configuring the
  setting value to a larger value or to zero might reduce the frequency of hot
  bucket rolls (see the "auto" value above). You can check splunkd.log for a
  message like the following for rolls due to this setting:
    INFO  HotBucketRoller - finished moving hot to warm
      bid=_internal~0~97597E05-7156-43E5-85B1-B0751462D16B idx=_internal
      from=hot_v1_0 to=db_1462477093_1462477093_0 size=40960 caller=lru
      maxHotBuckets=3, count=4 hot buckets,evicting_count=1 LRU hots
```

```
* Default: auto

splitByIndexKeys = <comma separated list>
* By default, splunkd splits buckets by time ranges. When this happens, each
  bucket is defined by an earliest and latest time.
* Use this setting to optionally split buckets by one or more index key fields
  instead of time ranges.
* Valid key values are: host, sourcetype, source.
* This setting applies only to event indexes and requires that the minimal
  value of 'maxHotBuckets' is 2.
* If not set, splunkd splits buckets by time span.
* Default: empty string (no key)

metric.splitByIndexKeys = <comma separated list>
* By default, splunkd splits buckets by time ranges. When this happens, each
  bucket is defined by an earliest and latest time.
* Use this setting to optionally split buckets by one or more index key fields
  instead of time ranges.
* Valid key values are: host, sourcetype, source, metric_name.
* This setting applies only to metric indexes and requires that the minimal
  value of 'metric.maxHotBuckets' is 2.
* If not set, the setting 'splitByIndexKeys' applies. If 'splitByIndexKeys' is
  not set either, splunkd splits buckets by time span.
* Default: empty string (no key)

quarantinePastSecs = <positive integer>
* Determines what constitutes an anomalous past timestamp for quarantining
  purposes.
* If an event has a timestamp of 'quarantinePastSecs' older than the
  current time ("now"), the indexer puts that event in the quarantine bucket.
* This is a mechanism to prevent the main hot buckets from being polluted
  with fringe events.
* The highest legal value is 4294967295
* Default: 77760000 (900 days)

quarantineFutureSecs = <positive integer>
* Determines what constitutes an anomalous future timestamp for quarantining
  purposes.
* If an event has a timestamp of 'quarantineFutureSecs' newer than the
  current time ("now"), the indexer puts that event in the quarantine bucket.
* This is a mechanism to prevent the main hot buckets from being polluted with
  fringe events.
* The highest legal value is 4294967295
* Default: 2592000 (30 days)

maxMetaEntries = <nonnegative integer>
* The maximum number of unique lines in .data files in a bucket, which
  might help to reduce memory consumption
* If this value is exceeded, a hot bucket is rolled to prevent further increase
* If your buckets are rolling due to Strings.data reaching this limit, the
  culprit might be the 'punct' field in your data. If you do not use 'punct',
  it might be best to simply disable this (see props.conf.spec)
  * NOTE: since at least 5.0.x, large strings.data from usage of punct are rare.
* There is a delta between when 'maxMetaEntries' is exceeded and splunkd rolls
  the bucket.
* This means a bucket might end up with more lines than specified in
  'maxMetaEntries', but this is not a major concern unless that excess
  is significant.
* If set to 0, splunkd ignores this setting (it is treated as infinite)
* Highest legal value is 4294967295.
* Default: 1000000
```

```
syncMeta = <boolean>
* Whether or not splunkd calls a sync operation before the file descriptor
  is closed on metadata file updates.
* When set to "true", splunkd calls a sync operation before it closes the
  file descriptor on metadata file updates.
* This functionality was introduced to improve integrity of metadata files,
  especially in regards to operating system crashes/machine failures.
* NOTE: Do not change this setting without the input of a Splunk support
  professional.
* You must restart splunkd after changing this setting. Reloading the
  configuration does not suffice.
* Default: true

serviceMetaPeriod = <positive integer>
* Defines how frequently, in seconds, that metadata is synced to disk.
* You might want to set this to a higher value if the sum of your metadata
  file sizes is larger than many tens of megabytes, to avoid the negative effect on I/O
  in the indexing fast path.
* The highest legal value is 4294967295
* Default: 25

partialServiceMetaPeriod = <positive integer>
* The amount of time, in seconds, that splunkd syncs metadata for records that
  can be synced efficiently in place without requiring a full rewrite of the
  metadata file.
* Related to 'serviceMetaPeriod'. Records that require a full rewrite of the
  metadata file are synced every 'serviceMetaPeriod' seconds.
* If you set this to 0, the feature is turned off, and 'serviceMetaPeriod'
  is the only time when metadata sync happens.
* If the value of 'partialServiceMetaPeriod' is greater than
  the value of 'serviceMetaPeriod', this setting has no effect.
* Splunkd ignores this setting if 'serviceOnlyAsNeeded' = "true" (the default).
* The highest legal value is 4294967295.
* Default: 0 (disabled)

throttleCheckPeriod = <positive integer>
* How frequently, in seconds, that splunkd checks for index throttling
  conditions.
* NOTE: Do not change this setting unless a Splunk Support
  professional asks you to.
* The highest legal value is 4294967295.
* Default: 15

maxTimeUnreplicatedWithAcks = <nonnegative decimal>
* How long, in seconds, that events can remain in an unacknowledged state
  within a raw slice.
* This value is important if you have enabled indexer acknowledgment on
  forwarders by configuring the 'useACK' setting in outputs.conf and
  have enabled replication through indexer clustering.
* This is an advanced setting. Confirm that you understand the settings
  on all your forwarders before changing it.
    * Do not exceed the ack timeout configured on any forwarders.
    * Set to a number that is at most half of the minimum value of that timeout.
  Review the 'readTimeout' setting in the [tcpout] stanza in outputs.conf.spec
  for information about the ack timeout.
* Configuring this setting to 0 disables the check. Do not do this.
* The highest legal value is 2147483647.
* Default: 60

maxTimeUnreplicatedNoAcks = <nonnegative decimal>
* How long, in seconds, that events can remain in a raw slice.
* This setting is important only if replication is enabled for this index,
```

```
  otherwise it is ignored.
* If there are any acknowledged events that share this raw slice, this setting
  does not apply. Instead, splunkd uses the value in the
  'maxTimeUnreplicatedWithAcks' setting.)
* The highest legal value is 2147483647.
* Configuring this setting to 0 disables the check.
* Do not configure this setting on remote storage enabled indexes.
* NOTE: Take care and understand the consequences before changing this setting.
* Default: 300

isReadOnly = <boolean>
* Whether or not the index is read-only.
* If you set to "true", no new events can be added to the index, but the
  index is still searchable.
* You must restart splunkd after changing this setting. Reloading the
  index configuration does not suffice.
* Do not configure this setting on remote storage enabled indexes.
* If set to 'true', replication must be turned off (repFactor=0) for the index.
* Default: false

homePath.maxDataSizeMB = <nonnegative integer>
* Specifies the maximum size of 'homePath' (which contains hot and warm
  buckets).
* If this size is exceeded, splunkd moves buckets with the oldest value
  of latest time (for a given bucket) into the cold DB until homePath is
  below the maximum size.
* If you set this setting to 0, or do not set it, splunkd does not constrain the
  size of 'homePath'.
* The highest legal value is 4294967295.
* Splunkd ignores this setting for remote storage enabled indexes.
* Default: 0

coldPath.maxDataSizeMB = <nonnegative integer>
* Specifies the maximum size of 'coldPath' (which contains cold buckets).
* If this size is exceeded, splunkd freezes buckets with the oldest value
  of latest time (for a given bucket) until coldPath is below the maximum
  size.
* If you set this setting to 0, or do not set it, splunkd does not constrain the
  size of 'coldPath'.
* If splunkd freezes buckets due to enforcement of this setting, and
  'coldToFrozenScript' and/or 'coldToFrozenDir' archiving settings are also
  set on the index, these settings are used.
* Splunkd ignores this setting for remote storage enabled indexes.
* The highest legal value is 4294967295.
* Default: 0

disableGlobalMetadata = <boolean>
* NOTE: This option was introduced in version 4.3.3, but as of 5.0 it
  is obsolete and splunkd ignores it if you set it.
* It used to disable writing to the global metadata. In 5.0,  global metadata
  was removed.

repFactor = 0|auto
* Valid only for indexer cluster peer nodes.
* Determines whether an index gets replicated.
* Configuring this setting to 0 turns off replication for this index.
* Configuring to "auto" turns on replication for this index.
* You must configure this setting to the same value on all peer nodes.
* Default: 0

minStreamGroupQueueSize = <nonnegative integer>
* Minimum size of the queue that stores events in memory before committing
```

```
  them to a tsidx file.
* As splunkd operates, it continually adjusts this size internally. Splunkd
  could decide to use a small queue size and thus generate tiny tsidx files
  under certain unusual circumstances, such as file system errors.  The
  danger of a very low minimum is that it can generate very tiny tsidx files
  with one or very few events, making it impossible for splunk-optimize to
  catch up and optimize the tsidx files into reasonably sized files.
* Do not configure this setting unless a Splunk Support professional
  asks you to.
* The highest legal value is 4294967295.
* Default: 2000

streamingTargetTsidxSyncPeriodMsec = <nonnegative integer>
* The amount of time, in milliseconds, that splunkd forces a sync
  of tsidx files on streaming targets.
* This setting is needed for multisite clustering where streaming targets
  might be primary.
* If you configure this setting to 0, syncing of tsidx files on
  streaming targets does not occur.
* No default.

journalCompression = gzip|lz4|zstd
* The compression algorithm that splunkd should use for the rawdata journal
  file of new index buckets.
* This setting does not have any effect on already created buckets. There is
  no problem searching buckets that are compressed with different algorithms.
* Default: zstd

enableTsidxReduction = <boolean>
* When set to true, this setting enables tsidx file reduction for event indexes.
* Under tsidx file reduction, the indexer reduces the tsidx files of buckets
  when the buckets reach the age specified by
  'timePeriodInSecBeforeTsidxReduction'.
* CAUTION: Do not set this setting to "true" for event indexes that are
  configured to use remote storage with the "remotePath" setting.
* NOTE: This setting applies to buckets in warm, cold, and thawed.
  It does not apply to metrics index buckets
* Default: false

metric.enableFloatingPointCompression = <boolean>
* Determines whether the floating-point values compression is enabled for metric
  index files.
* Set this to false only if you are experiencing high CPU usage during data
  ingestion. However, doing this will cause you to lose the disk space savings
  that the compression gives you.
* Default: true

metric.compressionBlockSize = <integer>
* The block size, in words (eight-byte multiples), that the floating-point compression
  algorithm should use to store compressed data within a single block in a column.
* Valid only if 'metric.enableMetricTsidxFloatingPointCompression' is set to "true".
* Minimum value: 128 (1024 bytes)
* Default: 1024 (8192 bytes)

metric.stubOutRawdataJournal = <boolean>
* For metrics indexes only.
* Determines whether the data in the rawdata file is deleted when the hot bucket
  rolls to warm. The rawdata file itself remains in place in the bucket.
* Tsidx files are not affected by this setting.
* This setting does not take effect for indexes that have replication enabled ("repFactor=auto")
  in an indexer cluster deployment.
* A change to this setting affects only future buckets or buckets that are currently hot
```

```
  when the change occurs. It does not affect buckets already in the warm or cold state.
* Searches over metrics indexes do not use the rawdata file. Therefore, changing this
  setting to "true" does not affect search results.
* The benefits of setting to true are:
    * Reduces storage requirements, by reducing rawdata files to the minimal size.
    * Potentially improves search time, because the maximum bucket size (controlled by "maxDataSizeMB")
      now allows for larger tsidx files, since the rawdata file no longer occupies significant space.
      The rawdata file size is discounted from the overall bucket size while writing continues in a hot
bucket,
      even though the rawdata file is not removed until the bucket rolls to warm. Thus, the hot bucket might
      exceed "maxDataSizeMB", but, once the bucket rolls to warm, its size will no longer exceed
"maxDataSizeMB".
* Caution: Because setting this attribute to "true" eliminates the data in the rawdata files, those
  files can no longer be used in bucket repair operations.
* Default: true

suspendHotRollByDeleteQuery = <boolean>
* Whether or not splunkd rolls hot buckets upon running of the "delete"
  search command, or waits to roll them for other reasons.
* When the "delete" search command is run, all buckets that contain data
  to be deleted are marked for updating of their metadata files. The indexer
  normally first rolls any hot buckets as rolling must precede the metadata
  file updates.
* When 'suspendHotRollByDeleteQuery' is set to "true", the rolling of hot
  buckets for the "delete" command is suspended. The hot buckets, although
  marked, do not roll immediately, but instead wait to roll in response to
  the same circumstances operative for any other hot buckets; for example,
  due to reaching a limit set by 'maxHotBuckets', 'maxDataSize', etc. When
  these hot buckets finally roll, their metadata files are then updated.
* Default: false

tsidxReductionCheckPeriodInSec = <positive integer>
* The amount of time, in seconds, between service runs to reduce the tsidx
  files for any buckets that have reached the age specified by
  'timePeriodInSecBeforeTsidxReduction'.
* Default: 600

timePeriodInSecBeforeTsidxReduction = <positive integer>
* The amount of time, in seconds, that a bucket can age before it
  becomes eligible for tsidx reduction.
* The bucket age is the difference between the current time
  and the timestamp of the bucket's latest event.
* When this time difference is exceeded, a bucket becomes eligible
  for tsidx reduction.
* Default: 604800

tsidxDedupPostingsListMaxTermsLimit = <positive integer>
* This setting is valid only when 'tsidxWritingLevel' is at 4 or higher.
* This max term limit sets an upper bound on the number of terms kept inside an
  in-memory hash table that serves to improve tsidx compression.
* The tsidx optimizer uses the hash table to identify terms with identical
  postings lists. When the first instance of a term is received its postings
  list is stored. When successive terms with identical postings lists are
  received the tsidx optimizer makes them refer to the first instance of the
  postings list rather than creating and storing term postings list duplicates.
* Consider increasing this limit to improve compression for large tsidx files.
  For example, a tsidx file created with 'tsidxTargetSizeMB' over 1500MB can
  contain a large number of terms with identical postings lists.
* Reducing this limit helps conserve memory consumed by optimization processes,
  at the cost of reduced tsidx compression.
* Set this limit to 0 to disable deduplicated postings list compression.
* This setting cannot exceed 1,073,741,824 (2^30).
```

```
* Default: 8,388,608 (2^23)

tsidxTargetSizeMB = <positive integer>
* The target size for tsidx files. The indexer attempts to make all tsidx files
  in index buckets as close to this size as possible when:
      (a) buckets merge.
      (b) hot buckets roll to warm buckets.
* This value is used to help tune the performance of tsidx-based search queries,
  especially 'tstats'.
* If this value exceeds 'maxDataSize', then the hot bucket will roll based
  on the 'maxDataSize' configuration even if no tsidx file has met the
  specified 'tsidxTargetSizeMB'.
* Cannot exceed 4096 MB (4 GB).
* Default: 1500 (MB)

metric.tsidxTargetSizeMB = <positive integer>
* The target size for msidx files (tsidx files for metrics data). The indexer
  attempts to make all msidx files in index buckets as close to this size as
  possible when:
      (a) buckets merge.
      (b) hot buckets roll to warm buckets.
* This value is used to help tune the performance of metrics search queries,
  especially 'mstats'.
* If this value exceeds 'maxDataSize', then the hot bucket will roll based
  on the 'maxDataSize' configuration even if no msidx file has met the
  specified 'metric.tsidxTargetSizeMB'.
* Cannot exceed 4096 MB (4 GB).
* Default: 1500 (MB)

metric.timestampResolution = <s|ms>
* This setting specifies the timestamp resolution for metrics tsidx files.
  Specify 's' for timestamps with second resolution. Specify 'ms' for
  timestamps with millisecond resolution.
* Indexes with millisecond timestamp precision have reduced search performance.
* Optional.
* Default: s

datatype = <event|metric>
* Determines whether the index stores log events or metric data.
* If set to "metric", the indexer optimizes the index to store metric
  data which can be  queried later only using the 'mstats' operator,
  as searching metric data is different from traditional log events.
* Use the "metric" data type only for metric sourcetypes like statsd.
* Optional.
* Default: event

waitPeriodInSecsForManifestWrite = <nonnegative integer>
* This setting specifies the minimum interval, in seconds, between periodic
  updates of an index's manifest file.
* Setting to a lower value can reduce the performance of bucket operations like
  fix-ups, freezes, etc.
* Do not increase this value beyond the default except through consultation with
  Splunk Support. Increasing the value can lead to inconsistencies in data.
* The highest legal value is 4294967295.
* Default: 60 (1 min)

hotBucketStreaming.sendSlices = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Enables uploading of journal slices of hot buckets to the remote storage.
* Default: false
```

```
hotBucketStreaming.removeRemoteSlicesOnRoll = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Enables removal of uploaded journal slices of hot buckets from the remote
  storage after a bucket rolls from hot to warm.
* This setting should be enabled only if 'hotBucketStreaming.sendSlices' is
  also enabled.
* Default: false

hotBucketStreaming.removeRemoteSlicesOnFreeze = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Enables removal of uploaded journal slices of hot buckets from the remote
  storage after a bucket rolls from warm to frozen.
* This setting should be enabled only if 'hotBucketStreaming.sendSlices' is
  also enabled.
* Default: false

hotBucketStreaming.reportStatus = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Default: false

hotBucketStreaming.deleteHotsAfterRestart = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Default: false
```

PER PROVIDER FAMILY OPTIONS

```
# A provider family is a way of collecting properties that are common to
# multiple providers. There are no properties that can only be used in a
# provider family, and not in a provider. If the same property is specified
# in a family, and in a provider belonging to that family, then the latter
# value "wins".
#
# All family stanzas begin with "provider-family:". For example:
# [provider-family:family_name]
# vix.mode=stream
# vix.command = java
# vix.command.arg.1 = -Xmx512m
# ....
#*******************************************************************
```

PER PROVIDER OPTIONS

```
# These options affect External Resource Providers (ERPs). All provider stanzas
# begin with "provider:". For example:
#   [provider:provider_name]
#   vix.family                  = hadoop
#   vix.env.JAVA_HOME           = /path/to/java/home
#   vix.env.HADOOP_HOME         = /path/to/hadoop/client/libraries
#
# Each virtual index must reference a provider.
```

```
#********************************************************************
vix.family = <family>
* A provider family to which this provider belongs.
* The only family available by default is "hadoop". Others can be added.

vix.mode = stream|report
* Usually specified at the family level.
* Typically should be "stream".
* In general, do not use "report" without consulting Splunk Support.

vix.command = <command>
* The command to be used to launch an external process for searches on this
  provider.
* Usually specified at the family level.

vix.command.arg.<N> = <argument>
* The Nth argument to the command specified by vix.command.
* Usually specified at the family level, but frequently overridden at the
  provider level, for example to change the jars used depending on the
  version of Hadoop to which a provider connects.

vix.<property name> = <property value>
* All such properties will be made available as "configuration properties" to
  search processes on this provider.
* For example, if this provider is in the Hadoop family, the configuration
  property "mapreduce.foo = bar" can be made available to the Hadoop
  via the property "vix.mapreduce.foo = bar".

vix.env.<env var name> = <env var variable>
* Will create an environment variable available to search processes on this
  provider.
* For example, to set the JAVA_HOME variable to "/path/java" for search
  processes on this provider, use "vix.env.JAVA_HOME = /path/java".

#********************************************************************
# PER PROVIDER OPTIONS -- HADOOP
# These options are specific to ERPs with the Hadoop family.
# NOTE: Many of these properties specify behavior if the property is not
#       set. However, default values set in system/default/indexes.conf
#       take precedence over the "unset" behavior.
#********************************************************************

vix.javaprops.<JVM system property name> = <value>
* All such properties will be used as Java system properties.
* For example, to specify a Kerberos realm (say "foo.com") as a Java
  system property, use the property
  "vix.javaprops.java.security.krb5.realm = foo.com".

vix.mapred.job.tracker = <logical name or server:port>
* In high-availability mode, use the logical name of the Job Tracker.
* Otherwise, should be set to server:port for the single Job Tracker.
* Note: this property is passed straight to Hadoop. Not all such properties
  are documented here.

vix.fs.default.name = <logical name or hdfs://server:port>
* In high-availability mode, use the logical name for a list of Name Nodes.
* Otherwise, use the URL for the single Name Node.
* Note: this property is passed straight to Hadoop. Not all such properties
  are documented here.

vix.splunk.setup.onsearch = true|false
* Whether to perform setup (install & bundle replication) on search.
```

```
* Default: false

vix.splunk.setup.package = current|<path to file>
* Splunk .tgz package to install and use on data nodes
  (in vix.splunk.home.datanode).
* Uses the current install if set to value 'current' (without quotes).

vix.splunk.home.datanode = <path to dir>
* Path to where splunk should be installed on datanodes/tasktrackers, i.e.
  SPLUNK_HOME.
* Required.

vix.splunk.home.hdfs = <path to dir>
* Scratch space for this Splunk instance on HDFS
* Required.

vix.splunk.search.debug = true|false
* Whether to run searches against this index in debug mode. In debug mode,
  additional information is logged to search.log.
* Optional.
* Default: false

vix.splunk.search.recordreader = <list of classes>
* Comma separated list of data preprocessing classes.
* Each such class must extend BaseSplunkRecordReader and return data to be
  consumed by Splunk as the value.

vix.splunk.search.splitter = <class name>
* Set to override the class used to generate splits for MR jobs.
* Classes must implement com.splunk.mr.input.SplitGenerator.
* Unqualified classes will be assumed to be in the package com.splunk.mr.input.
* May be specified in either the provider stanza, or the virtual index stanza.
* To search Parquet files, use ParquetSplitGenerator.
* To search Hive files, use HiveSplitGenerator.

vix.splunk.search.mr.threads = <postive integer>
* Number of threads to use when reading map results from HDFS
* Numbers less than 1 will be treated as 1.
* Numbers greater than 50 will be treated as 50.
* Default: 10

vix.splunk.search.mr.maxsplits = <positive integer>
* Maximum number of splits in an MR job.
* Default: 10000

vix.splunk.search.mr.minsplits = <positive integer>
* Number of splits for first MR job associated with a given search.
* Default: 100

vix.splunk.search.mr.splits.multiplier = <decimal greater than or equal to 1.0>
* Factor by which the number of splits is increased in consecutive MR jobs for
  a given search, up to the value of maxsplits.
* Default: 10

vix.splunk.search.mr.poll = <positive integer>
* Polling period for job status, in milliseconds.
* Default: 1000 (1 second).

vix.splunk.search.mr.mapper.output.replication = <positive integer>
* Replication level for mapper output.
* Default: 3
```

```
vix.splunk.search.mr.mapper.output.gzlevel = <integer between 0 and 9, inclusive>
* The compression level used for the mapper output.
* Default: 2

vix.splunk.search.mixedmode = <boolean>
* Whether mixed mode execution is enabled.
* Default: true

vix.splunk.search.mixedmode.maxstream = <nonnegative integer>
* Maximum number of bytes to stream during mixed mode.
* Value = 0 means there's no stream limit.
* Will stop streaming after the first split that took the value over the limit.
* Default: 10737418240 (10 GB).

vix.splunk.jars = <list of paths>
* Comma delimited list of Splunk dirs/jars to add to the classpath in the
  Search Head and MR.

vix.env.HUNK_THIRDPARTY_JARS = <list of paths>
* Comma delimited list of 3rd-party dirs/jars to add to the classpath in the
  Search Head and MR.

vix.splunk.impersonation = true|false
* Enable/disable user impersonation.

vix.splunk.setup.bundle.replication = <positive integer>
* Set custom replication factor for bundles on HDFS.
* Must be an integer between 1 and 32767.
* Increasing this setting may help performance on large clusters by decreasing
  the average access time for a bundle across Task Nodes.
* Optional.
* Default: The default replication factor for the file-system applies.

vix.splunk.setup.bundle.max.inactive.wait = <positive integer>
* A positive integer represent a time interval in seconds.
* While a task waits for a bundle being replicated to the same node by another
  task, if the bundle file is not modified for this amount of time, the task
  will begin its own replication attempt.
* Default: 5

vix.splunk.setup.bundle.poll.interval = <positive integer>
* A positive number, representing a time interval in milliseconds.
* While a task waits for a bundle to be installed by another task on the same
  node, it will check once per interval whether that installation is complete.
* Default: 100

vix.splunk.setup.bundle.setup.timelimit = <positive integer>
* A positive number, representing a time duration in milliseconds.
* A task will wait this long for a bundle to be installed before it quits.
* Default: 20000 (20 seconds).

vix.splunk.setup.package.replication = true|false
* Set custom replication factor for the Splunk package on HDFS. This is the
  package set in the property vix.splunk.setup.package.
* Must be an integer between 1 and 32767.
* Increasing this setting may help performance on large clusters by decreasing
  the average access time for the package across Task Nodes.
* Optional. If not set, the default replication factor for the file-system
  will apply.

vix.splunk.setup.package.max.inactive.wait = <positive integer>
* A positive integer represent a time interval in seconds.
```

```
* While a task waits for a Splunk package being replicated to the same node by
  another task, if the package file is not modified for this amount of time,
  the task will begin its own replication attempt.
* Default: 5

vix.splunk.setup.package.poll.interval = <positive integer>
* A positive number, representing a time interval in milliseconds.
* While a task waits for a Splunk package to be installed by another task on
  the same node, it will check once per interval whether that installation is
  complete.
* Default: 100

vix.splunk.setup.package.setup.timelimit = <positive integer>
* A positive number, representing a time duration in milliseconds.
* A task will wait this long for a Splunk package to be installed before it quits.
* Default: 20000 (20 seconds)

vix.splunk.setup.bundle.reap.timelimit = <positive integer>
* Specific to Hunk provider
* For bundles in the working directory on each data node, this property controls
  how old they must be before they are eligible for reaping.
* Unit is milliseconds
* Values larger than 86400000 will be treated as if set to 86400000.
* Default: 86400000 (24 hours)

vix.splunk.search.column.filter = <boolean>
* Enables/disables column filtering. When enabled, Hunk will trim columns that
  are not necessary to a query on the Task Node, before returning the results
  to the search process.
* Should normally increase performance, but does have its own small overhead.
* Works with these formats: CSV, Avro, Parquet, Hive.
* Default: true

#
# Kerberos properties
#

vix.kerberos.principal = <kerberos principal name>
* Specifies principal for Kerberos authentication.
* Should be used with vix.kerberos.keytab and either
  1) vix.javaprops.java.security.krb5.realm and
     vix.javaprops.java.security.krb5.kdc, or
  2) security.krb5.conf

vix.kerberos.keytab = <kerberos keytab path>
* Specifies path to keytab for Kerberos authentication.
* See usage note with vix.kerberos.principal.


#
# The following properties affect the SplunkMR heartbeat mechanism. If this
# mechanism is turned on, the SplunkMR instance on the Search Head updates a
# heartbeat file on HDFS. Any MR job spawned by report or mix-mode searches
# checks the heartbeat file. If it is not updated for a certain time, it will
# consider SplunkMR to be dead and kill itself.
#

vix.splunk.heartbeat = <boolean>
* Turn on/off heartbeat update on search head, and checking on MR side.
* Default: true

vix.splunk.heartbeat.path = <path on HDFS>
```

```
* Path to heartbeat file.
* If not set, defaults to <vix.splunk.home.hdfs>/dispatch/<sid>/

vix.splunk.heartbeat.interval = <positive integer>
* The frequency, in milliseconds, with which the Heartbeat will be updated
  on the Search Head.
* Minimum value is 1000. Smaller values will cause an exception to be thrown.
* Default: 6000 (6 seconds)

vix.splunk.heartbeat.threshold = <positive integer>
* The number of times the MR job will detect a missing heartbeat update before
  it considers SplunkMR dead and kills itself.
* Default: 10

## The following sections are specific to data input types.

#
# Sequence file
#

vix.splunk.search.recordreader.sequence.ignore.key = <boolean>
* When reading sequence files, if this key is enabled, events will be expected
  to only include a value. Otherwise, the expected representation is
  key+"\t"+value.
* Default: true

#
# Avro
#

vix.splunk.search.recordreader.avro.regex = <string>
* The regular expression that files must match in order to be considered avro files.
* Optional.
* Default: \.avro$

#
# Parquet
#

vix.splunk.search.splitter.parquet.simplifyresult = <boolean>
* If enabled, field names for map and list type fields will be simplified by
  dropping intermediate "map" or "element" subfield names. Otherwise, a field
  name will match parquet schema completely.
* May be specified in either the provider stanza or in the virtual index stanza.
* Default: true

#
# Hive
#

vix.splunk.search.splitter.hive.ppd = <boolean>
* Enable or disable Hive ORC Predicate Push Down.
* If enabled, ORC PPD will be applied whenever possible to prune unnecessary
  data as early as possible to optimize the search.
* May be specified in either the provider stanza or in the virtual index stanza.
* Default: true

vix.splunk.search.splitter.hive.fileformat = textfile|sequencefile|rcfile|orc
* Format of the Hive data files in this provider.
* May be specified in either the provider stanza or in the virtual index stanza.
* Default: "textfile"
```

```
vix.splunk.search.splitter.hive.dbname = <DB name>
* Name of Hive database to be accessed by this provider.
* Optional.
* May be specified in either the provider stanza or in the virtual index stanza.
* Default: "default"

vix.splunk.search.splitter.hive.tablename = <table name>
* Table accessed by this provider.
* Required property.
* May be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.columnnames = <list of column names>
* Comma-separated list of file names.
* Required if using Hive, not using metastore.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.columntypes = string:float:int # COLON separated list of column types,
required
* Colon-separated list of column- types.
* Required if using Hive, not using metastore.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.serde = <SerDe class>
* Fully-qualified class name of SerDe.
* Required if using Hive, not using metastore, and if specified in creation of Hive table.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.serde.properties = <list of key-value pairs>
* Comma-separated list of "key=value" pairs.
* Required if using Hive, not using metastore, and if specified in creation of Hive table.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.fileformat.inputformat = <InputFormat class>
* Fully-qualified class name of an InputFormat to be used with Hive table data.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.rowformat.fields.terminated = <delimiter>
* Will be set as the Hive SerDe property "field.delim".
* Optional.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.rowformat.escaped = <escape char>
* Will be set as the Hive SerDe property "escape.delim".
* Optional.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.rowformat.lines.terminated = <delimiter>
* Will be set as the Hive SerDe property "line.delim".
* Optional.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.rowformat.mapkeys.terminated  = <delimiter>
* Will be set as the Hive SerDe property "mapkey.delim".
* Optional.
* Can be specified in either the provider stanza or in the virtual index stanza.

vix.splunk.search.splitter.hive.rowformat.collectionitems.terminated = <delimiter>
* Will be set as the Hive SerDe property "colelction.delim".
* Optional.
* Can be specified in either the provider stanza or in the virtual index stanza.

#
```

```
# Archiving
#

vix.output.buckets.max.network.bandwidth = 0|<bits per second>
* Throttles network bandwidth to <bits per second>
* Set at provider level. Applied to all virtual indexes using a provider
  with this setting.
* Default: 0 (no throttling)
```

PER VIRTUAL INDEX OPTIONS

```
# These options affect virtual indexes. Like indexes, these options may
# be set under an [<virtual-index>] entry.
#
# Virtual index names have the same constraints as normal index names.
#
# Each virtual index must reference a provider. I.e:
# [virtual_index_name]
# vix.provider = <provider_name>
#
# All configuration keys starting with "vix." will be passed to the
# external resource provider (ERP).
#********************************************************************

vix.provider = <provider_name>
* Name of the external resource provider to use for this virtual index.

#********************************************************************
# PER VIRTUAL INDEX OPTIONS -- HADOOP
# These options are specific to ERPs with the Hadoop family.
#********************************************************************

#
# The vix.input.* configurations are grouped by an id.
# Inputs configured via the UI always use '1' as the id.
# In this spec we'll use 'x' as the id.
#

vix.input.x.path = <path>
* Path in a Hadoop filesystem (usually HDFS or S3).
* May contain wildcards.
* Checks the path for data recursively when ending with '...'
* Can extract fields with ${field}. I.e: "/data/${server}/...", where server
  will be extracted.
* May start with a schema.
  * The schema of the path specifies which hadoop filesystem implementation to
    use. Examples:
    * hdfs://foo:1234/path, will use a HDFS filesystem implementation
    * s3a://s3-bucket/path, will use a S3 filesystem implementation

vix.input.x.accept = <regex>
* Specifies an allow list regex.
* Only files within the location given by matching vix.input.x.path, whose
  paths match this regex, will be searched.

vix.input.x.ignore = <regex>
* Specifies a deny list regex.
* Searches will ignore paths matching this regex.
* These matches take precedence over vix.input.x.accept matches.
```

```
vix.input.x.required.fields = <comma separated list of fields>
* Fields that will be kept in search results even if the field is not
  required by the search

# Earliest time extractions - For all 'et' settings, there's an equivalent 'lt' setting.
vix.input.x.et.regex = <regex>
* Regex extracting earliest time from vix.input.x.path

vix.input.x.et.format = <java.text.SimpleDateFormat date pattern>
* Format of the extracted earliest time.
* See documentation for java.text.SimpleDateFormat

vix.input.x.et.offset = <seconds>
* Offset in seconds to add to the extracted earliest time.

vix.input.x.et.timezone = <java.util.SimpleTimeZone timezone id>
* Timezone in which to interpret the extracted earliest time.
* Examples: "America/Los_Angeles" or "GMT-8:00"

vix.input.x.et.value = mtime|<epoch time in milliseconds>
* Sets the earliest time for this virtual index.
* Can be used instead of extracting times from the path via vix.input.x.et.regex
* When set to "mtime", uses the file modification time as the earliest time.

# Latest time extractions - See "Earliest time extractions"

vix.input.x.lt.regex = <regex>
* Latest time equivalent of vix.input.x.et.regex

vix.input.x.lt.format = <java.text.SimpleDateFormat date pattern>
* Latest time equivalent of vix.input.x.et.format

vix.input.x.lt.offset = <seconds>
* Latest time equivalent of vix.input.x.et.offset

vix.input.x.lt.timezone = <java.util.SimpleTimeZone timezone id>
* Latest time equivalent of vix.input.x.et.timezone

vix.input.x.lt.value = <mod time>
* Latest time equivalent of vix.input.x.et.value

#
# Archiving
#

vix.output.buckets.path = <hadoop path>
* Path to a hadoop filesystem where buckets will be archived

vix.output.buckets.older.than = <integer>
* The age of a bucket, in seconds, before it is archived.
* The age of a bucket is determined by the the earliest _time field of
  any event in the bucket.

vix.output.buckets.from.indexes = <comma separated list of splunk indexes>
* List of (non-virtual) indexes that will get archived to this (virtual) index.

vix.unified.search.cutoff_sec = <seconds>
* Window length before present time that configures where events are retrieved
  for unified search
* Events from now to now-cutoff_sec will be retrieved from the splunk index
  and events older than cutoff_sec will be retrieved from the archive index
```

```
#*******************************************************************************
# PER VIRTUAL INDEX OR PROVIDER OPTIONS -- HADOOP
# These options can be set at either the virtual index level or provider
# level, for the Hadoop ERP.
#
# Options set at the virtual index level take precedence over options set
# at the provider level.
#
# Virtual index level prefix:
# vix.input.<input_id>.<option_suffix>
#
# Provider level prefix:
# vix.splunk.search.<option_suffix>
#*******************************************************************************

# The following options are just defined by their <option_suffix>

#
# Record reader options
#

recordreader.<name>.<conf_key> = <conf_value>
* Sets a configuration key for a RecordReader with <name> to <conf_value>

recordreader.<name>.regex = <regex>
* Regex specifying which files this RecordReader can be used for.

recordreader.journal.buffer.size = <bytes>
* Buffer size used by the journal record reader

recordreader.csv.dialect = default|excel|excel-tab|tsv
* Set the csv dialect for csv files
* A csv dialect differs on delimiter_char, quote_char and escape_char.
* Here is a list of how the different dialects are defined in order delimiter,
  quote, and escape:
  * default   = ,  " \
  * excel     = ,  " "
  * excel-tab = \t " "
  * tsv       = \t " \

#
# Splitter options
#

splitter.<name>.<conf_key> = <conf_value>
* Sets a configuration key for a split generator with <name> to <conf_value>
* See comment above under "PER VIRTUAL INDEX OR PROVIDER OPTIONS". This means
  that the full format is:
   vix.input.N.splitter.<name>.<conf_key> (in a vix stanza)
   vix.splunk.search.splitter.<name>.<conf_key> (in a provider stanza)

splitter.file.split.minsize = <integer>
* Minimum size, in bytes, for file splits.
* Default: 1

splitter.file.split.maxsize = <integer>
* Maximum size, in bytes, for file splits.
* Default: Long.MAX_VALUE

#*******************************************************************************
# Dynamic Data Self Storage settings.
```

```
# This section describes settings that affect the archiver-
# optional and archiver-mandatory settings only.
#
# As the first step in the Dynamic Data Self Storage feature, it allows users
# to move their data from Splunk indexes to customer-owned external storage
# in AWS S3 when the data reaches the end of the retention period. Note that
# only the raw data and delete marker files are transferred to the external
# storage.
#
# Future development may include the support for storage hierarchies and the
# automation of data rehydration.
#
# For example, use the following settings to configure Dynamic Data Self Storage.
#   archiver.selfStorageProvider     = S3
#   archiver.selfStorageBucket       = mybucket
#   archiver.selfStorageBucketFolder = folderXYZ
#*******************************************************************
archiver.selfStorageProvider = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the storage provider for Self Storage.
* Optional. Only required when using Self Storage.
* Self Storage only supports the Simple Storage Service (S3) and Google Cloud Storage (GCS)
  for Amazon Web Services (AWS) and Google Cloud Platform (GCP), respectively.
* NOTE: This setting value is case-sensitive.

archiver.selfStorageBucket = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the destination bucket for Self Storage.
* Optional. Only required when using Self Storage.

archiver.selfStorageBucketFolder = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the folder on the destination bucket for Self Storage.
* Optional.
* If not specified, data is uploaded to the root path in the destination bucket.

archiver.selfStorageDisableMPU = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* A value of "true" disables uploading in multiple chunks. Files are uploaded to
  the destination bucket as a single (large) chunk.
* Optional.
* Default: false

archiver.selfStorageEncryption = sse-s3 | none
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the scheme to use for server-side encryption for Self Storage.
* A value of sse-s3 enables SSE-S3 server-side encryption mode on Amazon S3 for
  Self Storage.
* A value of 'none' disables server-side encryption. Data is stored unencrypted
  on the Self Storage.
* Optional.
* Default: sse-s3

#*******************************************************************
# Dynamic Data Archive lets you move your data from your Splunk Cloud indexes to a
# storage location. You can configure Splunk Cloud to automatically move the data
# in an index when the data reaches the end of the Splunk Cloud retention period
```

```
# you configure. In addition, you can restore your data to Splunk Cloud if you need
# to perform some analysis on the data.
# For each index, you can use Dynamic Data Self Storage or Dynamic Data Archive,
# but not both.
#
# For example, use the following settings to configure Dynamic Data Archive.
#   archiver.coldStorageProvider        = Glacier
#   archiver.coldStorageRetentionPeriod = 365
#*******************************************************************
archiver.coldStorageProvider = <string>
* This feature is supported on Splunk Cloud only.
 Do not configure this setting in a Splunk Enterprise environment.
* Specifies the storage provider for Dynamic Data Archive.
* Optional. Only required when using Dynamic Data Archive.
* The only providers currently supported are Glacier and GCSArchive for
  Amazon Web Services (AWS) and Google Cloud Platform (GCP), respectively.
* NOTE: This setting value is case-sensitive.

archiver.coldStorageRetentionPeriod = <unsigned integer>
* This feature is supported on Splunk Cloud only.
 Do not configure this setting in a Splunk Enterprise environment.
* Defines how long Splunk will maintain data in days, including the
  archived period.
* Optional. Only required when using Dynamic Data Archive.
* Must be greater than 0

archiver.enableDataArchive = <boolean>
* This feature is supported on Splunk Cloud only.
 Do not configure this setting in a Splunk Enterprise environment.
* If set to true, Dynamic Data Archiver is enabled for the index.
* Default: false

archiver.maxDataArchiveRetentionPeriod = <nonnegative integer>
* This feature is supported on Splunk Cloud only.
 Do not configure this setting in a Splunk Enterprise environment.
* The maximum total time in seconds, that data for the specified index is
  maintained by Splunk, including the archived period.
* The archiver.maxDataArchiveRetentionPeriod controls the maximum value of the
  coldStorageRetentionPeriod. coldStorageRetentionPeriod cannot exceed this
  value.
* Default: 0

#*******************************************************************
# Volume settings.  This section describes settings that affect the volume-
# optional and volume-mandatory settings only.
#
# All volume stanzas begin with "volume:". For example:
#   [volume:volume_name]
#   path = /foo/bar
#
# These volume stanzas can then be referenced by individual index
# settings, e.g. homePath or coldPath.  To refer to a volume stanza, use
# the "volume:" prefix. For example, to set a cold DB to the example stanza
# above, in index "hiro", use:
#   [hiro]
#   coldPath = volume:volume_name/baz
# This will cause the cold DB files to be placed under /foo/bar/baz.  If the
# volume spec is not followed by a path
# (e.g.  "coldPath=volume:volume_name"), then the cold path would be
# composed by appending the index name to the volume name ("/foo/bar/hiro").
#
# If "path" is specified with a URI-like value (e.g., "s3://bucket/path"),
```

```
# this is a remote storage volume.  A remote storage volume can only be
# referenced by a remotePath setting, as described above.  An Amazon S3
# remote path might look like "s3://bucket/path", whereas an NFS remote path
# might look like "file:///mnt/nfs".  The name of the scheme ("s3" or "file"
# from these examples) is important, because it can indicate some necessary
# configuration specific to the type of remote storage.  To specify a
# configuration under the remote storage volume stanza, you use settings
# with the pattern "remote.<scheme>.<param name>". These settings vary
# according to the type of remote storage.  For example, remote storage of
# type S3 might require that you specify an access key and a secret key.
# You would do this through the "remote.s3.access_key" and
# "remote.s3.secret_key" settings.
#
# Note: thawedPath may not be defined in terms of a volume.
# Thawed allocations are manually controlled by Splunk administrators,
# typically in recovery or archival/review scenarios, and should not
# trigger changes in space automatically used by normal index activity.
#*******************************************************************************

storageType = local | remote
* Optional.
* Specifies whether the volume definition is for indexer local storage or remote
  storage. Only the 'remotePath' setting references a remote volume.
* Default: "local"

path = <path on server>
* Required.
* If storageType is set to its default value of "local":
  * The 'path' setting points to the location on the file system where all
    indexes that will use this volume reside.
  * This location must not overlap with the location for any other volume
    or index.
* If storageType is set to "remote":
  * The 'path' setting points to the remote storage location where indexes
    reside.
  * The format for this setting is: <scheme>://<remote-location-specifier>
    * The "scheme" identifies a supported external storage system type.
    * The "remote-location-specifier" is an external system-specific string for
      identifying a location inside the storage system.
    * For Google Cloud Storage, this is specified as "gs://<bucket-name>/path/to/splunk/db"
    * For Microsoft Azure Blob storage, this is specified
      as "azure://<container-name>/path/to/splunk/db" Note that "<container-name>"
      is needed here only if 'remote.azure.container_name' is not set.

maxVolumeDataSizeMB = <positive integer>
* If set, this setting limits the total size of all databases that reside
  on this volume to the maximum size specified, in MB.  Note that this it
  will act only on those indexes which reference this volume, not on the
  total size of the path set in the 'path' setting of this volume.
* If the size is exceeded, splunkd removes buckets with the oldest value
  of latest time (for a given bucket) across all indexes in the volume,
  until the volume is below the maximum size. This is the trim operation.
  This can cause buckets to be chilled [moved to cold] directly
  from a hot DB, if those buckets happen to have the least value of
  latest-time (LT) across all indexes in the volume.
* The highest legal value is 4294967295.
* The lowest legal value is 1.
* Optional.
* This setting is ignored when 'storageType' is set to "remote" or
  when set to "local" and the volume contains any remote-storage enabled indexes.

rotatePeriodInSecs = <nonnegative integer>
```

```
* Optional, ignored for storageType=remote
* Specifies period of trim operation for this volume.
* The highest legal value is 4294967295.
* Default: The global 'rotatePeriodInSecs' value

remote.* = <string>
* With remote volumes, communication between the indexer and the external
  storage system may require additional configuration, specific to the type of
  storage system. You can pass configuration information to the storage
  system by specifying the settings through the following schema:
  remote.<scheme>.<config-variable> = <value>.
  For example: remote.s3.access_key = ACCESS_KEY
* Optional.

############################################################
##### S3 specific settings
############################################################

remote.s3.header.<http-method-name>.<header-field-name> = <string>
* Enable server-specific features, such as reduced redundancy, encryption,
  and so on, by passing extra HTTP headers with the REST requests.
  The <http-method-name> can be any valid HTTP method. For example, GET,
  PUT, or ALL, for setting the header field for all HTTP methods.
* Example: remote.s3.header.PUT.x-amz-storage-class = REDUCED_REDUNDANCY
* Optional.

remote.s3.access_key = <string>
* Specifies the access key to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, the indexer will look for these environment variables:
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the access key from the IAM role.
* Unencrypted access key cannot begin with "$1$" or "$7$". These prefixes are reserved
  for use by Splunk software to signify that the access key is already encrypted.
* Optional.
* No default.

remote.s3.secret_key = <string>
* Specifies the secret key to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, the indexer will look for these environment variables:
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the secret key from the IAM role.
* Unencrypted secret key cannot begin with "$1$" or "$7$". These prefixes are reserved
  for use by Splunk software to signify that the secret key is already encrypted.
* Optional.
* No default.

remote.s3.list_objects_version = v1|v2
* The AWS S3 Get Bucket (List Objects) Version to use.
* See AWS S3 documentation "GET Bucket (List Objects) Version 2" for details.
* Default: v1

remote.s3.signature_version = v2|v4
* The signature version to use when authenticating with the remote storage
  system supporting the S3 API.
* For 'sse-kms' and 'sse-c' server-side encryption schemes, and for 'cse'
  client-side encryption scheme, you must use signature_version=v4.
* For signature_version=v2 you must set url_version=v1.
* Optional.
```

```
* Default: v4

remote.s3.url_version = v1|v2
* Specifies which url version to use, both for parsing the endpoint/path, and
* for communicating with the remote storage. This value only needs to be
* specified when running on non-AWS S3-compatible storage that has been configured
* to use v2 urls.
* In v1 the bucket is the first element of the path.
* Example: mydomain.com/bucketname/rest/of/path
* In v2 the bucket is the outermost subdomain in the endpoint.
* Exmaple: bucketname.mydomain.com/rest/of/path
* Default: v1

remote.s3.auth_region = <string>
* The authentication region to use for signing requests when interacting
  with the remote storage system supporting the S3 API.
* Used with v4 signatures only.
* If unset and the endpoint (either automatically constructed or explicitly
  set with remote.s3.endpoint setting) uses an AWS URL (for example,
  https://<bucketname>.s3-us-west-1.amazonaws.com), the instance attempts to extract
  the value from the endpoint URL (for example, "us-west-1").  See the
  description for the remote.s3.endpoint setting.
* If unset and an authentication region cannot be determined, the request
  will be signed with an empty region value. This can lead to rejected
  requests when using non-AWS S3-compatible storage.
* Optional.
* No default.

remote.s3.use_delimiter = <boolean>
* Specifies whether a delimiter (currently "guidSplunk") should be
  used to list the objects that are present on the remote storage.
* A delimiter groups objects that have the same delimiter value
  so that the listing process can be more efficient as it
  does not need to report similar objects.
* Optional.
* Default: true

remote.s3.supports_versioning = <boolean>
* Specifies whether the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage.
* This setting determines how splunkd removes data from remote storage.
  If set to true, splunkd will delete all versions of objects at
  time of data removal. Otherwise, if set to false, splunkd will use a simple DELETE
  (See https://docs.aws.amazon.com/AmazonS3/latest/dev/DeletingObjectVersions.html).
* Optional.
* Default: true

remote.s3.endpoint = <URL>
* The URL of the remote storage system supporting the S3 API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified and the indexer is running on EC2, the endpoint will be
  constructed automatically based on the EC2 region of the instance where the
  indexer is running, as follows: https://<bucketname>.s3-<region>.amazonaws.com
* Example: https://<bucketname>.s3-us-west-2.amazonaws.com
* Optional.

remote.s3.bucket_name = <string>
* Specifies the S3 bucket to use when endpoint isn't set.
* Example
  path = s3://path/example
```

```
  remote.s3.bucket_name = mybucket
* Used for constructing the amazonaws.com hostname, as shown above.
* If neither endpoint nor bucket_name is specified, the bucket is assumed
  to be the first path element.
* Optional.

remote.s3.tsidx_compression = <boolean>
* Whether or not the indexer compresses tsidx files before it uploads them to S3.
  A value of "true" means the indexer compresses tsidx files before it uploads
  them to S3.
* Ensure that every indexer is running Splunk version 9.0.0 or above before
  enabling this feature.
* This feature is not backward compatible. Once activated, you will not be able
  to downgrade Splunk to versions earlier than 9.0.0.
* Default: false

remote.s3.multipart_download.part_size = <unsigned integer>
* Sets the download size of parts during a multipart download.
* This setting uses HTTP/1.1 Range Requests (RFC 7233) to improve throughput
  overall and for retransmission of failed transfers.
* The special value of 0 disables downloading in multiple parts. In other
  words, files will always get downloaded as a single (large) part.
* Do not change this value unless that value has been proven to improve
  throughput.
* Optional.
* Minimum value: 5242880 (5 MB)
* Default: 134217728 (128 MB)

remote.s3.multipart_upload.part_size = <unsigned integer>
* Sets the upload size of parts during a multipart upload.
* The special value of 0 disables uploading in multiple parts. In other
  words, files will always get uploaded as a single (large) part.
* Optional.
* Minimum value: 5242880 (5 MB)
* Default: 134217728 (128 MB)

remote.s3.multipart_max_connections = <unsigned integer>
* Specifies the maximum number of HTTP connections to have in progress for
  either multipart download or upload.
* A value of 0 means unlimited.
* Default: 8

remote.s3.max_idle_connections = <unsigned integer>
* Specifies the maximum number of idle HTTP connections that can be pooled for
  reuse by the S3 client when connecting to the S3 server.
* A value of 0 means pooling of connections is disabled.
* Default: 25

remote.s3.enable_data_integrity_checks = <boolean>
* If set to true, Splunk sets the data checksum in the metadata field of the
  HTTP header during upload operation to S3.
* The checksum is used to verify the integrity of the data on uploads.
* Default: false

remote.s3.enable_signed_payloads  = <boolean>
* If set to true, Splunk signs the payload during upload operation to S3.
* Valid only for remote.s3.signature_version = v4
* Default: true

remote.s3.retry_policy = max_count
* Sets the retry policy to use for remote file operations.
* A retry policy specifies whether and how to retry file operations that fail
```

```
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a file operation will be
    retried upon intermittent failure both for individual parts of a multipart
    download or upload and for files as a whole.
* Optional.
* Default: max_count

remote.s3.max_count.max_retries_per_part = <unsigned integer>
* When the 'remote.s3.retry_policy' setting is "max_count", sets the maximum number
  of times a file operation will be retried upon intermittent failure.
* The count is maintained separately for each file part in a multipart download
  or upload.
* Optional.
* Default: 9

remote.s3.max_count.max_retries_in_total = <unsigned integer>
* When the remote.s3.retry_policy setting is max_count, sets the maximum number
  of times a file operation will be retried upon intermittent failure.
* The count is maintained for each file as a whole.
* Optional.
* Default: 128

remote.s3.timeout.connect = <unsigned integer>
* Set the connection timeout, in milliseconds, to use when interacting
  with S3 for this volume
* Optional.
* Default: 5000

remote.s3.timeout.read = <unsigned integer>
* Set the read timeout, in milliseconds, to use when interacting with
  S3 for this volume
* Optional.
* Default: 60000

remote.s3.timeout.write = <unsigned integer>
* Set the write timeout, in milliseconds, to use when interacting with
  S3 for this volume
* Optional.
* Default: 60000

remote.s3.sslVerifyServerCert = <boolean>
* If this is set to true, Splunk verifies certificate presented by S3
  server and checks that the common name/alternate name matches the ones
  specified in 'remote.s3.sslCommonNameToCheck' and
  'remote.s3.sslAltNameToCheck'.
* Optional
* Default: false

remote.s3.sslVersions = <versions_list>
* Comma-separated list of SSL versions to connect to 'remote.s3.endpoint'.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list
  but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Optional.
* Default: tls1.2
```

```
remote.s3.sslCommonNameToCheck = <commonName1>, <commonName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to true,
  splunkd checks the common name of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this list
  of common names.
* Default: not set

remote.s3.sslAltNameToCheck = <alternateName1>, <alternateName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to true,
  splunkd checks the alternate name(s) of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this list of
  subject alternate names.
* No default.

remote.s3.sslRootCAPath = <path>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together. S3 certificate
  will be validated against the CAs present in this file.
* Optional.
* Default: [sslConfig/caCertFile] in server.conf

remote.s3.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Optional.
* Default: TLSv1+HIGH:TLSv1.2+HIGH:@STRENGTH

remote.s3.ecdhCurves = <comma-separated list>
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only supports named curves specified
  by their SHORT names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Optional.
* No default.

remote.s3.dhFile = <path>
* PEM format Diffie-Hellman parameter file name.
* DH group size should be no less than 2048bits.
* This file is required in order to enable any Diffie-Hellman ciphers.
* Optional.
* No default.

remote.s3.encryption = sse-s3 | sse-kms | sse-c | cse | none
* The encryption scheme to use for data buckets that are currently being stored (data at rest).
* sse-s3: Search for "Protecting Data Using Server-Side Encryption with Amazon S3-Managed
          Encryption Keys" on the Amazon Web Services documentation site.
* sse-kms: Search for "Protecting Data Using Server-Side Encryption with CMKs Stored in AWS
           Key Management Service (SSE-KMS)" on the Amazon Web Services documentation site.
* sse-c: Search for "Protecting Data Using Server-Side Encryption with Customer-Provided Encryption
         Keys (SSE-C)" on the Amazon Web Services documentation site.
* cse: Search for "SmartStore client-side encryption" on the Splunk Enterprise documentation site,
  and "Protecting Data Using Server-Side Encryption with Customer-Provided Encryption Keys (SSE-C)"
  on the Amazon Web Services documentation site.
* Optional.
* Default: none
```

```
remote.s3.encryption.sse-c.key_type = kms
* Determines the mechanism Splunk uses to generate the key for sending over to
  S3 for SSE-C.
* The only valid value is 'kms', indicating Amazon Web Services Key Management Service (AWS KMS).
* One must specify required KMS settings: e.g. remote.s3.kms.key_id
  for Splunk to start up while using SSE-C.
* Optional.
* Default: kms

remote.s3.encryption.sse-c.key_refresh_interval = <unsigned integer>
* Specifies period in seconds at which a new key will be generated and used
  for encrypting any new data being uploaded to S3.
* Optional.
* Default: 86400

remote.s3.encryption.cse.algorithm = aes-256-gcm
* Currently not supported. This setting is related to a feature that is
  still under development.
* The encryption algorithm to use for bucket encryption while
  client-side encryption is enabled.
* Optional.
* Default: aes-256-gcm

remote.s3.encryption.cse.key_type = kms
* Currently not supported. This setting is related to a feature that is
  still under development.
* The mechanism that the Splunk platform uses to generate the key
  for client-side encryption.
* The only valid value is 'kms', indicating AWS KMS service.
* You must specify the required KMS settings, for example, 'remote.s3.kms.key_id'
  for the Splunk platform to start with client-side encryption active.
* Optional.
* Default: kms

remote.s3.encryption.cse.key_refresh_interval = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The interval, in seconds, at which the Splunk platform generates a new key and uses
  it to encrypt any data that it uploads to S3 when client-side encryption is active.
* Optional.
* Default: 86400

remote.s3.encryption.cse.tmp_dir = <path>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The full path to the directory where the Splunk platform temporarily stores encrypted files.
* Optional.
* Default: $SPLUNK_HOME/var/run/splunk/cse-tmp

remote.s3.kms.endpoint = <string>
* Indicates the host name to use when server-side or client-side encryption
  is enabled e.g. https://internal-kms.mycompany.com:8443
* If not set, SmartStore uses 'remote.s3.kms.auth_region' to
  determine the endpoint.
* Optional.
* No default.

remote.s3.kms.key_id = <string>
* Required if remote.s3.encryption = sse-c | sse-kms | cse
* Specifies the identifier for Customer Master Key (CMK) on KMS. It can be the
  unique key ID or the Amazon Resource Name (ARN) of the CMK or the alias
  name or ARN of an alias that refers to the CMK.
```

```
* Examples:
  Unique key ID: 1234abcd-12ab-34cd-56ef-1234567890ab
  CMK ARN: arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
  Alias name: alias/ExampleAlias
  Alias ARN: arn:aws:kms:us-east-2:111122223333:alias/ExampleAlias
* No default.

remote.s3.kms.access_key = <string>
* Similar to 'remote.s3.access_key'.
* If not specified, KMS access uses 'remote.s3.access_key'.
* Optional.
* No default.

remote.s3.kms.secret_key = <string>
* Similar to 'remote.s3.secret_key'.
* If not specified, KMS access uses 'remote.s3.secret_key'.
* Optional.
* No default.

remote.s3.kms.auth_region = <string>
* Required if 'remote.s3.auth_region' is unset and Splunk can not
  automatically extract this information.
* Similar to 'remote.s3.auth_region'.
* If not specified, KMS access uses 'remote.s3.auth_region'.
* No default.

remote.s3.kms.max_concurrent_requests = <unsigned integer>
* Optional.
* Limits maximum concurrent requests to KMS from this Splunk instance.
* NOTE: Can severely affect search performance if set to very low value.
* Default: 10

remote.s3.kms.<ssl_settings> = <...>
* Optional.
* Check the descriptions of the SSL settings for remote.s3.<ssl_settings>
  above. e.g. remote.s3.sslVerifyServerCert.
* Valid ssl_settings are sslVerifyServerCert, sslVersions, sslRootCAPath,
  sslAltNameToCheck, sslCommonNameToCheck, cipherSuite, ecdhCurves, and dhFile.
* All of these settings are optional.
* All of these settings have the same defaults as
  'remote.s3.<ssl_settings>'.

remote.s3.max_download_batch_size = <unsigned integer>
* The maximum number of objects that can be downloaded in a single batch
  from remote storage. If the number of objects to be downloaded exceeds
  this value, the indexer downloads the objects in multiple batches.
* Default: 50

remote.s3.use_sdk = true|false|auto
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies whether to use the AWS C++ SDK or make direct HTTP requests to
  the S3 or S3-compatible storage endpoint.
* If auto is specified, the SDK will be used if the storage provider is S3
  and HTTP requests will be used if the storage provider is not S3, but is
  S3-compatible.
* Default: false

remote.s3.data_integrity_validation = disabled | sha256
* Specifies the signature algorithm that SmartStore uses to generate file
  signatures in buckets. SmartStore uses file signatures to test the data
  integrity of buckets.
```

```
* A value of "disabled" means that SmartStore ignores existing file signatures
  in buckets.
* A value of "sha256" means SmartStore uses the SHA-256 encryption algorithm
  to generate signatures.
* This setting is optional.
* Default: disabled

federated.provider = <provider_name>
* Identifies the federated provider on which this search is run.
* Select the stanza for the federated provider defined in the federated.conf file.
* Default: ""

federated.dataset = <string>
* Identifies the dataset located on the federated providers.
* The dataset takes a format of <prefix>:<remote_name>.
* If the 'federated.provider' is a "splunk" type provider:
  * <prefix> can be "index", "datamodel", "lastjob", or "savedsearch".
  * <remote_name> is the name of an index, data model, or saved search,
    depending on the <prefix> value. The dataset must be defined on the remote
    search head. A saved search name can be provided as the <remote_name> for
    both the savedsearch and lastjob <prefix> options.
* If the 'federated.provider' is an "aws_s3" type provider:
  * <prefix> must be "aws_glue_table".
    * <remote_name> is the name of an AWS Glue Data Catalog table that is used
      as a dataset schema.
      * The AWS Glue Data Catalog table contains metadata that represents data
        in an Amazon S3 data store.
      * This table is in your AWS Glue Data Catalog if you have set up a
        dataset with AWS Glue.
      * You use the 'sdselect' command to run federated searches against the
        AWS Glue Data Catalog table.
      * Provide "aws_glue_table" as  <prefix> only if the 'federated.provider'
        has non-empty 'database' and 'aws_glue_tables_allowlist' settings.
* If <prefix> is not defined, <prefix> defaults to 'index'.
* No default


############################################################
##### Google Cloud Storage settings
############################################################

remote.gs.credential_file = <credentials.json>
* Name of the json file with GCS credentials.
* For standalone indexers, this file must be located in the $SPLUNK_HOME/etc/auth
  directory.
* For indexer clusters, this file must be located either in the _cluster/local
  directory of the distributed bundle or the $SPLUNK_HOME/etc/auth directory.
  The distributed bundle location has precedence.
* You must set either this setting or 'service_account_email' to use
  custom credentials.
* The indexer tries different ways of providing credentials in the following order:
  1. This setting, for the json credential file, is used if it is set.
  2. The 'service_account_email' setting is used if it is set.
  3. The credential for the Compute Engine's default service_account is used.
  The last two methods both require that the indexer is running on GCP.
* The specified file is encrypted on startup.
* Optional if the indexer is running on GCP.
* Required if the indexer is not running on GCP.
* Default: Not set.

remote.gs.service_account_email = <email-address>
* Credential of the specified custom service_account is used.
```

```
* This service_account must be associated with every Compute Engine
  instance used with SmartStore-enabled indexer cluster.
* This setting uses GCP metadata server to get the credential. It requires
  the indexer to be running on GCP.
* This setting is used only if the 'credential_file' setting is unset. For
  more information, see the entry for the 'credential_file' setting.
* Optional
* Default: Not set.

remote.gs.project_id = <string>
* The ID of the GCP project associated with the volume.
* The project ID is a unique string across Google Cloud. It can found in GCP console.
* Required if 'remote.gs.encryption' is set to gcp-sse-c or gcp-sse-kms.
* Must be left unset if 'remote.gs.encryption' is set to gcp-sse-gcp.
* Default: Not set.

remote.gs.upload_chunk_size = <unsigned integer>
* Specifies the maximum size, in bytes, for file chunks in a parallel upload.
* A value of 0 disables uploading in multiple chunks. Files are uploaded
  as a single (large) chunk.
* Minimum value: 5242880 (5 MB)
* Default: 33554432 (32MB)

remote.gs.download_chunk_size = <unsigned integer>
* Specifies the maximum size for file chunks in a parallel download.
* Specify as bytes
* Minimum value: 5242880 (5 MB)
* Default: 33554432 (32MB)

remote.gs.max_parallel_non_upload_threads = <unsigned integer>
* Number of threads used for parallel downloads and other async gcs
  operations, per index volume.
* This is the total count across all such operations.
* This does not include parallel upload operations, which are specified
  with the 'max_threads_per_parallel_upload' setting.
* For SmartStore, this is only used for parallel download of files.
* Default: 250

remote.gs.max_threads_per_parallel_upload = <unsigned integer>
* Number of threads used for a single parallel upload operation.
* Default: 64

remote.gs.max_connection_pool_size = <unsigned integer>
* Size of the connection pool to the remote storage per index volume.
* Default: 500

remote.gs.max_download_batch_size = <unsigned integer>
* The maximum number of objects that can be downloaded in a single batch
  from remote storage. If the number of objects to be downloaded exceeds
  this value, the indexer downloads the objects in multiple batches.
* Default: 50

remote.gs.remove_all_versions = <boolean>
* If true, a remove operation on an object explicitly deletes all versions
  of that object.
* Default: true

remote.gs.use_delimiter = <boolean>
* Specifies whether a delimiter (currently "guidSplunk") should be
  used to list the objects that are present on the remote storage.
* A delimiter groups objects that have the same delimiter value
  so that the listing process can be more efficient as it
```

```
  does not need to report similar objects.
* Optional.
* Default: true

remote.gs.retry_policy = max_count
* Sets the retry policy to use for remote file operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times an operation will be
    retried upon intermittent failure
* Default: max_count

remote.gs.max_count.max_retries_per_part = <unsigned integer>
* When the remote.gs.retry_policy setting is max_count, sets the maximum number
  of times a file operation will be retried upon intermittent failure.
* The count is maintained separately for each file part in a multipart download
  or upload.
* Default: 9

remote.gs.backoff.initial_delay_ms = <unsigned integer>
* If retries are enabled, an exponential backoff interval is used to perform
  the retries.
* This setting specifies the delay for the first retry, in milliseconds.
* Default: 3000 (3s)

remote.gs.backoff.max_delay_ms = <unsigned integer>
* If retries are enabled, an exponential backoff interval is used to perform
  the retries.
* This setting specifies the maximum delay before the next retry, in milliseconds
* Default: 60000 (60s)

remote.gs.backoff.scaling = <unsigned integer>
* If retries are enabled, an exponential backoff interval is used to perform
  the retries.
* This setting specifies the amount by which subsequent delays are scaled,
  upto max_delay_ms.
* Default: 2

remote.gs.connectUsingIpVersion = auto|4-only|6-only
* When making outbound connections to the storage service, this setting
  controls whether connections are made using IPv4 or IPv6.
* Connections to literal IPv4 or IPv6 addresses are unaffected by this setting.
* "4-only" : Splunkd only attempts to connect to the IPv4 address.
* "6-only" : Splunkd only attempts to connect to the IPv6 address.
* "auto":
    * If [general]/listenOnIPv6 in server.conf is set to "only", this defaults
      to "6-only"
    * Otherwise, this defaults to "4-only"
* Default: auto

remote.gs.sslVersionsForClient = ssl3|tls1.0|tls1.1|tls1.2
* Defines the minimum ssl/tls version to use for outgoing connections.
* Default: tls1.2

remote.gs.sslVerifyServerCert = <boolean>
* If set to true, Splunkd authenticates the certificate of the services
  it connects to by using the configured CA.
* Default: false.

remote.gs.sslVerifyServerName = <boolean>
* Whether or not splunkd, as a client, performs a TLS hostname validation check
```

```
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

remote.gs.sslRootCAPath = <path>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together. Google Storage and
  related service certificates will be validated against the CAs in this file.
* Default: value of [sslConfig]/caCertFile in server.conf

remote.gs.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Default: value of [sslConfig]/cipherSuite in server.conf

remote.gs.encryption = gcp-sse-c | gcp-sse-kms | gcp-sse-gcp
* The encryption scheme to use for index buckets while stored on GCS (data-at-rest).
* gcp-sse-c: Maps to GCP customer-supplied encryption keys. See Google Cloud documentation for details.
* gcp-sse-kms: Maps to GCP customer-managed encryption keys. See Google Cloud documentation for details.
* gcp-sse-gcp: Maps to GCP Google-managed encryption keys. See Google Cloud documentation for details.
* Google Cloud always encrypts the incoming data on the server side.
* For the gcp-sse-kms scheme, you must grant your Cloud Storage service account permission to use
  your Cloud KMS key. For more details, search for "Assigning a Cloud KMS key to a service account"
  on the Google Cloud documentation site. To find your Cloud Storage service account, search for
  "Getting the Cloud Storage service account".
* Default: gcp-sse-gcp

remote.gs.encryption.gcp-sse-c.key_type = gcp_kms
* Affects only the gcp-sse-c encryption scheme.
* Determines the mechanism the indexer uses to generate the key for sending data to GCS.
* The only valid value is 'gcp_kms', indicating Google Cloud Key Management Service (GCP KMS).
* You must also specify the required KMS settings: 'remote.gs.gcp_kms.locations',
  'remote.gs.gcp_kms.key_ring' and 'remote.gs.gcp_kms.key'. If you do not specify
  those settings, the indexer cannot start while using gcp-sse-c.
* Default: gcp_kms

remote.gs.encryption.gcp-sse-c.key_refresh_interval = <unsigned integer>
* Specifies the interval, in seconds, for generating a new key that is used
  for encrypting data uploaded to GCS.
* Default: 86400

remote.gs.gcp_kms.locations = <string>
```

```
* Required if 'remote.gs.encryption' is set to gcp-sse-c or gcp-sse-kms.
* Specifies the geographical regions where KMS key rings and keys are stored for access.
* Google Cloud offers three types of locations: regional ones such as "us-central1",
  dual-regional ones such as "nam4", and multi-regional ones such as "global" and "us".
  Search for "Cloud KMS locations" on the Google Cloud documentation site for a complete list.
* For best performance, choose a key ring and a key in the same location as the cloud stack.
* Default: none.

remote.gs.gcp_kms.key_ring = <string>
* Required if 'remote.gs.encryption' is set to gcp-sse-c or gcp-sse-kms.
* Specifies the name of the  key ring used for encryption when uploading data to GCS.
* In Google Cloud, a key ring is a grouping of keys for organizational purposes. A key ring
  belongs to a Google Cloud Project and resides in a specific location. Search for "key ring"
  on the Google Cloud documentation site for more details.
* Default: none.

remote.gs.gcp_kms.key = <string>
* Required if 'remote.gs.encryption' is set to gcp-sse-c or gcp-sse-kms.
* Specifies the name of the encryption key used for uploading data to GCS.
* Default: none.

remote.gs.data_integrity_validation = disabled | sha256
* Specifies the signature algorithm that SmartStore uses to generate file
  signatures in buckets. SmartStore uses file signatures to test the data
  integrity of buckets.
* A value of "disabled" means that SmartStore ignores existing file signatures
  in buckets.
* A value of "sha256" means SmartStore uses the SHA-256 encryption algorithm
  to generate signatures.
* This setting is optional.
* Default: disabled

############################################################
##### Microsoft Azure Storage settings
############################################################

remote.azure.use_delimiter = <boolean>
* Specifies whether a delimiter (currently "guidSplunk") should be
  used to list the objects that are present on the remote storage.
* A delimiter groups objects that have the same delimiter value
  so that the listing process can be more efficient as it
  does not need to report similar objects.
* Default: true

remote.azure.sslVersions = ssl3|tls1.0|tls1.1|tls1.2
* Specifies the minimum SSL/TLS version to use for outgoing connections.
* Default: tls1.2

remote.azure.sslVerifyServerCert = <boolean>
* If set to true, the indexer cache manager authenticates the certificate of
  the services it connects to by using the configured CA.
* Default: false.

remote.azure.sslVerifyServerName = <boolean>
* Whether or not splunkd, as a client, performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
```

```
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

remote.azure.httpKeepAlive = <boolean>
* If set to true, All successful requests to the Microsoft Azure Storage API
  will keep the connection channel open to the remote storage service
* Default: true.

remote.azure.access_key = <string>
* Specifies the access key (storage account name) to use when authenticating
  with the remote storage system supporting the Microsoft Azure Storage API.
* If a value is not specified for the 'remote.azure.endpoint' setting, the
  value of this setting is used to construct the remote storage URI.
  For example: "https://<remote.azure.access_key>.blob.core.windows.net"
* No default.

remote.azure.secret_key = <string>
* Specifies the secret key to use when authenticating with the remote storage
  system supporting the Microsoft Azure Storage API.
* No default.

remote.azure.tenant_id = <string>
* Specifies the ID of the tenant (instance of an Azure AD directory). Check
  your Azure subscription for details.
* Needed only for client token-based authentication.
* No default.

remote.azure.client_id = <string>
* Specifies the ID of the client (also called application ID - the unique
  identifier Azure AD issues to an application registration that identifies a
  specific application and the associated configurations).
  You can obtain the client ID for an application from the Azure Portal in the
  Overview section for the registered application.
* Needed only for client token-based authentication.
* Optional for managed identity authentication.
* No default.

remote.azure.client_secret = <string>
* Specifies the secret key to use when authenticating using the client_id. You
  generate the secret key through the Azure Portal.
* Needed only for client token-based authentication.
* No default.

remote.azure.sslRootCAPath = <path>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together. Microsoft Azure
  Storage and related service certificates will be validated against the CAs
  in this file.
```

```
* Default: value of [sslConfig]/caCertFile in server.conf

remote.azure.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Default: value of [sslConfig]/cipherSuite in server.conf

remote.azure.encryption = azure-sse-kv | azure-sse-ms
* The encryption scheme to use for containers that are currently being stored.
* azure-sse-kv: Maps to Azure customer-managed keys in a key vault. See
  See the Azure documentation for customer-managed keys for Azure Store
  encryption for details.
* azure-sse-ms: Maps to Azure Microsoft-managed keys in Microsoft key store.
  See the Azure documentation for Azure Storage encryption for data at rest for
  details.
* Default: azure-sse-ms

remote.azure.azure-sse-kv.encryptionScope = <string>
* Required if remote.azure.encryption = azure-sse-kv
* Specifies the key used for encrypting blobs within the scope of this index.
* No default.

remote.azure.supports_versioning = <boolean>
* Specifies whether the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage.
* This setting determines how the indexer cache manager removes data from
  remote storage.
  If set to false, the indexer cache manager will delete all versions of
  objects at time of data removal. Otherwise, if set to false, the indexer
  cache manager will use a simple DELETE.
  For more information on Azure versioning, see the Microsoft Azure
  documentation.
* Default: true

remote.azure.endpoint = <URL>
* The URL of the Microsoft Azure Storage endpoint supporting the Azure
  REST API.
* The scheme, http or https, can be used to enable or disable SSL
  connectivity with the endpoint.
* The value of this setting must point to an Azure Blob storage location, not a
  container name. the container name should not be specified here but given
  seperately in either 'remote.azure.container_name' or as part of 'path'
* Example: https://<account-name>.blob.core.windows.net/

remote.azure.container_name = <string>
* Specifies the Azure container to use complying with Microsoft Azure
  Storage Container naming convention.

remote.azure.upload.chunk_size = <unsigned integer>
* Specifies the maximum size for file chunks in a parallel upload.
* Specify as bytes
* Default: 78643200 (75MB)

remote.azure.upload.concurrency = <unsigned integer>
* Specifies the number of threads used for a single parallel upload operation.
* Default: 5

remote.azure.download.chunk_size = <unsigned integer>
* Specifies the maximum size for file chunks in a parallel download.
* Specify as bytes
* Default: 78643200 (75MB)
```

```
remote.azure.download.concurrency = <unsigned integer>
* Specifies the number of threads used for a single parallel download operation.
* Default: 5

remote.azure.max_download_batch_size = <unsigned integer>
* The maximum number of objects that can be downloaded in a single batch
  from remote storage. If the number of objects to be downloaded exceeds
  this value, the indexer downloads the objects in multiple batches.
* Default: 50

remote.azure.max_listing_page_size = <unsigned integer>
* The maximum number of blobs returned in a single list query operation.
* Default: 1000

remote.azure.retry_policy = max_count
* Sets the retry policy to use for remote file operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times an operation will be
    retried upon intermittent failure
* Default: max_count

remote.azure.max_count.max_retries_in_total = <unsigned integer>
* When the remote.azure.retry_policy setting is max_count, sets the maximum
  number of times a file operation will be retried upon intermittent failure.
* The count is maintained for each file as a whole.
* Optional.
* Default: 3

remote.azure.backoff.initial_delay_ms = <unsigned integer>
* If retries are enabled, a backoff interval is used to perform
  the retries. This interval is doubled on each retry up to the limit set in
  remote.azure.backoff.max_retry_delay_ms.
* This setting specifies the delay between each retry, in milliseconds.
* Default: 4000 (4s)

remote.azure.backoff.max_retry_delay_ms = <unsigned integer>
* If retries are enabled, a backoff interval is used to perform
  the retries.
* This setting specifies the maximum delay before the next retry, in
  milliseconds
* Default: 2 * 60 * 1000 (120s)

remote.azure.data_integrity_validation = disabled | sha256
* Specifies the signature algorithm that SmartStore uses to generate file
  signatures in buckets. SmartStore uses file signatures to test the data
  integrity of buckets.
* A value of "disabled" means that SmartStore ignores existing file signatures
  in buckets.
* A value of "sha256" means SmartStore uses the SHA-256 encryption algorithm
  to generate signatures.
* This setting is optional.
* Default: disabled
```

indexes.conf.example

```
#   Version 9.2.2
```

```
#
# This file contains an example indexes.conf.  Use this file to configure
# indexing properties.
#
# To use one or more of these configurations, copy the configuration block
# into indexes.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#

# The following example defines a new high-volume index, called "hatch", and
# sets this to be the default index for both incoming data and search.
#
# Note that you may want to adjust the indexes that your roles have access
# to when creating indexes (in authorize.conf)

defaultDatabase = hatch

[hatch]

homePath   = $SPLUNK_DB/hatchdb/db
coldPath   = $SPLUNK_DB/hatchdb/colddb
thawedPath = $SPLUNK_DB/hatchdb/thaweddb
maxDataSize = 10000
maxHotBuckets = 10



# The following example changes the default amount of space used on a
# per-index basis.

[default]
maxTotalDataSizeMB = 650000
maxGlobalRawDataSizeMB = 0
maxGlobalDataSizeMB = 0


# The following example changes the time data is kept around by default.
# It also sets an export script.  NOTE: You must edit this script to set
# export location before running it.

[default]
maxWarmDBCount = 200
frozenTimePeriodInSecs = 432000
rotatePeriodInSecs = 30
coldToFrozenScript = "$SPLUNK_HOME/bin/python" "$SPLUNK_HOME/bin/myColdToFrozenScript.py"

# This example freezes buckets on the same schedule, but lets Splunk do the
# freezing process as opposed to a script
[default]
maxWarmDBCount = 200
frozenTimePeriodInSecs = 432000
rotatePeriodInSecs = 30
coldToFrozenDir = "$SPLUNK_HOME/myfrozenarchive"

### This example demonstrates the use of volumes ###

# volume definitions; prefixed with "volume:"
```

```
[volume:hot1]
path = /mnt/fast_disk
maxVolumeDataSizeMB = 100000

[volume:cold1]
path = /mnt/big_disk
# maxVolumeDataSizeMB not specified: no data size limitation on top of the
# existing ones

[volume:cold2]
path = /mnt/big_disk2
maxVolumeDataSizeMB = 1000000

# index definitions

[idx1]
homePath = volume:hot1/idx1
coldPath = volume:cold1/idx1

# thawedPath must be specified, and cannot use volume: syntax
# choose a location convenient for reconstitition from archive goals
# For many sites, this may never be used.
thawedPath = $SPLUNK_DB/idx1/thaweddb

[idx2]
# note that the specific indexes must take care to avoid collisions
homePath = volume:hot1/idx2
coldPath = volume:cold2/idx2
thawedPath = $SPLUNK_DB/idx2/thaweddb

[idx3]
homePath = volume:hot1/idx3
coldPath = volume:cold2/idx3
thawedPath = $SPLUNK_DB/idx3/thaweddb

[idx4]
datatype = metric
homePath = volume:hot1/idx4
coldPath = volume:cold2/idx4
thawedPath = $SPLUNK_DB/idx4/thaweddb
metric.maxHotBuckets = 6
metric.splitByIndexKeys = metric_name

### Indexes may be allocated space in effective groups by sharing volumes  ###

# perhaps we only want to keep 100GB of summary data and other
# low-volume information
[volume:small_indexes]
path = /mnt/splunk_indexes
maxVolumeDataSizeMB = 100000

# and this is our main event series, allowing 50 terabytes
[volume:large_indexes]
path = /mnt/splunk_indexes
maxVolumeDataSizeMB = 50000000

# summary and rare_data together will be limited to 100GB
[summary]
homePath=volume:small_indexes/summary/db
coldPath=volume:small_indexes/summary/colddb
thawedPath=$SPLUNK_DB/summary/thaweddb
# low-volume indexes probably don't want a lot of hot buckets
```

```
maxHotBuckets = 2
# if the volume is quite low, and you have data sunset goals you may
# want to have smaller buckets
maxDataSize = 500


[rare_data]
homePath=volume:small_indexes/rare_data/db
coldPath=volume:small_indexes/rare_data/colddb
thawedPath=$SPLUNK_DB/rare_data/thaweddb
maxHotBuckets = 2

# main, and any other large volume indexes you add sharing large_indexes
# will be together be constrained to 50TB, separately from the 100GB of
# the small_indexes
[main]
homePath=volume:large_indexes/main/db
coldPath=volume:large_indexes/main/colddb
thawedPath=$SPLUNK_DB/main/thaweddb
# large buckets and more hot buckets are desirable for higher volume
# indexes, and ones where the variations in the timestream of events is
# hard to predict.
maxDataSize = auto_high_volume
maxHotBuckets = 10
# Allow the main index up to 8TB of the 50TB volume limit.
homePath.maxDataSizeMB = 8000000


[idx1_large_vol]
homePath=volume:large_indexes/idx1_large_vol/db
coldPath=volume:large_indexes/idx1_large_vol/colddb
thawedPath=$SPLUNK_DB/idx1_large/thaweddb
# this index will exceed the default of .5TB requiring a change to maxTotalDataSizeMB
maxTotalDataSizeMB = 750000
maxDataSize = auto_high_volume
maxHotBuckets = 10
# but the data will only be retained for about 30 days
frozenTimePeriodInSecs = 2592000

### This example demonstrates database size constraining ###

# In this example per-database constraint is combined with volumes.  While a
# central volume setting makes it easy to manage data size across multiple
# indexes, there is a concern that bursts of data in one index may
# significantly displace data from others.  The homePath.maxDataSizeMB setting
# can be used to assure that no index will ever take more than certain size,
# therefore alleviating the concern.

# global settings

# will be inherited by all indexes: no database will exceed 1TB
homePath.maxDataSizeMB = 1000000

# volumes

[volume:caliente]
path = /mnt/fast_disk
maxVolumeDataSizeMB = 100000

[volume:frio]
path = /mnt/big_disk
maxVolumeDataSizeMB = 1000000
```

```
# and this is our main event series, allowing about 50 terabytes
[volume:large_indexes]
path = /mnt/splunk_indexes
maxVolumeDataSizeMB = 50000000

# indexes

[i1]
homePath = volume:caliente/i1
# homePath.maxDataSizeMB is inherited
coldPath = volume:frio/i1
# coldPath.maxDataSizeMB not specified: no limit - old-style behavior

thawedPath = $SPLUNK_DB/i1/thaweddb

[i2]
homePath = volume:caliente/i2
# overrides the default maxDataSize
homePath.maxDataSizeMB = 1000
coldPath = volume:frio/i2
# limits the cold DB's
coldPath.maxDataSizeMB = 10000
thawedPath = $SPLUNK_DB/i2/thaweddb

[i3]
homePath = /old/style/path
homePath.maxDataSizeMB = 1000
coldPath = volume:frio/i3
coldPath.maxDataSizeMB = 10000
thawedPath = $SPLUNK_DB/i3/thaweddb

# main, and any other large volume indexes you add sharing large_indexes
# will together be constrained to 50TB, separately from the rest of
# the indexes
[main]
homePath=volume:large_indexes/main/db
coldPath=volume:large_indexes/main/colddb
thawedPath=$SPLUNK_DB/main/thaweddb
# large buckets and more hot buckets are desirable for higher volume indexes
maxDataSize = auto_high_volume
maxHotBuckets = 10
# Allow main index to override global and use 8TB of the 50TB volume limit.
homePath.maxDataSizeMB = 8000000


### This example demonstrates how to configure a volume that points to
### S3-based remote storage and indexes that use this volume.  The setting
### "storageType=remote" indicates that this is a remote-storage volume.
### The "remotePath" parameter associates the index with that volume
### and configures a top-level location for uploading buckets.

[volume:s3]
storageType = remote
path = s3://remote_volume
remote.s3.bucket_name = example-s3-bucket
remote.s3.access_key = S3_ACCESS_KEY
remote.s3.secret_key = S3_SECRET_KEY

[default]
remotePath = volume:s3/$_index_name
```

```
[i4]
coldPath = $SPLUNK_DB/$_index_name/colddb
homePath = $SPLUNK_DB/$_index_name/db
thawedPath = $SPLUNK_DB/$_index_name/thaweddb

[i5]
coldPath = $SPLUNK_DB/$_index_name/colddb
homePath = $SPLUNK_DB/$_index_name/db
thawedPath = $SPLUNK_DB/$_index_name/thaweddb

### This example demonstrates how to configure a volume that points to
### GCS-based remote storage.
### "storageType=remote" indicates that this is a remote-storage volume.
### The "remotePath" parameter associates the index with that volume
### and configures a top-level location for uploading buckets.

[volume:gs]
storageType = remote
path = gs://test-bucket/some/path
remote.gs.credential_file = credentials.json

[default]
remotePath = volume:gs/$_index_name

[i6]
coldPath = $SPLUNK_DB/$_index_name/colddb
homePath = $SPLUNK_DB/$_index_name/db
thawedPath = $SPLUNK_DB/$_index_name/thaweddb
```

inputs.conf

The following are the spec and example files for `inputs.conf`.

inputs.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains possible settings you can use to configure inputs,
# distributed inputs such as forwarders, and file system monitoring in
# inputs.conf.
#
# Each stanza controls different search commands settings.
#
# There is an inputs.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name inputs.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see inputs.conf.example.
#
```

```
# You must restart the Splunk platform instance to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file wins.
#   * If an setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

##############################################################################
# GENERAL SETTINGS:
# The following settings are valid for all input types (except file system
# change monitor, which is described in a separate section in this file).
# You must first enter a stanza header in square brackets, specifying the input
# type. See later in this file for examples. Then, use any of the
# following settings.
#
# To specify global settings for Windows Event Log inputs, place them in
# the [WinEventLog] global stanza as well as the [default] stanza.
##############################################################################

host = <string>
* Sets the host key/field to a static value for this input stanza.
* The input uses this field during parsing and indexing. It also uses this
  field at search time.
* As a convenience, the input prepends the chosen string with 'host::'.
* When set to '$decideOnStartup', sets the field to the hostname of executing
  machine. The hostname is checked and the field set at every splunkd startup.
* If you run multiple instances of the software on the same machine (hardware
  or virtual machine), choose unique values for 'host' to differentiate
  your data, ex. myhost-sh-1 or myhost-idx-2.
* Do not put the <string> value in quotes. Use host=foo, not host="foo".
* When 'host' is set to "$decideOnStartup", you can further control how splunkd
  derives the hostname by using the 'hostnameOption' setting in server.conf.
  * For example, if you want splunkd to use the fully qualified domain
    name for the machine, set "host = $decideOnStartup" in inputs.conf and
    "hostnameOption = fullyqualifiedname" in server.conf.
  * More information on hostname options can be found in the server.conf
    specification file.
* If you remove the 'host' setting from $SPLUNK_HOME/etc/system/local/inputs.conf
  or remove $SPLUNK_HOME/etc/system/local/inputs.conf, the setting reverts to
  "$decideOnStartup". Apps that need a resolved host value should use the
  'host_resolved' property in the response for the REST 'GET' call of the
  input source. This property is set to the hostname of the local Splunk
  instance. It is a read only  property that is not written to inputs.conf.
* Default: "$decideOnStartup"

run_only_one= <boolean>
* Determines if a scripted or modular inputs runs on one search head
```

```
  in SHC.
* Currently not supported. This setting is related to a feature that is
  still under development.
* Default: true

index = <string>
* Sets the index to store events from this input.
* Primarily used to specify the index to store events that come in through
  this input stanza.
* Default: main (or whatever you have set as your default index)

source = <string>
* Sets the source key/field for events from this input.
* Detail: Sets the source key initial value. The key is used during
  parsing/indexing, in particular to set the source field during
  indexing. It is also the source field used at search time.
* As a convenience, the chosen string is prepended with 'source::'.
* Avoid overriding the source key. The input layer provides a more accurate
  string to aid in problem analysis and investigation, recording the file
  from which the data was retrieved. Consider using source types, tagging,
  and search wildcards before overriding this value.
* Do not put the <string> value in quotes: Use source=foo,
  not source="foo".
* Default: the input file path

sourcetype = <string>
* Sets the sourcetype key/field for events from this input.
* Explicitly declares the source type for this input instead of letting
  it be determined through automated methods. This is important for
  search and for applying the relevant configuration for this data type
  during parsing and indexing.
* Sets the sourcetype key initial value. The key is used during
  parsing or indexing to set the source type field during
  indexing. It is also the source type field used at search time.
* As a convenience, the chosen string is prepended with 'sourcetype::'.
* Do not put the <string> value in quotes: Use sourcetype=foo,
  not sourcetype="foo".
* If not set, the indexer analyzes the data and chooses a source type.
* No default.

queue = [parsingQueue|indexQueue]
* Sets the queue where the input processor deposits the events it reads.
* Set to "parsingQueue" to apply the props.conf file and other parsing rules to
  your data. For more information about the props.conf file and rules
  timestamps and linebreaks, see the props.conf file and the
  online documentation at http://docs.splunk.com/Documentation.
* Set to "indexQueue" to send your data directly into the index.
* Default: parsingQueue

# Pipeline Key defaulting.

* Pipeline keys in general can be defaulted in inputs stanzas.
* The list of user-available, modifiable pipeline keys is described in
  transforms.conf.spec. See transforms.conf.spec for further information on
  these keys.
* The currently-defined keys which are available literally in inputs stanzas
  are as follows:

queue = <value>
_raw = <value>
_meta = <value>
_time = <value>
```

```
* Inputs have special support for mapping host, source, sourcetype, and index
  to their metadata names such as host -> Metadata:Host
* Defaulting these values is not recommended, and is
  generally only useful as a workaround to other product issues.
* Defaulting these keys in most cases overrides the default behavior of
  input processors, but this behavior is not guaranteed in all cases.
* Values defaulted here, as with all values provided by inputs, can be
  altered by transforms at parse time.

############################################################################
# This section contains options for routing data using inputs.conf rather than
# outputs.conf.
#
# NOTE: Concerning routing via inputs.conf:
# This is a simplified set of routing options you can use as data comes in.
# For more flexible options or details on configuring required or optional
# settings, see outputs.conf.spec.
############################################################################

_TCP_ROUTING = <comma-separated list>
* A comma-separated list of tcpout group names.
* This setting lets you selectively forward data to one or more specific indexers.
* Specify the tcpout group that the forwarder uses when forwarding the data.
  The tcpout group names are defined in outputs.conf with
  [tcpout:<tcpout_group_name>].
* To forward data to all tcpout group names that have been defined in
  outputs.conf, set to '*' (asterisk).
* To forward data from the "_internal" index, you must explicitly set
  '_TCP_ROUTING' to either "*" or a specific splunktcp target group.
* Default: The groups specified in 'defaultGroup' in [tcpout] stanza in
  the outputs.conf file

_SYSLOG_ROUTING = <comma-separated list>
* A comma-separated list of syslog group names.
* Use this setting to selectively forward the data to specific destinations as
  syslog events.
* Specify the syslog group to use when forwarding the data.
  The syslog group names are defined in outputs.conf with
  [syslog:<syslog_group_name>].
* The destination host must be configured in outputs.conf, using
  "server=[<ip>|<servername>]:<port>".
* This setting does not work on a universal forwarder.
* Default: The groups specified in 'defaultGroup' in the [syslog] stanza in
  the outputs.conf file

_INDEX_AND_FORWARD_ROUTING = <string>
* If set for any input stanza, causes all data coming from that input
  stanza to be labeled with this setting.
* When 'selectiveIndexing' is in use on a forwarder:
  * Data without this label will not be indexed by that forwarder.
  * Data with this label will be indexed in addition to any forwarding.
* This setting does not actually cause data to be forwarded or not forwarded in
  any way, nor does it control where the data is forwarded in multiple-forward
  path cases.
* Only has effect if you use the 'selectiveIndexing' feature in outputs.conf.
* Default: not set
```

Deny list

```
[blacklist:<path>]
* Protects files on the file system from being indexed or previewed.
* The input treats a file as denied if the file starts with any of the
  defined deny list <paths>.
* Adding a file to the deny list with the specified path occurs even if a monitor
  stanza defines an allow list that matches the file path.
* The preview endpoint returns an error when asked to preview an
  excluded file.
* The oneshot endpoint and command also returns an error.
* When a denied file is monitored, using monitor:// or batch://,
  the 'filestatus' endpoint shows an error.
* For fschange with the 'sendFullEvent' option enabled, contents of
  denied files are not indexed.
```

Input types

```
Valid input stanzas, along with their input-specific settings, follow:
```

MONITOR:

```
[monitor://<path>]
* Configures a file monitor input to watch all files in the <path> you specify.
* <path> can be an entire directory or a single file.
* You must specify the input type and then the path, so put three slashes in
  your path if you are starting at the root on *nix systems (to include the
  slash that indicates an absolute path).

# Additional settings:

host_regex = <regular expression>
* If specified, <regular expression> extracts host from the path to the file
  for each input file.
    * Detail: This feature examines the source key; if source is set
      explicitly in the stanza, that string is matched, not the original
      filename.
* Specifically, the first group of the regular expression (regex) is used
  as the host.
* If the regex fails to match, the input uses the default 'host' setting.
* If 'host_regex' and 'host_segment' are both set, the input ignores 'host_regex'.
* No default.

host_segment = <integer>
* If set to N, the Splunk platform sets the Nth "/"-separated segment of the path
  as 'host'.
    * For example, if you set "host_segment = 3" and the path is
      /logs/servers/host08/abc.txt, the third segment, "host08", is used.
* If the value is not an integer or is less than 1, the default 'host'
  setting is used.
* On Windows machines, the drive letter and colon before the backslash *does not*
  count as one segment.
    * For example, if you set "host_segment = 3" and the monitor path is
      D:\logs\servers\host01, Splunk software sets the host as "host01" because
```

```
      that is the third segment.
* No default.

whitelist = <regular expression>
* If set, files from this input are monitored only if their path matches the
  specified regular expression.
* Takes precedence over the deprecated '_whitelist' setting, which functions
  the same way.
* No default.

blacklist = <regular expression>
* If set, files from this input are NOT monitored if their path matches the
  specified regex.
* Takes precedence over the deprecated '_blacklist' setting, which functions
  the same way.
* If a file matches the regexes in both the deny list and allow list settings,
  the file is NOT monitored. Deny lists take precedence over allow lists.
* No default.

NOTE concerning wildcards and monitor:
* You can use wildcards to specify your input path for monitored inputs. Use
  "..." for recursive directory matching and "*" for wildcard matching in a
  single directory segment.
* "..." searches recursively through one or more directories. This means that
  /foo/.../bar matches foo/1/bar, foo/1/2/bar, etc.
* You can use multiple "..." specifications in a single input path. For
  example: /foo/.../bar/...
* The asterisk (*) matches anything in a single path segment; unlike "...", it
  does not search recursively. For example, /foo/*/bar matches the files
  /foo/1/bar, /foo/2/bar, etc. However, it does not match
  /foo/bar or /foo/1/2/bar.
  A second example: /foo/m*r/bar matches /foo/mr/bar, /foo/mir/bar,
  /foo/moor/bar, etc. It does not match /foo/mi/or/bar.
* You can combine "*" and "..." as needed: foo/.../bar/* matches any file in
  the bar directory within the specified path.
* A monitor stanza path will interpret regular expression metacharacters as
  strings unless they are preceded by the wildcard values "*" or "..." in a prior
  segment of the path.
* In the case where multiple unique monitor inputs overlap through the use of
  wildcards or specific paths defined in the monitor stanza, the Splunk platform
  processes the files using the monitor stanza that is the closest
  specific match.

crcSalt = <string>
* Use this setting to force the input to consume files that have matching CRCs,
  or cyclic redundancy checks.
    * By default, the input only performs CRC checks against the first 256
      bytes of a file. This behavior prevents the input from indexing the same
      file twice, even though you might have renamed it, as with rolling log
      files, for example. Because the CRC is based on only the first
      few lines of the file, it is possible for legitimately different files
      to have matching CRCs, particularly if they have identical headers.
* If set, <string> is added to the CRC.
* If set to the literal string "<SOURCE>" (including the angle brackets), the
  full directory path to the source file is added to the CRC. This ensures
  that each file being monitored has a unique CRC. When 'crcSalt' is invoked,
  it is usually set to <SOURCE>.
* Be cautious about using this setting with rolling log files; it could lead
  to the log file being re-indexed after it has rolled.
* In many situations, 'initCrcLength' can be used to achieve the same goals.
* Default: empty string
```

```
initCrcLength = <integer>
* How much of a file, in bytes, that the input reads before trying to
  identify whether it has already seen the file.
* You might want to adjust this if you have many files with common
  headers (comment headers, long CSV headers, etc) and recurring filenames.
* Cannot be less than 256 or more than 1048576.
* CAUTION: Improper use of this setting causes data to be re-indexed. You
  might want to consult with Splunk Support before adjusting this value - the
  default is fine for most installations.
* Default: 256 (bytes)

ignoreOlderThan = <non-negative integer>[s|m|h|d]
* The monitor input compares the modification time on files it encounters
  with the current time. If the time elapsed since the modification time
  is greater than the value in this setting, Splunk software puts the file
  on the ignore list.
* Files on the ignore list are not checked again until the Splunk
  platform restarts, or the file monitoring subsystem is reconfigured. This
  is true even if the file becomes newer again at a later time.
  * Reconfigurations occur when changes are made to monitor or batch
    inputs through Splunk Web or the command line.
* Use 'ignoreOlderThan' to increase file monitoring performance when
  monitoring a directory hierarchy that contains many older, unchanging
  files, and when removing or adding a file to the deny list from the
  monitoring location is not a reasonable option.
* Do NOT select a time that files you want to read could reach in
  age, even temporarily. Take potential downtime into consideration!
  * Suggested value: 14d, which means 2 weeks
  * For example, a time window in significant numbers of days or small
    numbers of weeks are probably reasonable choices.
  * If you need a time window in small numbers of days or hours,
    there are other approaches to consider for performant monitoring
    beyond the scope of this setting.
* NOTE: Most modern Windows file access APIs do not update file
  modification time while the file is open and being actively written to.
  Windows delays updating modification time until the file is closed.
  Therefore you might have to choose a larger time window on Windows
  hosts where files may be open for long time periods.
* Value must be: <number><unit>. For example, "7d" indicates one week.
* Valid units are "d" (days), "h" (hours), "m" (minutes), and "s"
  (seconds).
* No default, meaning there is no threshold and no files are
  ignored for modification time reasons

followTail = <boolean>
* Whether or not the input should skip past current data in a monitored file
  for a given input stanza.
* This setting lets you skip over data in files, and immediately begin indexing
  current data.
* If you set to "1", monitoring starts at the end of the file (like
  *nix 'tail -f'). The input does not read any data that exists in
  the file when it is first encountered. The input only reads data that
  arrives after the first encounter time.
* If you set to "0", monitoring starts at the beginning of the file.
* This is an advanced setting. Contact Splunk Support before using it.
* Best practice for using this setting:
  * Enable this setting and start the Splunk instance.
  * Wait enough time for the input to identify the related files.
  * Disable the setting and restart the instance.
* Do not leave 'followTail' enabled in an ongoing fashion.
* Do not use 'followTail' for rolling log files (log files that get renamed as
  they age) or files whose names or paths vary.
```

```
* Default: 0

alwaysOpenFile = <boolean>
* Whether or not an input opens a file to check whether it has already
  been indexed, by skipping the modification time and size checks.
* Only useful for files that do not update modification time or size.
* Only known to be needed when monitoring files on Windows, mostly for
  Internet Information Server logs.
* Configuring this setting to "1" can increase load and slow indexing. Use it
  only as a last resort.
* Default: 0

time_before_close = <integer>
* The amount of time, in seconds, that the file monitor must wait for
  modifications before closing a file after reaching an End-of-File
  (EOF) marker.
* Tells the input not to close files that have been updated in the
  past 'time_before_close' seconds.
* Default: 3

multiline_event_extra_waittime = <boolean>
* Whether or not the file monitor input delays sending an event delimiter when
  it reads a file with multiple-line events, to account for the time it
  sometimes takes for lines of those events to arrive.
* By default, the file monitor sends an event delimiter when:
  * It reaches EOF of a file it monitors and
  * The last character it reads is a newline.
* In some cases, it takes time for all lines of a multiple-line event to
  arrive.
* Set to "true" to delay sending an event delimiter until the time that the
  file monitor closes the file, as defined by the 'time_before_close' setting,
  to allow all event lines to arrive.
* Default: false

recursive = <boolean>
* Whether or not the input monitors subdirectories that it finds within a
  monitored directory.
* A value of "true" means the input monitors sub-directories.
* A value of "false" means the input does not monitor sub-directories.
* Default: true

followSymlink = <boolean>
* Whether or not the input follows any symbolic links within a monitored directory.
* A value of "true" means the input follows symbolic links
  and monitors files at the symbolic link destination.
* Additionally, any allow lists or deny lists that the input stanza defines
  also apply to files at the symbolic link destination.
* A value of "false" means the input ignores symbolic links
  that it finds within a monitored directory.
* Default: true

_whitelist = ...
* DEPRECATED.
* This setting is valid unless the 'whitelist' setting also exists.

_blacklist = ...
* DEPRECATED.
* This setting is valid unless the 'blacklist' setting also exists.
```

BATCH ("Upload a file" in Splunk Web):

```
Use the 'batch' input for large archives of historic data. If you
want to continuously monitor a directory or index small archives, use 'monitor'
(see the MONITOR section). 'batch' reads in the file and indexes it, and then
deletes the file on disk.

[batch://<path>]
* A one-time, destructive input of files in <path>.
* This stanza must include the 'move_policy = sinkhole' setting.
* This input reads and indexes the files, then DELETES THEM IMMEDIATELY.
* For continuous, non-destructive inputs of files, use 'monitor' instead.

# Additional settings:

move_policy = sinkhole
* This setting is required. You *must* include "move_policy = sinkhole"
  when you define batch inputs.
* This setting causes the input to load the file destructively.
* CAUTION: Do not use the 'batch' input type for files you do not want to
  delete after indexing.
* The 'move_policy' setting exists for historical reasons, but remains as a
  safeguard. As an administrator, you must explicitly declare
  that you want the data in the monitored directory (and its sub-directories) to
  be deleted after being read and indexed.

host_regex = see the definition in the MONITOR section.
host_segment = see the definition in the MONITOR section.
crcSalt = see the definition in the MONITOR section.
time_before_close = see the definition in the MONITOR section.

log_on_completion = <boolean>
* Whether or not the Splunk platform writes an entry into the
  splunkd.log file when it indexes files with this input.
* When set to "false", this setting prevents the Splunk platform from
  writing to splunkd.log when it indexes files with this input.
* Default: true

# 'batch' inputs do not use the following setting:
# source = <string>

followSymlink = <boolean>
* Works similarly to the same setting for monitor, but does not delete files
  after following a symbolic link out of the monitored directory.

# The following settings work identically as for [monitor::] stanzas,
# documented previously
host_regex = <regular expression>
host_segment = <integer>
crcSalt = <string>
recursive = <boolean>
whitelist = <regular expression>
blacklist = <regular expression>
initCrcLength = <integer>
time_before_close = <integer>
```

TCP: Transport Control Protocol (TCP) network inputs

```
[tcp://<remote server>:<port>]
* Configures the input to listen on a specific TCP network port.
* If a <remote server> makes a connection to this instance, the input uses this
  stanza to configure itself.
* If you do not specify <remote server>, this stanza matches all connections
  on the specified network port.
* Generates events with source set to "tcp:<port>", for example: tcp:514
* If you do not specify a sourcetype, the input generates events with sourcetype
  set to "tcp-raw".

# Additional settings:

connection_host = [ip|dns|none]
* How the network input sets the host field for the events it generates.
* A value of "ip" sets the host to the IP address of the system sending the data.
* A value of "dns" sets the host to the reverse DNS entry for the IP address of
  the system that sends the data. For this to work correctly, set the forward
  DNS lookup to match the reverse DNS lookup in your DNS configuration.
* A value of "none" leaves the host as specified in inputs.conf, typically the
  hostname of the system running Splunk software.
* Default: dns

queueSize = <integer>[KB|MB|GB]
* The maximum size of the in-memory input queue.
* Default: 500KB

persistentQueueSize = <integer>[KB|MB|GB|TB]
* The maximum size of the persistent queue file.
* Persistent queues can help prevent loss of transient data. For information on
  persistent queues and how the 'queueSize' and 'persistentQueueSize' settings
  interact, search the online documentation for "persistent queues".
* If you set this to a value other than 0, then 'persistentQueueSize' must
  be larger than either the in-memory queue size (as defined by the 'queueSize'
  setting in inputs.conf or 'maxSize' settings in [queue] stanzas in
  server.conf).
* Default: 0 (no persistent queue)

requireHeader = <boolean>
* Whether or not to require a header be present at the beginning of every
  stream.
* This header can be used to override indexing settings.
* Default: false

listenOnIPv6 = [no|yes|only]
* Whether or not the input listens on IPv4, IPv6, or both protocols.
* Set to "yes" to listen on both IPv4 and IPv6 protocols.
* Set to "only" to listen on only the IPv6 protocol.
* Default: The setting in the [general] stanza of the server.conf file

acceptFrom = <comma- or space-separated list>
* A list of TCP networks or addresses to accept connections from.
* Use commas or spaces to separate multiple network rules.
* The accepted formats for network and address rules are:
    1. A single IPv4 or IPv6 address (examples: "192.0.2.3", "2001:db8::2:1")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
       (examples: "192.0.2/24", "2001:DB8::/32")
    3. A DNS name. Use "*" as a wildcard.
       (examples: "myhost.example.com", "*.example.org")
```

```
    4. The wildcard "*" matches anything.
* A prefix of '!' for an entry sets a rule to deny and reject connections. The ACL
 applies rules in order, and uses the first matching rule. For example,
 the rules "!192.0.2/24, *" prevents connections from the 192.0.2/24
 network, but accepts all other connections.
* Default: * (accept from anywhere)

rawTcpDoneTimeout = <seconds>
* The amount of time, in seconds, that a network connection can remain idle
  before Splunk software declares that the last event over that connection
  has been received.
* If a connection over this port remains idle for more than
  'rawTcpDoneTimeout' seconds after receiving data, it adds a Done-key. This
  declares that the last event has been completely received.
* Default: 10

[tcp:<port>]
* Configures the input listen on the specified TCP network port.
* This stanza is similar to [tcp://<remote server>:<port>], but listens for
  connections to the specified port from any host.
* Generates events with a source of tcp:<port>.
* If you do not specify a sourcetype, generates events with a source type of
  tcp-raw.
* This stanza supports the following settings:

connection_host = [ip|dns|none]
queueSize = <integer>[KB|MB|GB]
persistentQueueSize = <integer>[KB|MB|GB|TB]
requireHeader = <boolean>
listenOnIPv6 = [no|yes|only]
acceptFrom = <comma- or space-separated list>
rawTcpDoneTimeout = <integer>
```

Data distribution:

```
# Global settings for splunktcp. Used on the receiving side for data forwarded
# from a forwarder.

[splunktcp]
route = [has_key|absent_key:<key>:<queueName>;...]
* Settings for the light forwarder.
* The receiver sets these parameters automatically -- you do not need to set
  them yourself.
* The property route is composed of rules delimited by ';' (semicolon).
* The receiver checks each incoming data payload through the cooked TCP port
  against the route rules.
* If a matching rule is found, the receiver sends the payload to the specified
  <queueName>.
* If no matching rule is found, the receiver sends the payload to the default
  queue specified by any queue= for this stanza. If no queue= key is set in
  the stanza or globally, the receiver sends the events to the parsingQueue.

enableS2SHeartbeat = <boolean>
* Specifies the global keepalive setting for all splunktcp ports.
* This option is used to detect forwarders which might have become unavailable
  due to network, firewall, or other problems.
* The receiver monitors each connection for presence of a heartbeat, and if the
  heartbeat is not seen for 's2sHeartbeatTimeout' seconds, it closes the
  connection.
```

```
* Default: true (heartbeat monitoring enabled)

s2sHeartbeatTimeout = <integer>
* The amount of time, in seconds, that a receiver waits for heartbeats from
  forwarders that connect to this instance.
* The receiver closes a forwarder connection if it does not receive
  a heartbeat for 's2sHeartbeatTimeout' seconds.
* Default: 600 (10 minutes)

inputShutdownTimeout = <integer>
* The amount of time, in seconds, that a receiver waits before shutting down
  inbound TCP connections after it receives a signal to shut down.
* Used during shutdown to minimize data loss when forwarders are connected to a
  receiver.
* During shutdown, the TCP input processor waits for 'inputShutdownTimeout'
  seconds and then closes any remaining open connections.
* If all connections close before the end of the timeout period,
  shutdown proceeds immediately, without waiting for the timeout.

stopAcceptorAfterQBlock = <integer>
* The amount of time, in seconds, to wait before closing the splunktcp port.
* If the receiver is unable to insert received data into the configured queue
  for more than the specified number of seconds, it closes the splunktcp port.
* This action prevents forwarders from establishing new connections to this
  receiver.
* Forwarders that have an existing connection will notice the port is closed
  upon test-connections and move to other receivers.
* After the queue unblocks, and the TCP input can continue processing data, the
  receiver starts listening on the port again.
* This setting should not be adjusted lightly as extreme values can interact
  poorly with other defaults.
* NOTE: If there are multiple tcp/splunktcp listener ports configured,
  all listening ports will be shut down regardless of whether other queues are
  blocked or not.
* Default: 300 (5 minutes)

listenOnIPv6 = no|yes|only
* See the description for this setting in the [tcp://<remote server>:<port>]
  stanza.

acceptFrom = <comma- or space-separated list>
* See the description for this setting in the [tcp://<remote server>:<port>]
  stanza.

negotiateProtocolLevel = <unsigned integer>
* If set, lets forwarders that connect to this receiver (or specific port)
  send data using only up to the specified feature level of the Splunk
  forwarder protocol.
* If set to a value that is lower than the default, denies the use
  of newer forwarder protocol features during connection negotiation. This
  might impact indexer efficiency.
* Default (if 'negotiateNewProtocol' is "true"): 1
* Default (if 'negotiateNewProtocol' is not "true"): 0

negotiateNewProtocol = <boolean>
* DEPRECATED.
* Use the 'negotiateProtocolLevel' setting instead.
* Controls the default configuration of the 'negotiateProtocolLevel' setting.
* Default: true

concurrentChannelLimit = <unsigned integer>
* The number of unique channel codes that are available for forwarders to
```

```
  use to communicate with an indexer.
* Each forwarder that connects to this indexer may use up to
  'concurrentChannelLimit' unique channel codes.
* In other words, each forwarder may have up to 'concurrentChannelLimit'
  channels in flight concurrently.
* The receiver closes a forwarder connection if a forwarder attempts to
  exceed this value.
* This setting only applies when the new forwarder protocol is in use.
* Default: 300

logRetireOldS2S = <boolean>
* Whether or not the Splunk platform logs the usage of old versions of Splunk-to-Splunk (S2S)
  protocol.
* The old S2S protocol retirement logs provide visibility into customers' usage
  of the old S2S protocol version V3 which is less performant than the current version V4.
* A value of "true" means that splunkd generates warning logs for the old S2S protocol
  versions.
* See the 'logRetireOldS2SRepeatFrequency' setting for additional constraints on
  when the Splunk platform logs the use of old S2S protocol versions.
* Default: true


logRetireOldS2SMaxCache = <unsigned integer>
* The size of the cache for tracking forwarders that use old S2S protocols.
* The cache keeps track of unique forwarders that use the old S2S protocol. When a
  forwarder is in the cache, the Splunk platform doesn't log usage of the old protocol
  for that forwarder for a time period of 'logRetireOldS2SRepeatFrequency', to avoid generating
  duplicate logs.
* If the cache fills before the 'logRetireOldS2SRepeatFrequency' period elapses,
  the Splunk platform removes the forwarder that has been in the cache the longest
  from the cache to make space.
* Update this setting as per the number of forwarders that currently use the old S2S
  protocol to send data to indexers. If the number of forwarders that use
  old S2S protocols is larger than the cache size, some forwarders might generate duplicate
  logs even though the previous log was within the 'logRetireOldS2SRepeatFrequency'
  period.
* When you restart Splunk Enterprise, the cache resets and the timer starts over.
* This setting takes effect only when 'logRetireOldS2S' has a value of "true".
* Default: 10000

logRetireOldS2SRepeatFrequency = <timespan>
* The interval between writing repeat entries into the retire old S2S warning log
  for a certain forwarder.
* This setting helps reduce retire old S2S log size by providing control over how
  often to log.
* When a forwarder uses the old S2S protocol version to communicate with splunkd, splunkd
  adds the forwarder to a cache. Subsequent communication with the same
  forwarder won't generate a new entry to the log until a period of
  'logRetireOldS2SRepeatFrequency' has elapsed. Splunkd then resets the log timestamp and
  writes another "retire old S2S protocol" warning log entry.
* The Splunk platform enforces this setting as long as the size of the cache
  does not exceed 'logRetireOldS2SMaxCache' entries. When there are more than
  'logRetireOldS2SMaxCache' entries, the cache removes the entry with the oldest
  access time to make space.
* When you restart Splunk Enterprise, the cache resets and the timer starts over.
* This setting takes effect only when 'logRetireOldS2S' has a value of "true".
* A value of "0" means that the platform logs old S2S protocol warning entries every time
  it receives a communication using the old S2S protocol version.
* Default: 1d

# Forwarder-specific settings for splunktcp.
```

```
[splunktcp://[<remote server>]:<port>]
* Receivers use this input stanza.
* This is the same as the [tcp://] stanza, except the remote server is assumed
  to be a Splunk instance, most likely a forwarder.
* <remote server> is optional. If you specify it, the receiver listens only for
  data from <remote server>.
  * Use of <remote server> is not recommended. Use the 'acceptFrom' setting,
    which supersedes this setting.

connection_host = [ip|dns|none]
* For splunktcp, the 'host' or 'connection_host' is be used if the remote
  Splunk instance does not set a host, or if the host is set to
  "<host>::<localhost>".
* "ip" sets the host to the IP address of the system sending the data.
* "dns" sets the host to the reverse DNS entry for IP address of the system
  that sends the data. For this to work correctly, set the forward DNS lookup
  to match the reverse DNS lookup in your DNS configuration.
* "none" leaves the host as specified in inputs.conf, typically the Splunk
  system hostname.
* Default: ip

compressed = <boolean>
* Whether or not the receiver communicates with the forwarder in
  compressed format.
* Applies to non-Secure Sockets Layer (SSL) receiving only. There is no
  compression setting required for SSL.
* A value of "true" means the receiver communicates with the forwarder in
  compressed format.
* If set to "true", there is no longer a requirement to also set
  "compressed = true" in the outputs.conf file on the forwarder.
* Default: false

enableS2SHeartbeat = <boolean>
* Specifies the keepalive setting for the splunktcp port.
* This option is used to detect forwarders which might have become unavailable
  due to network, firewall, or other problems.
* The receiver monitors the connection for presence of a heartbeat, and if it
  does not see the heartbeat in 's2sHeartbeatTimeout' seconds, it closes the
  connection.
* This overrides the default value specified at the global [splunktcp] stanza.
* Default: true (heartbeat monitoring enabled)

s2sHeartbeatTimeout = <integer>
* The amount of time, in seconds, that a receiver waits for heartbeats from
  forwarders that connect to this instance.
* The receiver closes the forwarder connection if it does not see a heartbeat
  for 's2sHeartbeatTimeout' seconds.
* This overrides the default value specified at the global [splunktcp] stanza.
* Default: 600 (10 minutes)

queueSize = <integer>[KB|MB|GB]
* The maximum size of the in-memory input queue.
* Default: 500KB

negotiateProtocolLevel = <unsigned integer>
* See the description for this setting in the [splunktcp] stanza.

negotiateNewProtocol = <boolean>
* See the description for this setting in the [splunktcp] stanza.

concurrentChannelLimit = <unsigned integer>
* See the description for this setting in the [splunktcp] stanza.
```

```
[splunktcp:<port>]
* This input stanza is the same as [splunktcp://[<remote server>]:<port>], but
  the input accepts connections from any server.
* See the online documentation for [splunktcp://[<remote server>]:<port>] for
  more information on the following supported settings:

connection_host = [ip|dns|none]
compressed = <boolean>
enableS2SHeartbeat = <boolean>
s2sHeartbeatTimeout = <integer>
queueSize = <integer>[KB|MB|GB]
negotiateProtocolLevel = <unsigned integer>
negotiateNewProtocol = <boolean>
concurrentChannelLimit = <unsigned integer>

# Access control settings.
[splunktcptoken://<token name>]
* Use this stanza to specify forwarders from which to accept data.
* You must configure a token on the receiver, then configure the same
  token on forwarders.
* The receiver discards data from forwarders that do not have the
  token configured.
* This setting is enabled for all receiving ports.
* This setting is optional.
* NOTE: When specifying a <token name>, you must use a specific format,
  as follows: NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN. Failure to use this
  format results in the token being ignored.
  * For example, 'A843001F-B2B5-4F94-847D-D07802685BB2'

token = <string>
* The value of the token.
* Must be in the format NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN. Failure to
  use this string format results in the token being ignored.

# SSL settings for data distribution:

[splunktcp-ssl:<port>]
* Use this stanza type if you are receiving encrypted, parsed data from a
  forwarder.
* Set <port> to the port on which the forwarder sends the encrypted data.
* Forwarder settings are set in outputs.conf on the forwarder.
* Compression for SSL is enabled by default. On the forwarder you can still
  specify compression with the 'useClientSSLCompression' setting in
  outputs.conf.
* The 'compressed' setting is used for non-SSL connections. However, if you
  still specify 'compressed' for SSL, ensure that the 'compressed' setting is
  the same as on the forwarder, as splunktcp protocol expects the same
  'compressed' setting from forwarders.

connection_host = [ip|dns|none]
* See the description for this setting in the [splunktcp:<port>] stanza.
* Default: ip

compressed = <boolean>
* See the description for this setting in the [splunktcp:<port>] stanza.

enableS2SHeartbeat = <boolean>
* See the description for this setting in the [splunktcp:<port>] stanza.

s2sHeartbeatTimeout = <seconds>
* See the description for this setting in the [splunktcp:<port>] stanza.
```

```
listenOnIPv6 = [no|yes|only]
* Select whether this receiver listens on IPv4, IPv6, or both protocols.
* Set to "yes" to listen on both IPv4 and IPv6 protocols.
* Set to "only" to listen on only the IPv6 protocol.
* Default: The setting in the [general] stanza of the server.conf file

acceptFrom = <comma- or space-separated list>
* See the description for this setting in the [tcp://<remote server>:<port>]
  stanza.

negotiateProtocolLevel = <unsigned integer>
* See the description for this setting in the [splunktcp] stanza.

negotiateNewProtocol = <boolean>
* See the description for this setting in the [splunktcp] stanza.

concurrentChannelLimit = <unsigned integer>
* See the description for this setting in the [splunktcp] stanza.

# To specify global ssl settings, that are applicable for all ports, add the
# settings to the SSL stanza.
# Specify any ssl setting that deviates from the global setting here.
# For a detailed description of each ssl setting, refer to the [SSL] stanza.

serverCert = <string>
sslPassword = <string>
requireClientCert = <boolean>
sslVersions = <string>
cipherSuite = <cipher suite string>
ecdhCurves = <comma separated list of ec curves>
dhFile = <string>
allowSslRenegotiation = <boolean>
sslQuietShutdown = <boolean>
sslCommonNameToCheck = <commonName1>, <commonName2>, ...
sslAltNameToCheck = <alternateName1>, <alternateName2>, ...
useSSLCompression = <boolean>

[tcp-ssl:<port>]
* Use this stanza type if you are receiving encrypted, unparsed data from a
  forwarder or third-party system.
* Set <port> to the port on which the forwarder/third-party system is sending
  unparsed, encrypted data.
* To create multiple SSL inputs, you can add the following attributes to each
  [tcp-ssl:<port>] input stanza. If you do not configure a certificate in the
  port, the certificate information is pulled from the default [SSL] stanza:
  * serverCert = <path_to_cert>
  * sslRootCAPath = <path_to_cert> Only add this setting if you
    have not configured the 'sslRootCAPath' setting in server.conf.
  * sslPassword = <string>

listenOnIPv6 = [no|yes|only]
* Select whether the receiver listens on IPv4, IPv6, or both protocols.
* Set to "yes" to listen on both IPv4 and IPv6 protocols.
* Set to "only" to listen on only the IPv6 protocol.
* If not present, the receiver uses the setting in the [general] stanza
  of server.conf.

acceptFrom = <comma- or space-separated list>
* See the description for this setting in the [tcp://<remote server>:<port>]
  stanza.
* Default: "*" (accept from anywhere)
```

```
# To specify global SSL settings, that are applicable for all ports, add the
# settings to the SSL stanza.
# Specify any SSL setting that deviates from the global setting here.
# For a detailed description of each ssl setting, refer to the [SSL] stanza.

serverCert = <string>
sslPassword = <string>
requireClientCert = <boolean>
sslVersions = <string>
cipherSuite = <cipher suite string>
ecdhCurves = <comma separated list of ec curves>
dhFile = <string>
allowSslRenegotiation = <boolean>
sslQuietShutdown = <boolean>
sslCommonNameToCheck = <commonName1>, <commonName2>, ...
sslAltNameToCheck = <alternateName1>, <alternateName2>, ...
useSSLCompression = <boolean>
```

SSL:

```
[SSL]
* Set the global specifications for receiving Secure Sockets Layer (SSL)
 communication underneath this stanza name.

serverCert = <string>
* The full path to the server certificate file.
* This file must be a Privacy-Enhanced Mail (PEM) format file.
* PEM is the most common text-based storage format for SSL certificate files.
* No default.

sslPassword = <string>
* The server certificate password, if it exists.
* Set this to a plain-text password initially.
* Upon first use, the input encrypts and rewrites the password to
  $SPLUNK_HOME/etc/system/local/inputs.conf.

password = <string>
* DEPRECATED.
* Do not use this setting. Use the 'sslPassword' setting instead.

rootCA = <string>
* DEPRECATED.
* Do not use this setting. Use 'server.conf/[sslConfig]/sslRootCAPath' instead.
* Used only if 'sslRootCAPath' is not set.
* The path must refer to a PEM format file that contains one or more root CA
  certificates that have been concatenated together.

requireClientCert = <boolean>
* Whether or not a client must present an SSL certificate to authenticate.
* A value of "true" means that clients must present a certificate to authenticate.
* Default (if using self-signed and third-party certificates): false
* Default (if using the default certificates; overrides the existing
  "false" setting): true

sslVersions = <comma-separated list>
* A list of SSL versions to support.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2"
* The special version "*" selects all supported versions. The version "tls"
  selects all versions that begin with "tls".
```

```
* To remove a version from the list, prefix it with "-".
* SSLv2 is always disabled. Specifying "-ssl2" in the version list has
  no effect.
* When configured in Federal Information Processing Standard (FIPS) mode, the
  "ssl3" version is always disabled, regardless of this configuration.
* The default can vary. See the 'sslVersions' setting in
  $SPLUNK_HOME/etc/system/default/inputs.conf for the current default.

supportSSLV3Only = <boolean>
* DEPRECATED.
* SSLv2 is now always disabled.
* Use the 'sslVersions' setting to set the list of supported SSL versions.

cipherSuite = <string>
* If set, uses the specified cipher string for the input processors.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* The default can vary. See the 'cipherSuite' setting in
  $SPLUNK_HOME/etc/system/default/inputs.conf for the current default.

ecdhCurveName = <string>
* DEPRECATED.
* Use the 'ecdhCurves' setting instead.
* This setting specifies the Elliptic Curve Diffie-Hellman (ECDH) curve to
  use for ECDH key negotiation.
* Splunk software only supports named curves that have been specified
  by their SHORT name.
* The list of valid named curves by their short and long names
  can be obtained by running this CLI command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Default: empty string

ecdhCurves = <comma-separated list>
* A list of ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of an SSL Client Hello.
* The server supports only the curves specified in the list.
* Splunk software only supports named curves that have been specified
  by their SHORT names.
* The list of valid named curves by their short and long names can be obtained
  by running this CLI command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Example setting: "ecdhCurves = prime256v1,secp384r1,secp521r1"
* The default can vary. See the 'ecdhCurves' setting in
  $SPLUNK_HOME/etc/system/default/inputs.conf for the current default.

dhFile = <string>
* Full path to the Diffie-Hellman parameter file.
* DH group size should be no less than 2048 bits.
* This file is required in order to enable any Diffie-Hellman ciphers.
* No default.

dhfile = <string>
* DEPRECATED.
* Use the 'dhFile' setting instead.
* Yes, the setting name is case-sensitive.

allowSslRenegotiation = <boolean>
* Whether or not to let SSL clients renegotiate their connections.
* In the SSL protocol, a client might request renegotiation of the connection
  settings from time to time.
* A value of "false" means the server rejects all renegotiation
  attempts, which breaks the connection.
```

```
* This limits the amount of CPU a single TCP connection can use, but it can
  cause connectivity problems, especially for long-lived connections.
* Default: true

sslQuietShutdown = <boolean>
* Enables quiet shutdown mode in SSL.
* Default: false

logCertificateData = <boolean>
* Whether or not the Splunk platform logs certificate data for
  Transport Layer Security (TLS) certificates.
* The certificate data logs provide visibility into the certificates
  in use for the Splunk-to-Splunk (S2S) channel. The logs collect data such
  as common name (CN), issuer name, SHA256 fingerprint, serial number, and
  validity dates.
* A value of "true" means that splunkd generates logs for TLS certificates.
* Refer to the 'certLogRepeatFrequency' setting for additional constraints on
  when the Splunk platform logs certificate data.
* Default: true

certLogMaxCacheEntries = <integer>
* The size of the cache for tracking certificate entries.
* The cache keeps track of the certificates for a time period of
  'certLogRepeatFrequency' to avoid generating duplicate logs.
* If the cache fills before the 'certLogRepeatFrequency' period elapses, the
  cache removes the entry with the oldest access time to make space.
* Update this setting as per the number of forwarders that are
  sending data to indexers. If the number of forwarders is larger than the
  cache size, some of the certificates might generate duplicate logs
  even though the previous log was within the 'certLogRepeatFrequency' period.
* When you restart Splunk Enterprise, the cache resets and the timer starts over.
* This setting takes effect only when 'logCertificateData' has a value of 'true'.
* Default: 10000

certLogRepeatFrequency = <timespan>
* The interval between writing repeat entries into the certificate data
  log for a certain certificate.
* This setting helps reduce certificate data log size by providing control
  over how often to log certificate data.
* When the Splunk platform receives a certificate the first time in a TLS
  connection, it adds the certificate to a cache. Subsequent connections
  with the same certificate won't generate a new entry to the log until
  a period of 'certLogRepeatFrequency' has passed. After this amount of
  time elapses, splunkd resets the log timestamp and writes another
  certificate log entry.
* The Splunk platform enforces this setting as long as the size of the cache
  does not reach 'certLogMaxCacheEntries'. When there are more than
  'certLogMaxCacheEntries', the cache removes the entry with the oldest
  access time to make space.
* When you restart Splunk Enterprise, the cache resets and the timer starts over.
* This setting takes effect only when 'logCertificateData' has a value of 'true'.
* A value of "0" means that the platform logs certificate data every time
  it receives a certificate.
* Default: 1d

sslCommonNameToCheck = <comma-separated list>
* Checks the common name of the client certificate against this list of names.
* If there is no match, assumes that the Splunk instance is not authenticated
  against this server.
* For this setting to work, you must also set 'requireClientCert' to "true".
* This setting is optional.
* Default: empty string (no common name checking)
```

```
sslAltNameToCheck = <comma-separated list>
* Checks the alternate name of the client certificate against this list of names.
* If there is no match, assumes that the Splunk instance is not authenticated
  against this server.
* For this setting to work, you must also set 'requireClientCert' to "true".
* This setting is optional.
* Default: empty string (no alternate name checking)

useSSLCompression = <boolean>
* Whether or not the server lets forwarders that connect to it negotiate SSL-
  layer data compression.
* A value of "true" means the server lets forwarders negotiate
  SSL-layer data compression.
* Default: The value of 'server.conf/[sslConfig]/allowSslCompression'

sslServerHandshakeTimeout = <integer>
* The timeout, in seconds, for an SSL handshake to complete between
  forwarder and the TCP input processor.
* If the TCP input processor does not receive a "Client Hello" from the forwarder
  within 'sslServerHandshakeTimeout' seconds, the server terminates
  the connection.
* Default: 60
```

UDP (User Datagram Protocol network input):

```
[udp://<remote server>:<port>]
* Similar to the [tcp://] stanza, except that this stanza causes the Splunk
  instance to listen on a UDP port.
* Only one stanza per port number is currently supported.
* Configures the instance to listen on a specific port.
* If you specify <remote server>, the specified port only accepts data
  from that host.
* If <remote server> is empty - [udp://<port>] - the port accepts data sent
  from any host.
  * The use of <remote server> is not recommended. Use the 'acceptFrom'
    setting, which supersedes this setting.
* Generates events with source set to udp:portnumber, for example: udp:514
* If you do not specify a sourcetype, generates events with sourcetype set
  to udp:portnumber.

# Additional settings:

connection_host = [ip|dns|none]
* "ip" sets the host to the IP address of the system sending the data.
* "dns" sets the host to the reverse DNS entry for IP address of the system
  that sends the data. For this to work correctly, set the forward DNS lookup
  to match the reverse DNS lookup in your DNS configuration.
* "none" leaves the host as specified in inputs.conf, typically the Splunk
  system hostname.
* If the input is configured with a 'sourcetype' that has a transform that
  overrides the 'host' field e.g. 'sourcetype=syslog', that takes
  precedence over the host specified here.
* Default: ip

_rcvbuf = <integer>
* The receive buffer, in bytes, for the UDP port.
* If you set the value to 0 or a negative number, the input ignores the value.
* If the default value is too large for an OS, the instance tries to set
```

```
  the value to 1572864/2. If that value is also too large, the instance
  retries with 1572864/(2*2). It continues to retry by halving the value until
  it succeeds.
* Default: 1572864

no_priority_stripping = <boolean>
* Whether or not the input strips <priority> syslog fields from events it
  receives over the syslog input.
* A value of "true" means the instance does NOT strip the <priority>
  syslog field from received events.
* NOTE: Do NOT set this setting if you want to strip <priority>.
* Default: false

no_appending_timestamp = <boolean>
* Whether or not to append a timestamp and host to received events.
* A value of "true" means the instance does NOT append a timestamp
  and host to received events.
* NOTE: Do NOT set this setting if you want to append timestamp and host
  to received events.
* Default: false

queueSize = <integer>[KB|MB|GB]
* The maximum size of the in-memory input queue.
* Default: 500KB

persistentQueueSize = <integer>[KB|MB|GB|TB]
* The maximum size of the persistent queue file.
* Persistent queues can help prevent loss of transient data. For information on
  persistent queues and how the 'queueSize' and 'persistentQueueSize' settings
  interact, search the online documentation for "persistent queues"..
* If you set this to a value other than 0, then 'persistentQueueSize' must
  be larger than either the in-memory queue size (as defined by the 'queueSize'
  setting in inputs.conf or 'maxSize' settings in [queue] stanzas in
  server.conf).
* Default: 0 (no persistent queue)

listenOnIPv6 = [no|yes|only]
* Select whether the instance listens on the IPv4, IPv6, or both protocols.
* Set this to 'yes' to listen on both IPv4 and IPv6 protocols.
* Set to 'only' to listen on only the IPv6 protocol.
* If not present, the input uses the setting in the [general] stanza
  of server.conf.

acceptFrom = <comma- or space-separated list>
* See the description for this setting in the [tcp://<remote server>:<port>]
  stanza.
* Default: "*" (accept from anywhere)

[udp:<port>]
* This input stanza is the same as [udp://<remote server>:<port>], but does
  not have a <remote server> restriction.
* See the documentation for [udp://<remote server>:<port>] to configure
  supported settings:

connection_host = [ip|dns|none]
_rcvbuf = <integer>
no_priority_stripping = <boolean>
no_appending_timestamp = <boolean>
queueSize = <integer>[KB|MB|GB]
persistentQueueSize = <integer>[KB|MB|GB|TB]
listenOnIPv6 = <no | yes | only>
acceptFrom = <comma- or space-separated list>
```

FIFO (First In, First Out queue):

```
[fifo://<path>]
* This stanza configures the monitoring of a FIFO at the specified path.

queueSize = <integer>[KB|MB|GB]
* Maximum size of the in-memory input queue.
* Default: 500KB

persistentQueueSize = <integer>[KB|MB|GB|TB]
* Maximum size of the persistent queue file.
* Persistent queues can help prevent loss of transient data. For information on
  persistent queues and how the 'queueSize' and 'persistentQueueSize' settings
  interact, search the online documentation for "persistent queues"..
* If you set this to a value other than 0, then 'persistentQueueSize' must
  be larger than either the in-memory queue size (as defined by the 'queueSize'
  setting in inputs.conf or 'maxSize' settings in [queue] stanzas in
  server.conf).
* Default: 0 (no persistent queue)
```

Scripted Input:

```
[script://<cmd>]
* Runs <cmd> at a configured interval and indexes the output
  that <cmd> returns.
* To determine the interval at which the input runs <cmd>,
  use the 'interval' setting.
* The <cmd> must reside in one of the following directories:
  * $SPLUNK_HOME/etc/system/bin/
  * $SPLUNK_HOME/etc/apps/$<APPNAME>/bin/
  * $SPLUNK_HOME/bin/scripts/
* The path to <cmd> can be an absolute path, make use of an environment
  variable such as $SPLUNK_HOME, or use the special pattern of an initial '.'
  as the first directory to indicate a location inside the current app.
  For more scripted input examples, search the documentation for
  "Add a scripted input with inputs.conf."
* <cmd> can also be a path to a file that ends with a ".path" suffix. A file
  with this suffix is a special type of pointer file that points to a command
  to be run. Although the pointer file is bound by the same location
  restrictions mentioned previously, the command referenced inside it can
  reside anywhere on the file system. The .path file must contain exactly
  one line: the path to the command to run, optionally followed by command-line
  arguments. The file can contain additional empty lines and lines that begin
  with '#'. The input ignores these lines.

interval = [<decimal>|<cron schedule>]
* How often, in seconds, to run the specified command, or a valid "cron"
  schedule.
* If you specify the interval as a number, it may have a fractional
  component; for example, 3.14
* To specify a cron schedule, use the following format:
  * "<minute> <hour> <day of month> <month> <day of week>"
  * Cron special characters are acceptable. You can use combinations of "*",
  ",", "/", and "-" to specify wildcards, separate values, specify ranges
```

```
  of values, and step values.
* The cron implementation for data inputs does not currently support names
  of months or days.
* The special value "0" forces this scripted input to be run continuously.
  As soon as the script exits, the input restarts it.
* The special value "-1" causes the scripted input to run once on start-up.
* NOTE: when you specify a cron schedule, the input does not run the
  script on start-up.
* Default: 60.0

passAuth = <string>
* The user to run the script as.
* If you provide a username, the instance generates an auth token for that
  user and passes it to the script through the stdin data stream.
* No default.

python.version = [default|python|python2|python3]
* For Python scripts only, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.

queueSize = <integer>[KB|MB|GB]
* Maximum size of the in-memory input queue.
* Default: 500KB

persistentQueueSize = <integer>[KB|MB|GB|TB]
* Maximum size of the persistent queue file.
* Persistent queues can help prevent loss of transient data. For information on
  persistent queues and how the 'queueSize' and 'persistentQueueSize' settings
  interact, search the online documentation for "persistent queues"..
* If you set this to a value other than 0, then 'persistentQueueSize' must
  be larger than either the in-memory queue size (as defined by the 'queueSize'
  setting in inputs.conf or 'maxSize' settings in [queue] stanzas in
  server.conf).
* Default: 0 (no persistent queue)

index = <string>
* The index where the scripted input sends the data.
* The script passes this parameter as a command-line argument to <cmd> in
  the format: -index <index name>.
  If the script does not need the index info, it can ignore this argument.
* If you do not specify an index, the script uses the default index.

send_index_as_argument_for_path = <boolean>
* Whether or not to pass the index as an argument when specified for
  stanzas that begin with 'script://'
* A value of "true" means the script passes the argument as
  '-index <index name>'.
* To avoid passing the index as a command line argument, set this to "false".
* Default: true

start_by_shell = <boolean>
* Whether or not to run the specified command through the operating system
  shell or command prompt.
* A value of "true" means the host operating system runs the
  specified command through the OS shell ("/bin/sh -c" on *NIX,
  "cmd.exe /c" on Windows.)
* A value of "false" means the input runs the program directly
  without attempting to expand shell metacharacters.
* You might want to explicitly set a value of "false" for scripts
```

```
  that you know do not need UNIX shell metacharacter expansion. This is
  a Splunk best practice.
* Default (on *nix machines): true
* Default (on Windows machines): false
```

File system change monitor (fschange monitor)

```
#
# The file system change monitor has been deprecated as of Splunk Enterprise
# version 5.0 and might be removed in a future version of the product.
#
# You cannot simultaneously monitor a directory with both the 'fschange'
# and 'monitor' stanza types.
#

[fschange:<path>]
* Monitors changes (such as additions, updates, and deletions) to this
  directory and any of its sub-directories.
* <path> is the direct path. Do not preface it with '//' like with
  other inputs.
* Sends an event for every change.

disabled = <boolean>
* Whether or not the file system change monitor input is active.
* Set a value of "true" to disable the input, and "false" to enable it.
* Default: false

# Additional settings:
# NOTE: The 'fschange' stanza type does not use the same settings as
# other input types. It uses only the following settings:

index = <string>
* The index where the input sends the data.
* Default: (if you either do not set 'signedaudit' or
  set 'signedaudit' to "false"): _audit
* Default: (in all other cases): the default index

signedaudit = <boolean>
* Whether or not to send cryptographically signed add/update/delete events.
* A value of "true" means the input does the following to
  events that it generates:
  * Puts the events in the _audit index.
  * Sets the event sourcetype to 'audittrail'
* A value of "false" means the input:
  * Places events in the default index.
  * Sets the sourcetype to whatever you specify (or "fs_notification"
    by default).
* You must set 'signedaudit' to "false" if you want to set the index for
  fschange events.
* You must also enable auditing by using the audit.conf file.
* Default: false

filters = <comma-separated list>
* The fschange input applies each filter, left to right, for each file
  or directory found during the monitor poll cycle.
* See the "File System Monitoring Filters" section later in this file
  for help on how to define a fschange filter.

recurse = <boolean>
* Whether or not the fschange input should look through all sub-directories
```

```
  for changes to files in a directory.
* A value of "true" means the input searches recursively through
  sub-directories within the directory specified in [fschange].
* Default: true

followLinks = <boolean>
* Whether or not the fschange input follows any symbolic
  links it encounters.
* A value of "true" means the input follows symbolic links.
* CAUTION: Do not set this setting to "true" unless you can confirm that
  doing so will not create a file system loop (For example, in
  Directory A, symbolic link B points back to Directory A.)
* Default: false

pollPeriod = <integer>
* How often, in seconds, to check a directory for changes.
* Default: 3600 (1 hour)

hashMaxSize = <integer>
* The maximum size, in bytes, that a file can be for the fschange input to
  calculate a SHA256 hash for that file.
* Tells the fschange input to calculate a SHA256 hash for every file that
  is this size or smaller, in bytes.
* The input uses this hash as an additional method for detecting changes to the
  file or directory.
* Default: -1 (disabled)

fullEvent = <boolean>
* Whether or not to send the full event if the input detects an add or
  update change.
* Set to "true" to send the full event if an add or update change is detected.
* Further qualified by the 'sendEventMaxSize' setting.
* Default: false

sendEventMaxSize = <integer>
* The maximum size, in bytes, that an fschange event can be for the input to
  send the full event to be indexed.
* Limits the size of event data that the fschange input sends.
* This also limits the size of indexed file data.
* Default: -1 (unlimited)

sourcetype = <string>
* Sets the source type for events from this input.
* The input automatically prepends "sourcetype=" to <string>.
* Default (if you set the 'signedaudit' setting to "true"): audittrail
* Default (if you set the 'signedaudit' setting to "false"): fs_notification

host = <string>
* Sets the host name for events from this input.
* Default: whatever host sent the event

filesPerDelay = <integer>
* The number of files that the fschange input processes between processing
  delays, as specified by the 'delayInMills' setting.
* After a delay of 'delayInMills' milliseconds, the fschange input processes
  'filesPerDelay' files, then waits 'delayInMills' milliseconds again before
  repeating this process.
* This setting helps throttle file system monitoring so it consumes less CPU.
* Default: 10

delayInMills = <integer>
* The delay, in milliseconds, that the fschange input waits between
```

```
  processing 'filesPerDelay' files.
* After a delay of 'delayInMills' milliseconds, the fschange input processes
  'filesPerDelay' files, then waits 'delayInMills' milliseconds again before
  repeating this process.
* This setting helps throttle file system monitoring so it consumes less CPU.
* Default: 100
```

File system monitoring filters:

```
[filter:<filtertype>:<filtername>]
* Defines a filter of type <filtertype> and names it <filtername>.
* <filtertype>:
  * Filter types are either 'blacklist' or 'whitelist.' 'blacklist' is the
    deny list filter type and 'whitelist' is the allow list filter type.
  * An allow list filter processes all file names that match the
    regular expression list that you define within the stanza.
  * A deny list filter skips all file names that match the
    regular expression list.
* <filtername>
  * The fschange input uses filter names that you specify with
    the 'filters' setting for a given fschange stanza.
  * You can specify multiple filters by separating them with commas.

regex<integer> = <regular expression>
* Deny list and allow list filters can include a set of regular expressions.
* The name of each regular expression MUST be 'regex<integer>', meaning the
  string "regex" and then an integer. <integer> starts at 1 and increments by 1.
* The input applies each regular expression in numeric order:
  regex1=<regular expression>
  regex2=<regular expression>
  ...
```

http: (HTTP Event Collector)

```
# Global settings for the HTTP Event Collector (HEC) Input.

[http]
port = <positive integer>
* The event collector data endpoint server port.
* Default: 8088

disabled = <boolean>
* Whether or not the event collector input is active.
* Give this setting a value of "1" to disable the input, and "0" to enable it.
* Default: 1 (disabled)

outputgroup = <string>
* The name of the output group to which the event collector forwards data.
* There is no support for using this setting to send data over HTTP with a heavy forwarder.
* Default: empty string

useDeploymentServer = <boolean>
* Whether or not the HTTP event collector input writes its
  configuration to a deployment server repository.
```

```
* When you enable this setting, the input writes its
  configuration to the directory that you specify with the
  'repositoryLocation' setting in the serverclass.conf file.
* You must copy the full contents of the splunk_httpinput app directory
  to this directory for the configuration to work.
* When enabled, only the tokens defined in the splunk_httpinput app in this
  repository are viewable and editable through the API and Splunk Web.
* When disabled, the input writes its configuration to
  $SPLUNK_HOME/etc/apps by default.
* Default: 0 (disabled)

index = <string>
* The default index to use.
* Default: the "default" index

sourcetype = <string>
* The default source type for the events that the input generates.
* If you do not specify a sourcetype, the input does not set a sourcetype
  for events it generates.

enableSSL = <boolean>
* Whether or not the HTTP Event Collector uses SSL.
* HEC shares SSL settings with the Splunk management server and cannot have
  SSL enabled when the Splunk management server has SSL disabled.
* Default: 1 (enabled)

dedicatedIoThreads = <non-negative integer>
* The number of dedicated input/output threads in the event collector
  input.
* Default: 0 (The input uses a single thread)

replyHeader.<name> = <string>
* Adds a static header to all HTTP responses that this server generates.
* For example, "replyHeader.My-Header = value" causes the
  response header "My-Header: value" to be included in the reply to
  every HTTP request made to the event collector endpoint server.
* No default.

maxSockets = <integer>
* The number of HTTP connections that the HTTP event collector input
  accepts simultaneously.
* Set this setting to constrain resource usage.
* If you set this setting to 0, the input automatically sets it to
  one third of the maximum allowable open files on the host.
* If this value is less than 50, the input sets it to 50. If this value
  is greater than 400000, the input sets it to 400000.
* If set to a negative value, the input does not enforce a limit on
  connections.
* Default: 0

maxThreads = <integer>
* The number of threads that can be used by active HTTP transactions.
* Set this to constrain resource usage.
* If you set this setting to 0, the input automatically sets the limit to
  one third of the maximum allowable threads on the host.
* If this value is less than 20, the input sets it to 20. If this value is
  greater than 150000, the input sets it to 150000.
* If the 'maxSockets' setting has a positive value and 'maxThreads'
  is greater than 'maxSockets', then the input sets 'maxThreads' to be equal
  to 'maxSockets'.
* If set to a negative value, the input does not enforce a limit on threads.
* Default: 0
```

```
rollingRestartReturnServerBusy = <boolean>
* Whether or not HTTP Event Collector endpoints return HTTP errors 404 (not found) or 503 (server busy)
  when a client connects to an indexer that is currently shutting down during a rolling restart.
* This setting applies to instances on the Classic Experience only.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true

keepAliveIdleTimeout = <integer>
* How long, in seconds, that the HTTP Event Collector input lets a keep-alive
  connection remain idle before forcibly disconnecting it.
* If this value is less than 7200, the input sets it to 7200.
* Default: 7200

busyKeepAliveIdleTimeout = <integer>
* How long, in seconds, that the HTTP Event Collector lets a keep-alive
  connection remain idle while in a busy state before forcibly disconnecting it.
* CAUTION: Setting this to a value that is too large
  can result in file descriptor exhaustion due to idling connections.
* If this value is less than 12, the input sets it to 12.
* Default: 12

serverCert = <string>
* The full path to the server certificate PEM format file.
* The same file may also contain a private key.
* Splunk software automatically generates certificates when it first
  starts.
* You may replace the auto-generated certificate with your own certificate.
* Default: $SPLUNK_HOME/etc/auth/server.pem

sslKeysfile = <string>
* DEPRECATED.
* Use the 'serverCert' setting instead.
* The file that contains the SSL keys. Splunk software looks for this file
  in the directory specified by 'caPath'.
* Default: server.pem

sslPassword = <string>
* The server certificate password.
* Initially set to a plain-text password.
* Upon first use, Splunk software encrypts and rewrites the password.
* Default: password

sslKeysfilePassword = <string>
* DEPRECATED.
* Use the 'sslPassword' setting instead.

caCertFile = <string>
* DEPRECATED.
* Use the 'server.conf:[sslConfig]/sslRootCAPath' setting instead.
* Used only if you do not set the 'sslRootCAPath' setting.
* Specifies the file name (relative to 'caPath') of the CA
  (Certificate Authority) certificate PEM format file that contains one or
  more certificates concatenated together.
* Default: cacert.pem

caPath = <string>
* DEPRECATED.
* Use absolute paths for all certificate files.
* If certificate files given by other settings in this stanza are not absolute
  paths, then they are relative to this path.
* Default: $SPLUNK_HOME/etc/auth
```

```
sslVersions = <comma-separated list>
* A comma-separated list of SSL versions to support.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2"
* The special version "*" selects all supported versions. The version "tls"
  selects all versions "tls1.0" or newer.
* To remove a version from the list, prefix it with "-".
* SSLv2 is always disabled. Specifying "-ssl2" in the version list
  has no effect.
* When configured in Federal Information Processing Standard (FIPS) mode, the
  "ssl3" version is always disabled, regardless of this configuration.
* Default: *,-ssl2  (anything newer than SSLv2)

cipherSuite = <string>
* The cipher string to use for the HTTP Event Collector input.
* Use this setting to ensure that the server does not accept connections using
  weak encryption protocols.
* If you set this setting, the input uses the specified cipher string for
  the HTTP server.
* Default: The default cipher string that 'OpenSSL' provides

sslServerHandshakeTimeout = <integer>
* The timeout, in seconds, for an SSL handshake to complete between an
  HEC client and the Splunk HEC server.
* If the HEC server does not receive a "Client Hello" from the HEC client within
  'sslServerHandshakeTimeout' seconds, the server terminates
  the connection.
* Default: 60

listenOnIPv6 = [no|yes|only]
* Whether or not this input listens on IPv4, IPv6, or both.
* Set to "no" to make the input listen only on the IPv4 protocol.
* Set to "yes" to make the input listen on both IPv4 and IPv6 protocols.
* Set to "only" to make the input listen on only the IPv6 protocol.
* Default: The setting in the [general] stanza of the server.conf file

acceptFrom = <comma- or space-separated list>
* See the description for this setting in the [tcp://<remote server>:<port>]
  stanza.
* Default: "*" (accept from anywhere)

requireClientCert = <boolean>
* Requires that any client connecting to the HEC port has a certificate that
  can be validated by the certificate authority specified in the
  'caCertFile' setting.
* Default: false

ecdhCurveName = <string>
* DEPRECATED.
* Use the 'ecdhCurves' setting instead.
* This setting specifies the ECDH curve to use for ECDH key negotiation.
* Splunk software only supports named curves that have been specified by their
  SHORT names.
* The list of valid named curves by their short or long names
  can be obtained by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Default: empty string

ecdhCurves = <comma-separated list>
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
```

```
* The server supports only the curves specified in the list.
* Splunk software only supports named curves that have been specified by their
  SHORT names.
* The list of valid named curves by their short or long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Example setting: ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: empty string

crossOriginSharingPolicy = <origin_acl> ...
* A list of the HTTP Origins for which to return Access-Control-Allow-*
  Cross-origin Resource Sharing (CORS) headers.
* These headers tell browsers that web applications at those sites
  can be trusted to make requests to the REST interface.
* The origin is passed as a URL without a path component (for example
  "https://app.example.com:8000").
* This setting can take a list of acceptable origins, separated
  by spaces and/or commas.
* Each origin can also contain wildcards for any part.  Examples:
    * *://app.example.com:*  (either HTTP or HTTPS on any port)
    * https://*.example.com  (any host under example.com, including
      example.com itself).
* An address can be prefixed with a '!' to negate the match, with
  the first matching origin taking precedence. Example:
    * "!*://evil.example.com:* *://*.example.com:*" to not avoid
      matching one host in a domain.
* "*" matches all origins.
* Default: empty string

crossOriginSharingHeaders = <string>
* A list of the HTTP headers to which splunkd sets
  "Access-Control-Allow-Headers" when replying to
  Cross-Origin Resource Sharing (CORS) preflight requests.
* The "Access-Control-Allow-Headers" header is used in response to
  a CORS preflight request to tell browsers which HTTP headers can be
  used during the actual request.
* A CORS preflight request is a CORS request that checks to see if
  the CORS protocol is understood and a server is aware of using
  specific methods and headers.
* This setting can take a list of acceptable HTTP headers, separated
  by commas.
* A single "*" can also be used to match all headers.
* Default: empty string

forceHttp10 = [auto|never|always]
* Whether or not the REST HTTP server forces clients that connect
  to it to use the HTTP 1.0 specification for web communications.
* When set to "always", the REST HTTP server does not use some
  HTTP 1.1 features such as persistent connections or chunked
  transfer encoding.
* When set to "auto", it does this only if the client did not send
  a User-Agent header, or if the user agent is known to have bugs
  in its support of HTTP/1.1.
* When set to "never" it always allows HTTP 1.1, even to
  clients it suspects might be buggy.
* Default: auto

sslCommonNameToCheck = <comma-separated list>
* A list of SSL Common Names to match against certificates that incoming
  HTTPS connections present to this instance.
* If you configure this setting and also set 'requireClientCert' to "true",
  splunkd limits most inbound HTTPS connections to hosts that use
```

```
  a cert with one of the listed common names.
* The most important scenario to use this setting is distributed search.
* This feature does not work with the deployment server and client
  communication over SSL.
* This setting is optional.
* Default: empty string (no common name checking)

sslAltNameToCheck = <comma-separated list>
* If you set this setting and also set 'requireClientCert' to true,
  splunkd can verify certificates that have a so-called
  "Subject Alternate Name" that matches any of the alternate
  names in this list.
  * Subject Alternate Names are effectively extended descriptive
    fields in SSL certs beyond the commonName. A common practice for
    HTTPS certs is to use these values to store additional valid
    hostnames or domains where the cert should be considered valid.
* Accepts a comma-separated list of Subject Alternate Names to consider
  valid.
* Items in this list are never validated against the SSL Common Name.
* This feature does not work with the deployment server and client
  communication over SSL.
* This setting is optional.
* Default: empty string (no alternate name checking)

sendStrictTransportSecurityHeader = <boolean>
* Whether or not to force inbound connections to always use SSL with
  the "Strict-Transport-Security" header..
* If set to "true", the REST interface sends a "Strict-Transport-Security"
  header with all responses to requests made over SSL.
* This can help prevent a client being tricked later by a Man-In-The-Middle
  attack to accept a non-SSL request. However, this requires a commitment that
  no non-SSL web hosts will ever be run on this hostname on any port. For
  example, if Splunk Web is in default non-SSL mode this can break the
  ability of the browser to connect to it. Enable with caution.
* Default: false

allowSslCompression = <boolean>
* Whether or not to allow data compression over SSL.
* If set to "true", the server allows clients to negotiate
  SSL-layer data compression.
* Default: true

allowSslRenegotiation = <boolean>
* Whether or not to let SSL clients renegotiate their connections.
* In the SSL protocol, a client may request renegotiation of the connection
  settings from time to time.
* Setting this to false causes the server to reject all renegotiation
  attempts, which breaks the connection.
* This limits the amount of CPU a single TCP connection can use, but it can
  cause connectivity problems, especially for long-lived connections.
* Default: true

ackIdleCleanup = <boolean>
* Whether or not to remove ACK channels that have been idle after a period
  of time, as defined by the 'maxIdleTime' setting.
* A value of "true" means the server removes the ACK channels that are idle
  for 'maxIdleTime' seconds.
* Default: true

maxIdleTime = <integer>
* The maximum amount of time, in seconds, that ACK channels can be idle
  before they are removed.
```

```
* If 'ackIdleCleanup' is "true", the system removes ACK channels that have
  been idle for 'maxIdleTime' seconds.
* Default: 600 (10 minutes)

channel_cookie = <string>
* The name of the cookie to use when sending data with a specified channel ID.
* The value of the cookie is the channel sent. For example, if you have
  set 'channel_cookie=foo' and sent a request with channel ID set to 'bar',
  then you will have a cookie in the response with the value 'foo=bar'.
* If no channel ID is present in the request, then no cookie is returned.
* This setting is to be used for load balancers (for example, AWS ELB) that can
  only provide sticky sessions on cookie values and not general header values.
* If no value is set (the default), then no cookie is returned.
* Default: empty string (no cookie)

maxEventSize = <positive integer>[KB|MB|GB]
* The maximum size of a single HEC (HTTP Event Collector) event.
* HEC disregards and triggers a parsing error for events whose size is
  greater than 'maxEventSize'.
* Default: 5MB
```

HTTP Event Collector (HEC) - Local stanza for each token

```
[http://name]

token = <string>
* The value of the HEC token.
* HEC uses this token to authenticate inbound connections. Your application
  or web client must present this token when attempting to connect to HEC.
* No default.

disabled = <boolean>
* Whether or not this token is active.
* Default: 0 (enabled)

description = <string>
* A human-readable description of this token.
* Default: empty string

indexes = <comma-separated list>
* The indexes that events for this token can go to.
* If you do not specify this value, the index list is empty, and any index
  can be used.
* Separate multiple indexes with commas.
* The Splunk platform accepts and indexes events without a specified
  index to a default index.
* No default.

s2s_indexes_validation = [ disabled | disabled_for_internal | enabled_for_all ]
* The method of index validation for Splunk-to-Splunk (S2S) events for this
  token.
* A value of "disabled" means the Splunk platform doesn't validate the
  event's index and the "indexes" setting has no effect for the S2S events.
* A value of "disabled_for_internal" means the Splunk platform doesn't
  validate internal indexes and allows all S2S events destined for them. The
  platform validates other indexes according to the "indexes" setting.
* A value of "enabled_for_all" means the platform validates all indexes
  according to the "indexes" setting.
```

```
* The platform silently drops rejected events.
* Default: disabled

index = <string>
* The default index to use for this token.
* Default: the default index

sourcetype = <string>
* The default sourcetype to use if it is not specified in an event.
* Default: empty string

outputgroup = <string>
* The name of the forwarding output group to send data to.
* There is no support for using this setting to send data over HTTP with a heavy forwarder.
* Default: empty string

queueSize = <integer>[KB|MB|GB]
* The maximum size of the in-memory input queue.
* Default: 500KB

persistentQueueSize = <integer>[KB|MB|GB|TB]
* Maximum size of the persistent queue file.
* Persistent queues can help prevent loss of transient data. For information on
  persistent queues and how the 'queueSize' and 'persistentQueueSize' settings
  interact, search the online documentation for "persistent queues"..
* If you set this to a value other than 0, then 'persistentQueueSize' must
  be larger than either the in-memory queue size (as defined by the
  'queueSize' setting in inputs.conf or 'maxSize' settings in [queue] stanzas
  in server.conf).
* Default: 0 (no persistent queue)

connection_host = [ip|dns|proxied_ip|none]
* Specifies the host if an event doesn't have a host set.
* "ip" sets the host to the IP address of the system sending the data.
* "dns" sets the host to the reverse DNS entry for IP address of the system
  that sends the data. For this to work correctly, set the forward DNS lookup
  to match the reverse DNS lookup in your DNS configuration.
* "proxied_ip" checks whether an X-Forwarded-For header was sent
  (presumably by a proxy server) and if so, sets the host to that value.
  Otherwise, the IP address of the system sending the data is used.
* "none" leaves the host as specified in the HTTP header.
* No default.

useACK = <boolean>
* When set to "true", acknowledgment (ACK) is enabled. Events in a request
  are tracked until they are indexed. An events status (indexed or not) can be
  queried from the ACK endpoint with the ID for the request.
* When set to false, acknowledgment is not enabled.
* This setting can be set at the stanza level.
* Default: false

allowQueryStringAuth = <boolean>
* Enables or disables sending authorization tokens with a query string.
* This is a token level configuration. It may only be set for
  a particular token.
* To use this feature, set to "true" and configure the client application to
  include the token in the query string portion of the URL they use to send
  data to HEC in the following format:
  "https://<URL>?<your=query-string>&token=<your-token>" or
  "https://<URL>?token=<your-token>" if the token is the first element in the
  query string.
* If a token is sent in both the query string and an HTTP header, the token in
```

```
  the query string takes precedence, even if this feature is disabled. In
  other words, if a token is present in the query string, any token in the
  header for that request is not used.
* NOTE: Query strings may be observed in transit and/or logged in cleartext.
  There is no confidentiality protection for the transmitted tokens.
    * Before using this in production, consult security personnel in your
      organization to understand and plan to mitigate the risks.
    * At a minimum, always use HTTPS when you enable this feature. Check your
      client application, proxy, and logging configurations to confirm that
      the token is not logged in clear text.
    * Give minimal access permissions to the token in HEC and restrict the
      use of the token only to trusted client applications.
* Default: false
```

WINDOWS INPUTS:

```
* Windows platform specific input processor.
# ***********
# Splunk software on Windows ships with several Windows-only inputs. They are
# defined in the default inputs.conf.

* Use the "disabled=" setting to enable/disable any of them.
* A short summary of the inputs follows:
  * Perfmon: Monitors Windows performance counters, objects, and instances.
  * WinRegMon: Tracks and report any changes that occur in the
    local system Registry.
  * ADMon: Indexes existing Active Directory (AD) objects and listens for AD
    changes.
  * WMI: Retrieves event logs remotely and locally through the Windows
    Management Instrumentation subsystem. It can also gather performance
    data remotely, as well as receive various system notifications. See
    wmi.conf.spec for information on how to configure this input.

#*******
# The following Windows input specifications are for parsing on non-Windows
# platforms.
#*******
```

Performance Monitor

```
[perfmon://<name>]

* This section explains possible settings for configuring
  the Windows Performance Monitor input.
* Each perfmon:// stanza represents an individually configured performance
  monitoring input. If you configure the input through Splunk Web, then the
  value of "<NAME>" matches what was specified there. While you can add
  performance monitor inputs manually, it is a best practice to use Splunk
  Web to configure them, because it is easy to mistype the values for
  Performance Monitor objects, counters, and instances.
* NOTE: The perfmon stanza is for local systems ONLY. To define performance
  monitor inputs for remote machines, use wmi.conf.

object = <string>
* A valid Performance Monitor object as defined within Performance
```

```
  Monitor (for example, "Process," "Server," "PhysicalDisk").
* You can specify a single valid Performance Monitor object or use a
  regular expression (regex) to specify multiple objects.
* This setting is required, and the input does not run if the setting is
  not present.
* No default.

counters = <semicolon-separated list>
* This can be a single counter, or multiple valid Performance Monitor
  counters.
* This setting is required, and the input does not run if the setting is
  not present.
* "*" is equivalent to all available counters for a given Performance
  Monitor object.
* No default.

nonmetric_counters = <semicolon-separated list>
* A list of performance counters on which the performance monitor input
  must not perform sampling.
* When the input retrieves the value for a counter that is in this list,
  it returns the latest value it retrieves, rather than an average of
  the values that it got over the sampling interval, as defined by the
  'samplingInterval' setting.
* Add counters to this setting in cases where the values that the input
  returns for a setting would be incorrect if it were averaged over a
  'samplingInterval', or where average, minimum, or maximum values for a
  counter would not be of any interest.
* As an example, the "ID Process" counter works better as a non metric counter
  because the most recent measurement of the counter is more relevant
  than the average of any measurements of that counter.
* No default.

instances = <semicolon-separated list>
* One or more multiple valid Performance Monitor instances.
* "*" is equivalent to all available instances for a given Performance Monitor
  counter.
* If applicable instances are available for a counter and this setting is not
  present, then the input logs data for all available instances (this is the
  same as setting "instances = *").
* If there are no applicable instances for a counter, then you can omit
  this setting.
* No default.

interval = <integer>
* How often, in seconds, to poll for new data.
* This setting is required, and the input does not run if the setting is
  not present.
* The recommended setting depends on the Performance Monitor object,
  counter(s), and instance(s) that you define in the input, and how much
  performance data you need.
  * Objects with numerous instantaneous or per-second counters, such
    as "Memory", "Processor", and "PhysicalDisk" should have shorter
    interval times specified (anywhere from 1-3 seconds).
  * Less volatile counters such as "Terminal Services", "Paging File",
    and "Print Queue" can have longer intervals configured.
* Default: 300

mode = [single|multikv]
* Specifies how the performance monitor input generates events.
* Set to "single" to print each event individually.
* Set to "multikv" to print events in multikv (formatted multiple
  key-value pair) format.
```

```
* Default: single

samplingInterval = <positive integer>
* How often, in milliseconds, to poll for new data.
* This is an advanced setting.
* Enables high-frequency performance sampling. The input collects
  performance data every sampling interval. It then reports averaged data
  and other statistics at every interval.
* The minimum legal value is 100, and the maximum legal value must be less
  than the 'interval' setting.
* If not set, high-frequency sampling does not occur.
* No default (disabled).

stats = <average;count;dev;min;max>
* Reports statistics for high-frequency performance sampling.
* This is an advanced setting.
* Setting a 'samplingInterval' is required to use 'stats'.
* Acceptable values are: average, count, dev, min, max.
* You can specify multiple values by separating them with semicolons.
* Adds new fields that append the stats function name.
  Setting 'average' replaces the stats displayed in the default field.
* No default. (disabled)

disabled = <boolean>
* Specifies whether or not the input is enabled.
* Set to 1 to disable the input, and 0 to enable it.
* Default: 0 (enabled)

showZeroValue = <boolean>
* Specifies whether or not the input collects zero-value event data.
* Set to 1 to capture zero value event data, and 0 to ignore such data.
* Default: 0 (ignore zero value event data)

useEnglishOnly = <boolean>
* Controls which Windows Performance Monitor API the input uses.
* If set to "true", the input uses PdhAddEnglishCounter() to add the
  counter string. This ensures that counters display in English
  regardless of the Windows machine locale.
* If set to "false", the input uses PdhAddCounter() to add the counter string.
* NOTE: if you set this setting to true, the 'object' setting does not
  accept a regular expression as a value on machines that have a non-English
  locale.
* Default: false

useWinApiProcStats = <boolean>
* Whether or not the Performance Monitor input uses process kernel mode and
  user mode times to calculate CPU usage for a process, rather than using
  the standard Performance Data Helper (PDH) APIs to calculate those values.
* A problem was found in the PDH APIs that causes Performance Monitor inputs
  to show maximum values of 100% usage for a process on multicore Windows
  machines, even when the process uses more than 1 core at a time.
* When you configure this setting to "true", the input uses the
  GetProcessTime() function in the core Windows API to calculate
  CPU usage for a process, for the following Performance Monitor
  counters, only:
** Processor Time
** User Time
** Privileged Time
* This means that, if a process uses 5 of 8 cores on an 8-core machine, that
  the input should return a value of around 500, rather than the incorrect 100.
* When you configure the setting to "false", the input uses the standard
  PDH APIs to calculate CPU usage for a process. On multicore systems, the
```

```
  maximum value that PDH APIs return is 100, regardless of the number of
  cores in the machine that the process uses.
* Performance monitor inputs use the PDH APIs for all other Performance
  Monitor counters. Configuring this setting has no effect on those counters.
* NOTE: If the Windows machine uses a non-English system locale, and you
  have set 'useWinApiProcStats' to "true" for a Performance Monitor input,
  then you must also set 'useEnglishOnly' to "true" for that input.
* Default: false

formatString = <string>
* Controls the print format for double-precision statistic counters.
* Do not use quotes when specifying this string.
* Default: %.20g

usePDHFmtNoCap100 = <boolean>
* Whether or not performance counter values that are greater than 100 (for example,
  counter values that measure the processor load on computers with multiple
  processors) are reset to 100.
* If set to "true", the counter values can exceed 100.
* If set to "false", the input resets counter values to 100 if the
  processor load on multiprocessor computers exceeds 100.
* Default: false
```

Direct Access File Monitor

```
# For Windows systems only.
# Does not use file handles

[MonitorNoHandle://<path>]

* This input intercepts file writes to the specific file.
* <path> must be a fully qualified path name to a specific file. Wildcards
  and directories are not accepted.
* This input type does not function on *nix machines.
* You can specify more than one stanza of this type.

disabled = <boolean>
* Whether or not the input is enabled.
* Default: 0 (enabled)

index = <string>
* Specifies the index where this input sends the data.
* This setting is optional.
* Default: the default index
```

Windows Event Log Monitor

```
[WinEventLog://<name>]

* This section explains possible settings for configuring the
  Windows Event Log monitor.
* Each WinEventLog:// stanza represents an individually configured WinEventLog
  monitoring input. If you you configure the input through Splunk Web, the
  value of "<NAME>" matches what was specified there. While you can add
  event log monitor inputs manually, it is best practice to use Splunk
```

```
  Web to configure Windows event log monitor inputs because it is
  easy to mistype the values for event log channels.
* NOTE: The WinEventLog stanza is for local systems ONLY. To define event log
  monitor inputs for remote machines, use wmi.conf.

start_from = <string>
* How the input should chronologically read the Event Log channels.
* If you set this setting to "oldest", the input reads Windows event logs
  from oldest to newest.
* If you set this setting to "newest" the input reads Windows event logs
  in reverse, from newest to oldest. Once the input consumes the backlog of
  events, it stops.
* If you set this setting to "newest", and at the same time set the
  "current_only" setting to 0, the combination can result in the input
  indexing duplicate events.
* Do not set this setting to "newest" and at the same time set the
  "current_only" setting to 1. This results in the input not collecting
  any events because you instructed it to read existing events from oldest
  to newest and read only incoming events concurrently (A logically
  impossible combination.)
* Default: "oldest"

use_old_eventlog_api = <boolean>
* Whether or not to read Event Log events with the Event Logging API.
* This is an advanced setting. Contact Splunk Support before you change it.
* A value of "true" means the input uses the Event Logging API (instead of the
  Windows Event Log API) to read from the Event Log on Windows Server 2008,
  Windows Vista, and later installations.
* Default: false (Use the API that is specific to the OS)

use_threads = <integer>
* Specifies the number of threads, in addition to the default writer thread,
  that can be created to filter events with the deny list/allow list
  regular expression.
* This is an advanced setting. Contact Splunk Support before you change it.
* The maximum number of threads is 15.
* Default: 0

thread_wait_time_msec = <integer>
* The interval, in milliseconds, between attempts to re-read Event Log files
  when a read error occurs.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: 5000

#
# NOTE: The 'suppress_*' settings are similar to, but operate differently than,
# the 'evt_exclude_fields' setting. The 'suppress_*' settings avoid using the
# Windows API to gather Windows events that match the available
# fields, which helps with CPU performance. The 'evt_exclude_fields'
# is valid for all Windows Event Log fields, and while it does use
# the Windows API for all transactions, it discards the fields in
# each event that match, which helps reduce total data ingestion.
#

suppress_checkpoint = <boolean>
* Whether or not the Event Log strictly follows the 'checkpointInterval'
  setting when it saves a checkpoint.
* This is an advanced setting. Contact Splunk Support before you change it.
* By default, the Event Log input saves a checkpoint from between zero
  and 'checkpointInterval' seconds, depending on incoming event volume.
  If you set this setting to "true", that does not happen.
* Default: false
```

```
suppress_sourcename = <boolean>
* Whether or not to exclude the 'sourcename' field from events.
* This is an advanced setting. Contact Splunk Support before you change it.
* When set to true, the input excludes the 'sourcename' field from events
  and thruput performance (the number of events processed per second) improves.
* Default: false

suppress_keywords = <boolean>
* Whether or not to exclude the 'keywords' field from events.
* This is an advanced setting. Contact Splunk Support before you change it.
* When set to true, the input excludes the 'keywords' field from events and
  thruput performance (the number of events processed per second) improves.
* Default: false

suppress_type = <boolean>
* Whether or not to exclude the 'type' field from events.
* This is an advanced setting. Contact Splunk Support before you change it.
* When set to true, the input excludes the 'type' field from events and
  thruput performance (the number of events processed per second) improves.
* Default: false

suppress_task = <boolean>
* Whether or not to exclude the 'task' field from events.
* This is an advanced setting. Contact Splunk Support before you change it.
* When set to true, the input excludes the 'task' field from events and
  thruput performance (the number of events processed per second) improves.
* Default: false

suppress_opcode = <boolean>
* Whether or not to exclude the 'opcode' field from events.
  When set to true, the input excludes the 'opcode' field from events and
  thruput performance (the number of events processed per second) improves.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: false

current_only = <boolean>
* Whether or not to acquire only events that arrive while the instance is
  running.
* If you set this setting to 1, the input only acquires events that arrive
  while the instance runs and the input is enabled. The input does not read
  data which was stored in the Windows Event Log while the instance was not
  running. This means that there will be gaps in the data if you restart the
  instance or experiences downtime.
* If you set the setting to 0, the input first gets all existing events
  already stored in the log that have higher event IDs (have arrived more
  recently) than the most recent events acquired. The input then monitors
  events that arrive in real time.
* If you set this setting to 0, and at the same time set the
  'start_from' setting to "newest", the combination can result in the
  indexing of duplicate events.
* Do not set this setting to 1 and at the same time set the
  'start_from' setting to "newest". This results in the input not collecting
  any events because you instructed it to read existing events from oldest
  to newest and read only incoming events concurrently (A logically
  impossible combination.)
* Default: 0 (false, gathering stored events first before monitoring
  live events)

batch_size = <integer>
* How many Windows Event Log items to read per request.
* If troubleshooting identifies that the Event Log input is a bottleneck in
```

```
  acquiring data, increasing this value can help.
  * NOTE: Splunk Support has seen cases where large values can result in a
    stall in the Event Log subsystem. If you increase this value
    significantly, monitor closely for trouble.
* In local and customer acceptance testing, a value of 10 was acceptable
  for both throughput and reliability.
* Default: 10

checkpointInterval = <integer>
* How often, in seconds, that the Windows Event Log input saves a checkpoint.
* Checkpoints store the eventID of acquired events. This lets the input
  continue monitoring at the correct event after a shutdown or outage.
* Default: 0

checkpointSync = <boolean>
* Determines whether the input processor forces writing a checkpoint file to disk
  immediately or lets the operating system handle when writing of the
  file to disk occurs.
* A value of "true" means the input processor triggers writing of a checkpoint
  file to disk immediately. It also saves the file to a temporary location and
  renames it instead of overwriting the existing file.
* Default: false

channel_wait_time = <integer>
* How long, in seconds, that the Windows Event Log input waits for an Event Log
  channel that is not available to become available again.
* Some Event Log channels, like the Windows Defender channel, become
  unavailable during a Windows Defender Platform update and it takes
  some time to become available again.
* If the Event Log input is unable to collect event logs from a certain
  Event Log channel, change this setting to an appropriate value.
  For example, if the input does not collect Windows Defender event logs
  after a Windows Defender Platform update, increase this value.
* The maximum wait time is 180 (3 minutes).
* Default: 0

disabled = <boolean>
* Whether or not the input is enabled.
* Set to 1 to disable the input, and 0 to enable it.
* Default: 0 (enabled)

evt_resolve_ad_obj = <boolean>
* How the input should interact with Active Directory while indexing Windows
  Event Log events.
* If you set this setting to true, the input resolves the Active
  Directory Security IDentifier (SID) objects to their canonical names for
  a specific Windows Event Log channel.
* If you enable the setting, the rate at which the input reads events
  on high-traffic Event Log channels can decrease. Latency can also increase
  during event acquisition. This is due to the overhead involved in performing
  AD translations.
* When you set this setting to true, you can optionally specify the domain
  controller name or dns name of the domain to bind to with the 'evt_dc_name'
  setting. The input connects to that domain controller to resolve the AD
  objects.
* If you set this setting to false, the input does not attempt any resolution.
* Default: false (disabled) for all channels

evt_skip_GUID_resolution = <comma-separated list>
* A list of Windows Event Codes for which the Splunk platform does not contact
  the domain controller to resolve global unique identifiers (GUIDs) that
  are withing the event.
```

```
* Separate multiple event IDs or event ID ranges with commas.
* If the event code matches an event, The Splunk platform does not contact
  the DC to resolve any GUIDs in this event.
* This setting only takes effect if 'evt_resolve_ad_obj' has a value of "true".
* If 'evt_resolve_ad_obj' has a value of "false", this setting has no effect.
* This setting has no effect on SID resolution.
* See 'Event ID list format' later in this file for the proper
  formatting of the event list.
* Default: none

evt_dc_name = <string>
* Which Active Directory domain controller to bind to for AD object
  resolution.
* If you prefix a dollar sign to a value (for example, $my_domain_controller),
  the input interprets the value as an environment variable. If the
  environment variable has not been defined on the host, it is the same
  as if the value is blank.
* This setting is optional.
* This setting can be set to the NetBIOS name of the domain controller
  or the fully-qualified DNS name of the domain controller. Either name
  type can, optionally, be preceded by two backslash characters. The following
  examples represent correctly formatted domain controller names:

    * "FTW-DC-01"
    * "\\FTW-DC-01"
    * "FTW-DC-01.splunk.com"
    * "\\FTW-DC-01.splunk.com"
    * $my_domain_controller

evt_dns_name = <string>
* The fully-qualified DNS name of the domain that the input should bind to for
  AD object resolution.
* This setting is optional.

evt_resolve_ad_ds = [auto|PDC]
* How the input should choose the domain controller to bind for
  AD resolution.
* This setting is optional.
* If set to PDC, the input only contacts the primary domain controller
  to resolve AD objects.
* If set to auto, the input lets Windows chose the best domain controller.
* If you set the 'evt_dc_name' setting, the input ignores this setting.
* Default: auto (let Windows determine the domain controller to use)

evt_ad_cache_disabled = <boolean>
* Enables or disables the AD object cache.
* Default: false (enabled)

evt_ad_cache_exp = <integer>
* The expiration time, in seconds, for AD object cache entries.
* This setting is optional.
* Default: 3600 (1 hour)

evt_ad_cache_exp_neg = <integer>
* The expiration time, in seconds, for negative AD object cache entries.
* This setting is optional.
* Default: 10

evt_ad_cache_max_entries = <integer>
* The maximum number of AD object cache entries.
* This setting is optional.
* Default: 1000
```

```
evt_exclude_fields = <comma-separated list>
* A list of valid Windows Event Log fields to exclude from Windows
  Event Log events.
* Specify fields that you want excluded from each event report.
* Do not exclude fields that you have also added to allow lists or
  deny lists. If fields are present in both, then 'evt_exclude_fields'
  excludes those fields, regardless of their presence in the allow list
  or deny list and the allow list or deny list will not behave as
  expected. The input logs an error to splunkd.log in this case.
* This setting is similar to, but operates differently than, the
  'suppress_*' settings. The 'suppress_*' settings avoid using the
  Windows API to gather Windows events that match the available
  fields, which helps with CPU performance. The 'evt_exclude_fields'
  is valid for all Windows Event Log fields, and while it does use
  the Windows API for all transactions, it discards the fields in
  each event that match, which helps reduce total data ingestion.
* Does not effect event report if 'renderXML' is set to "true".
* The 'evt_exclude_fields' setting is valid for all Windows Event Log fields.
* No default.

evt_sid_cache_disabled = <boolean>
* Enables or disables account Security IDentifier (SID) cache.
* This setting is global. It affects all Windows Event Log stanzas.
* Default: 0

evt_sid_cache_exp = <unsigned integer>
* The expiration time, in seconds, for account SID cache entries.
* This setting is global. It affects all Windows Event Log stanzas.
* This setting is optional.
* Default: 3600

evt_sid_cache_exp_neg = <unsigned integer>
* The expiration time, in seconds, for negative account SID cache entries.
* This setting is optional.
* This setting is global. It affects all Windows Event Log stanzas.
* Default: 10

evt_sid_cache_max_entries = <unsigned integer>
* The maximum number of account SID cache entries.
* This setting is global. It affects all Windows Event Log stanzas.
* This setting is optional.
* Default: 10

wec_event_format = [raw_event|rendered_event]
* The content format of the events that the Splunk platform expects to receive
  from a Windows Event Collector (WEC) subscription, before WEC sends the
  events to their destination log, for example, a Windows Event Log channel.
* This setting helps associate incoming WEC event formats with the Splunk
  platform internal interpretation before the platform looks up pre-rendered
  messages in Windows event logs.
* If the WEC subscription that targets this channel has its 'content Format'
  set to "Events", then set 'wec_event_format' to "raw_event".
* If the WEC subscription that targets this channel has its 'content Format'
  set to "RenderedText", then set 'wec_event_format' to "rendered_event".
* If multiple WEC subscriptions share the same value for the 'destination log'
  setting, but have different 'content Format' values, you have two options:
  * You can update the WEC subscriptions so that they share the same values for
    'content format'.
  * Or you can create custom ForwardedEvents channels for each WEC
    subscription, point each WEC subscription to a custom ForwardedEvents
    channel, and set equivalent values for 'wec_event_format' as described
```

```
    previously.
* If Windows Event Collector does not forward these events, this setting is
  optional.
* NOTE: You must restart the Splunk platform when you update WEC subscriptions,
  to synchronize with the new subscription configuration.
* Default (for 'ForwardedEvents' and custom channels named 'ForwardedEvents-1',
  'ForwardedEvents-2', etc.): rendered_event
* Default (for all other channels): raw_event

index = <string>
* Specifies the index that this input should send the data to.
* This setting is optional.
* Default: The default index
```

Event Log filtering

```
# Filtering at the input layer is desirable to reduce the total
# processing load in network transfer and computation on the Splunk platform
# nodes that acquire and processing Event Log data.

whitelist = <comma-separated list> | key=regex [key=regex]
blacklist = <comma-separated list> | key=regex [key=regex]

whitelist1 = <comma-separated list> | key=regex [key=regex]
whitelist2 = <comma-separated list> | key=regex [key=regex]
whitelist3 = <comma-separated list> | key=regex [key=regex]
whitelist4 = <comma-separated list> | key=regex [key=regex]
whitelist5 = <comma-separated list> | key=regex [key=regex]
whitelist6 = <comma-separated list> | key=regex [key=regex]
whitelist7 = <comma-separated list> | key=regex [key=regex]
whitelist8 = <comma-separated list> | key=regex [key=regex]
whitelist9 = <comma-separated list> | key=regex [key=regex]
blacklist1 = <comma-separated list> | key=regex [key=regex]
blacklist2 = <comma-separated list> | key=regex [key=regex]
blacklist3 = <comma-separated list> | key=regex [key=regex]
blacklist4 = <comma-separated list> | key=regex [key=regex]
blacklist5 = <comma-separated list> | key=regex [key=regex]
blacklist6 = <comma-separated list> | key=regex [key=regex]
blacklist7 = <comma-separated list> | key=regex [key=regex]
blacklist8 = <comma-separated list> | key=regex [key=regex]
blacklist9 = <comma-separated list> | key=regex [key=regex]

* These settings are optional.
* Both numbered and unnumbered allow lists and deny lists support two formats:
  * A comma-separated list of event IDs.
  * A list of key=regular expression pairs.
  * You cannot combine these formats. You can use either format on a specific
    line.

* Numbered allow list settings are permitted from 1 to 9, so whitelist1 through
  whitelist9 and blacklist1 through blacklist9 are supported.
* If no allow list or deny  list rules are present, the input reads all events.
```

Event Log allow list and deny list formats

```
* Event ID list format:
```

```
  * A comma-separated list of terms.
  * Terms may be a single event ID (e.g. 6) or range of event IDs (e.g. 100-200)
  * Example: 4,5,7,100-200
    * This applies to events with IDs 4, 5, 7, or any event ID between 100
      and 200, inclusive.
  * A single asterisk (*) means all event codes.
  * The event ID list format provides no additional functionality over the
    key=regex format, but can be easier to understand:
    List format:      4,5,7,100-200
    Regex equivalent: EventCode=%^(4|5|7|1..|200)$%

* key=regex format:
  * A whitespace-separated list of Event Log components to match, and
    regular expressions to match against against them.
  * There can be one match expression or multiple expressions per line.
  * The key must belong to the set of valid keys provided in the "Valid
    keys for the key=regex format" section.
  * The regex consists of a leading delimiter, the regex expression, and a
    trailing delimiter. Examples: %regex%, *regex*, "regex"
  * When multiple match expressions are present, they are treated as a
    logical AND.  In other words, all expressions must match for the line to
    apply to the event.
  * If the value represented by the key does not exist, it is not considered
    a match, regardless of the regex.
  * Example:
    whitelist = EventCode=%^200$% User=%jrodman%
    Include events only if they have EventCode 200 and relate to User jrodman

# Valid keys for the key=regex format:

* The following keys are equivalent to the fields that appear in the text of
  the acquired events:
  * Category, CategoryString, ComputerName, EventCode, EventType, Keywords,
    LogName, Message, OpCode, RecordNumber, Sid, SidType, SourceName,
    TaskCategory, Type, User
* There are three special keys that do not appear literally in the event.
  * $TimeGenerated: The time that the computer generated the event
  * $Timestamp: The time that the event was received and recorded by the
                Event Log service.
  * $XmlRegex: Use this key for filtering when you render Windows Event
    log events in XML by setting the 'renderXml' setting to "true". Search
    the online documentation for "Filter data in XML format with the
    XmlRegex key" for details.
* The 'EventType' key is only available on Windows Server 2003 /
  Windows XP and earlier.
* The 'Type' key is only available on Windows Server 2008 /
  Windows Vista and later.
* For a detailed definition of these keys, see the
  "Monitor Windows Event Log Data" topic in the online documentation.

suppress_text = <boolean>
* Whether or not to include the description of the event text for a
  given Event Log event.
* This setting is optional.
* Set this setting to true to suppress the inclusion of the event
  text description.
* Set this value to false to include the event text description.
* Default: false

renderXml = <boolean>
* Whether or not the input returns the event data in XML (eXtensible Markup
  Language) format or in plain text.
```

```
* A value of "true" means that the input renders events in XML format.
* A value of "false" means that the input renders events in plain text.
* If you give this setting a value of "true", you should also give the
  'suppress_text', 'suppress_sourcename', 'suppress_keywords', 'suppress_task', and
  'suppress_opcode' settings a value of "true" to improve thruput performance.
* A value of "true" also changes the method by which you create allow-
  and deny lists to filter events. For these kinds of lists to work, you
  must use the '$xmlRegex' special key and assign regular expression values
  to use those lists.
* Search the Splunk Documentation for "Filter data in XML format with the
  XmlRegex Key" for details.
* Default: false
```

Active Directory Monitor

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

* This section explains possible settings for configuring the Active Directory
  monitor input.
* Each admon:// stanza represents an individually configured Active
  Directory monitoring input. If you configure the input with Splunk Web,
  then the value of "<NAME>" matches what was specified there. While
  you can add Active Directory monitor inputs manually, it is best practice
  to use Splunk Web to configure Active Directory monitor
  inputs because it is easy to mistype the values for Active Directory
  monitor objects.

targetDc = <string>
* The fully qualified domain name of a valid, network-accessible
  Active Directory domain controller (DC).
* This setting is case sensitive. Do not use 'targetdc' or 'targetDC',
  but rather 'targetDc'.
* Default: The DC that the local host used to connect to AD. The
  input binds to its root Distinguished Name (DN).

startingNode = <string>
* Where in the Active Directory directory tree to start monitoring.
* The user that you configure Splunk software to run as at
  installation determines where the input starts monitoring.
* Default: the root of the directory tree

monitorSubtree = <boolean>
* Whether or not to monitor the subtree(s) of a given Active
  Directory tree path.
* Set this to 1 to monitor subtrees of a given directory tree
  path and 0 to monitor only the path itself.
* Default: 1 (monitor subtrees of a given directory tree path)

disabled = <boolean>
* Whether or not the input is enabled.
* Set this to 1 to disable the input and 0 to enable it.
* Default: 0 (enabled)

index = <string>
* The index to store incoming data into for this input.
* This setting is optional.
* Default: the default index

printSchema = <boolean>
```

```
* Whether or not to print the Active Directory schema.
* Set this to 1 to print the schema and 0 to not print
  the schema.
* Default: 1 (print the Active Directory schema)

baseline = <boolean>
* Whether or not to query baseline objects.
* Baseline objects are objects which currently reside in Active Directory.
* Baseline objects also include previously deleted objects.
* Set this to 1 to query baseline objects, and 0 to not query
  baseline objects.
* Default: 0 (do not query baseline objects)
```

Windows Registry Monitor

```
[WinRegMon://<name>]

* This section explains possible settings for configuring the Windows Registry
  Monitor input.
* Each WinRegMon:// stanza represents an individually configured
  WinRegMon monitoring input.
* If you configure the inputs with Splunk Web, the value of "<NAME>" matches
  what was specified there. While you can add event log monitor inputs
  manually, it is best practice to use Splunk Web to configure
  Windows registry monitor inputs because it is easy to mistype the values
  for Registry hives and keys.
* The WinRegMon input is for local systems only. You cannot monitor the
  Registry remotely.

proc = <string>
* The processes this input should monitor for Registry access.
* If set, matches against the process name which performed the Registry
  access.
* The input includes events from processes that match the regular expression
  that you specify here.
* The input filters out events for processes that do not match the
  regular expression.
* Default: .* (match all processes)

hive = <string>
* The Registry hive(s) that this input should monitor for Registry access.
* If set, matches against the Registry key that was accessed.
* The input includes events from Registry hives that match the
  regular expression that you specify here.
* The input filters out events for Registry hives that do not match the
  regular expression.
* No default.

type = <string>
* A regular expression that specifies the type(s) of Registry event(s)
  that you want the input to monitor.
* No default.

baseline = <boolean>
* Whether or not the input should get a baseline of Registry events
  when it starts.
* If you set this to 1, the input captures a baseline for
  the specified hive when it starts for the first time. It then
  monitors live events.
```

```
* Default: 0 (do not capture a baseline for the specified hive
  first before monitoring live events)

baseline_interval = <integer>
* Selects how much downtime in continuous registry monitoring should trigger
  a new baseline for the monitored hive and/or key.
* In detail:
  * Sets the minimum time interval, in seconds, between baselines.
  * At startup, a WinRegMon input does not generate a baseline if less time
    has passed since the last checkpoint than baseline_interval chooses.
  * In normal operation, checkpoints are updated frequently as data is
    acquired, so this will cause baselines to occur only when monitoring was
    not operating for a period of time.
* If baseline is set to 0 (disabled), the setting has no effect.
* Default: 86400 (1 day)

disabled = <boolean>
* Whether or not the input is enabled.
* Set this to 1 to disable the input, or 0 to enable it.
* Default: 0 (enabled)

index = <string>
* The index that this input should send the data to.
* This setting is optional.
* Default: the default index
```

Windows Host Monitoring

```
[WinHostMon://<name>]

* This section explains possible settings for configuring the Windows host
  monitor input.
* Gathers status information from the local Windows system components as
  per the 'type' field, described after this section.
* Each WinHostMon:// stanza represents an WinHostMon monitoring input.
* The "<name>" component of the stanza name is used as the source field
  on generated events, unless an explicit source setting is added to the
  stanza.  It does not affect what data is collected (see type setting for
  that).
* If you configure the input in Splunk Web, the value of "<name>" matches
  what was specified there.
* NOTE: The WinHostMon input is for local Windows systems only. You
  cannot monitor Windows host information remotely.

type = <semicolon-separated list>
* An expression that specifies the type(s) of host inputs
  that you want the input to monitor.
* Type can be (case insensitive):
  Computer;Process;Processor;NetworkAdapter;Service;OperatingSystem;Disk;Driver;Roles
* No default.

interval = <integer>
* The interval, in seconds, between when the input runs to gather
  Windows host information and generate events.
* See 'interval' in the Scripted input section for more information.

disabled = <boolean>
* Whether or not the input is enabled.
* Set this to 1 to disable the input, or 0 to enable it.
```

```
* Default: 0 (enabled)

index = <string>
* The index that this input should send the data to.
* This setting is optional.
* Default: the default index

[WinPrintMon://<name>]

* This section explains possible settings for configuring the Windows print
  monitor input.
* Each WinPrintMon:// stanza represents an WinPrintMon monitoring input.
  The value of "<name>" matches what was specified in Splunk Web.
* NOTE: The WinPrintMon input is for local Windows systems only.
* The "<name>" component of the stanza name is used as the source field
  on generated events, unless an explicit source setting is added to the
  stanza.  It does not affect what data is collected (see type setting for
  that).

type = <semicolon-separated list>
* An expression that specifies the type(s) of print inputs
  that you want the input to monitor.
* Type can be (case insensitive):
  Printer;Job;Driver;Port
* No default.

interval = <integer>
* The interval, in seconds, between when the input runs to gather
  Windows host information and generate events.
* See 'interval' in the Scripted input section for more information.

baseline = <boolean>
* Whether or not to capture a baseline of print objects when the
  input starts for the first time.
* If you set this setting to 1, the input captures a baseline of
  the current print objects when the input starts for the first time.
* Default: 0 (do not capture a baseline)

disabled = <boolean>
* Whether or not the input is enabled.
* Set to 1 to disable the input, or 0 to enable it.
* Default: 0 (enabled)

index = <string>
* The index that this input should send the data to.
* This setting is optional.
* Default: the default index

[WinNetMon://<name>]

* This section explains possible settings for configuring
  a Network Monitor input.
* Each WinNetMon:// stanza represents an individually configured network
  monitoring input. The value of "<name>" matches what was specified
  in Splunk Web. It is best practice to use Splunk Web to
  configure Network Monitor inputs because it is easy to mistype
  the values for Network Monitor objects.

remoteAddress = <regular expression>
* A regular expression that represents the remote IP address of a
  host that is involved in network communication.
* This setting accepts a regular expression that matches against
```

```
  IP addresses only, not host names. For example: 192\.168\..*
* The input includes events for remote IP addresses that match
  the regular expression that you specify here.
* The input filters out events for remote IP addresses that do not
  match the regular expression.
* No default (include all remote address events).

process = <regular expression>
* A regular expression that represents the process or application that
  performed a network access.
* The input includes events for processes that match the
  regular expression that you specify here.
* The input filters out events for processes that do not match the
  regular expression.
* No default (include all processes and application events).

user = <regular expression>
* A regular expression that represents the Windows user name that
  performed a network access.
* The input includes events for user names that match the
  regular expression that you specify here.
* The input filters out events for user names that do not match the
  regular expression.
* No default (include all user name events).

addressFamily = [ipv4];[ipv6]
* Determines the events to include by network address family.
* Setting "ipv4" alone includes only IPv4 packets, while "ipv6" alone
  includes only IPv6 packets.
* To specify both families, separate them with a semicolon.
  For example: ipv4;ipv6
* No default (include events with both address families).

packetType = [connect];[accept];[transport]
* Determines the events to include by network packet type.
* To specify multiple packet types, separate them with a semicolon.
  For example: connect;transport
* No default (include events with any packet type).

direction = [inbound];[outbound]
* Determines the events to include by network transport direction.
* To specify multiple directions, separate them with a semicolon.
  For example: inbound;outbound
* No default (include events with any direction).

protocol = [tcp];[udp]
* Determines the events to include by network protocol.
* To specify multiple protocols, separate them with a semicolon.
  For example: tcp;udp
* For more information about protocols, see
  http://www.ietf.org/rfc/rfc1700.txt
* No default (include events with all protocols)

readInterval = <integer>
* How often, in milliseconds, that the input should read the network
  kernel driver for events.
* Advanced option. Use the default value unless there is a problem
  with input performance.
* Set this to adjust the frequency of calls into the network kernel driver.
* Choosing lower values (higher frequencies) can reduce network
  performance, while higher numbers (lower frequencies) can cause event
  loss.
```

```
* The minimum allowed value is 10 and the maximum allowed value is 1000.
* Default: 100

driverBufferSize = <integer>
* The maximum number of packets that the network kernel driver retains
  for retrieval by the input.
* Set to adjust the maximum number of network packets retained in
  the network driver buffer.
* Advanced option. Use the default value unless there is a problem
  with input performance.
* Configuring this setting to lower values can result in event loss, while
  higher values can increase the size of non-paged memory on the host.
* The minimum allowed value is 128 and the maximum allowed value is 32768.
* Default: 32768

userBufferSize = <integer>
* The maximum size, in megabytes, of the user mode event buffer.
* Controls amount of packets cached in the the user mode.
* Advanced option. Use the default value unless there is a problem
  with input performance.
* Configuring this setting to lower values can result in event loss, while
  higher values can increase the amount of memory that the network
  monitor uses.
* The minimum allowed value is 20 and the maximum allowed value is 500.
* Default: 20

mode = [single|multikv]
* Specifies how the network monitor input generates events.
* Set to "single" to generate one event per packet.
* Set to "multikv" to generate combined events of many packets in
  multikv format (many packets described in a single table as one event).
* Default: single

multikvMaxEventCount = <integer>
* The maximum number of packets to combine in multikv format when you set
  the 'mode' setting to "multikv".
* Has no effect when 'mode' is set to "single".
* Advanced option.
* The minimum allowed value is 10 and the maximum allowed value is 500.
* Default: 100

multikvMaxTimeMs = <integer>
* The maximum amount of time, in milliseconds, to accumulate packet data to
  combine into a large tabular event in multikv format.
* Has no effect when 'mode' is set to 'single'.
* Advanced option.
* The minimum allowed value is 100 and the maximum allowed value is 5000.
* Default: 1000

sid_cache_disabled = [0|1]
* Enables or disables account Security IDentifier (SID) cache.
* This setting is global. It affects all Windows Network Monitor stanzas.
* Default: 0

sid_cache_exp = <integer>
* The expiration time, in seconds, for account SID cache entries.
* Optional.
* This setting is global. It affects all Windows Network Monitor stanzas.
* Default: 3600

sid_cache_exp_neg = <integer>
* The expiration time, in seconds, for negative account SID cache entries.
```

```
* Optional.
* This setting is global. It affects all Windows Network Monitor stanzas.
* Default: 10

sid_cache_max_entries = <integer>
* The maximum number of account SID cache entries.
* Optional.
* This setting is global. It affects all Windows Network Monitor stanzas.
* Default: 10

disabled = <boolean>
* Whether or not the input is enabled.
* Set to 1 to disable the input, and 0 to enable it.
* Default: 0 (enabled)

index = <string>
* The index where this input sends the data.
* Optional.
* Default: the default index

# Global settings for the powershell modinput.

[powershell]
io_threads = <integer>
* The number of threads that Splunk software spawns to run PowerShell scripts
  that have been configured in the inputs.conf file.
* If you specify a value that is less than or equal to 0, Splunk software
  autotunes this setting.
* The default can vary. Splunk software autotunes the number of threads
  based on the availability of CPU resources on the machine.

serialization_threads = <integer>
* The number of threads that Splunk software spawns for serialization of
  PowerShell objects that it has collected into XML strings.
* This serialization, or conversion of objects, occurs according to the
  Modular Input XML protocol.
* If you specify a value that is less than or equal to 0, Splunk software
  autotunes this setting.
* The default can vary. Splunk software autotunes the number of threads
  based on available CPU resources on the machine.

event_serialization_format = [kv|json]
* The event format into which Powershell objects are serialized.
* The supported event formats are "kv" and "json".
* For example, given the following PowerShell object:

   $psObj = @{
      A: "a string"
      B: 18
      C: "a log line"
   }

   If you set 'event_serialization_format' to "kv", the Splunk platform
   indexes the event as follows:

   A="a string"
   B=18
   C="a log line"

   If you set 'event_serialization_format' to "json", the Splunk platform
   indexes the event as follows:
```

```
   {
       "A": "a string",
       "B": 18,
       "C": "a log line"
   }
* Default: kv

process_completion_check_interval = <integer>
* The interval, in milliseconds, between which the Splunk platform checks
  whether a PowerShell process has completed running.
* Default: 200

[powershell://<name>]
* Runs Windows PowerShell version 3 commands or scripts.

script = <string>
* A PowerShell command-line script or .ps1 script file that the input
  should run.
* No default.

schedule = [<positive integer>|<cron schedule>]
* How often to run the specified PowerShell command or script.
* You can specify a number in seconds, or provide a valid cron
  schedule.
* Default: Runs the command or script once, at startup.

# Global settings for the powershell2 modinput.

[powershell2]
io_threads = <integer>
* The number of threads that Splunk software spawns to run PowerShell scripts
  that have been configured in inputs.conf.
* If you specify a value that is less than or equal to 0, Splunk software
  autotunes this setting.
* The default can vary. Splunk software autotunes the number of threads
  based on the availability of CPU resources on the machine.

event_serialization_format = [ kv | json ]
* The event format into which Powershell objects are serialized.
* The supported event formats are "kv" and "json".
* For example, given the following PowerShell object:

   $psObj = @{
      A: "a string"
      B: 18
      C: "a log line"
   }

   If you set 'event_serialization_format' to "kv", the Splunk platform
   indexes the event as follows:

   A="a string"
   B=18
   C="a log line"

   If you set 'event_serialization_format' to "json", the Splunk platform
   indexes the event as follows:

   {
       "A": "a string",
       "B": 18,
       "C": "a log line"
```

```
   }
* Default: kv

process_completion_check_interval = <integer>
* The interval, in milliseconds, between which the Splunk platform checks
  whether a PowerShell process has completed running.
* Default = 200

[powershell2://<name>]
* Runs Windows PowerShell version 2 commands or scripts.

script = <string>
* A PowerShell command-line script or .ps1 script file that the input
  should run.
* No default.

schedule = <string>
* How often to run the specified PowerShell command or script.
* You can provide a valid cron schedule.
* Default: Runs the command or script once, at startup.
```

Remote Queue Monitor

```
[remote_queue:<name>]

* This section explains possible settings for configuring a remote queue.
* Each remote_queue: stanza represents an individually configured remote
  queue monitoring input.
* Note that only ONE remote queue stanza is supported as
  an input queue.

remote_queue.* = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* This section explains possible settings for configuring a remote queue.
* With remote queues, the splunk indexer might require additional configuration,
  specific to the type of remote queue. You can pass configuration information
  to the splunk indexer by specifying the settings through the following schema:
  remote_queue.<scheme>.<config-variable> = <value>.
  For example:
  remote_queue.sqs.access_key = ACCESS_KEY
* This setting is optional.
* No default.

remote_queue.type = [sqs|kinesis|sqs_smartbus]
* Currently not supported. This setting is related to a feature that is
  still under development.
* Required.
* Specifies the remote queue type, either Amazon Web Services (AWS)
  Simple Queue Service (SQS) or Amazon Kinesis or SQS Smartbus.

remote_queue.large_message_store.supports_versioning = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies whether or not the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage.
* This setting is optional.
```

```
* Default: true

compressed = <boolean>
* See the description for TCPOUT ATTRIBUTES in outputs.conf.spec.

negotiateProtocolLevel = <unsigned integer>
* See the description for TCPOUT ATTRIBUTES in outputs.conf.spec.

channelReapInterval = <integer>
* See the description for TCPOUT ATTRIBUTES in outputs.conf.spec.

channelTTL = <integer>
* See the description for TCPOUT ATTRIBUTES in outputs.conf.spec.

channelReapLowater = <integer>
* See the description for TCPOUT ATTRIBUTES in outputs.conf.spec.

concurrentChannelLimit = <unsigned integer>
* See the description for [splunktcp].
```

SQS specific settings

```
remote_queue.sqs.access_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The access key to use when authenticating with the remote queue
  system supporting the SQS API.
* If not specified, the indexer looks for these environment variables:
  'AWS_ACCESS_KEY_ID' or 'AWS_ACCESS_KEY' (in that order). If the environment
  variables are not set and the indexer is running on Elastic Compute Cloud
  (EC2), the indexer attempts to use the secret key from the Identity and
  Access Management (IAM) role.
* This setting is optional.
* No default.

remote_queue.sqs.secret_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The secret key to use when authenticating with the remote queue
  system supporting the SQS API.
* If not specified, the indexer looks for these environment variables:
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order). If the environment
  variables are not set and the indexer is running on EC2, the indexer attempts
  to use the secret key from the IAM role.
* This setting is optional.
* No default.

remote_queue.sqs.auth_region = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The authentication region to use when signing the requests when interacting
  with the remote queue system supporting the SQS API.
* If not specified and the indexer is running on EC2, the auth_region is
  constructed automatically based on the EC2 region of the instance where the
  the indexer is running.
* This setting is optional.
* No default.

remote_queue.sqs.endpoint = <string>
```

```
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote queue system supporting the SQS API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://sqs.<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified in 'remote_queue.sqs.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://sqs.us-west-2.amazonaws.com/
* This setting is optional.
* No default.

remote_queue.sqs.max_connections = <unsigned integer>
* Currently not supported. This setting is related to a feature that is still
  under development.
* The maximum number of HTTP connections to have in progress for
  certain queue operations.
* A value of 0 means unlimited.
* Default: 8

remote_queue.sqs.message_group_id = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The Message Group ID for Amazon Web Services Simple Queue Service
  (SQS) First-In, First-Out (FIFO) queues.
* Setting a Message Group ID controls how messages within an AWS SQS queue are
  processed.
* For information on SQS FIFO queues and how messages in those queues are
  processed, see "Recommendations for FIFO queues" in the AWS SQS Developer
  Guide.
* If you configure this setting, Splunk software assumes that the SQS queue is
  a FIFO queue, and that messages in the queue should be processed first-in,
  first-out.
* Otherwise, Splunk software assumes that the SQS queue is a standard queue.
* Can be between 1-128 alphanumeric or punctuation characters.
* NOTE: FIFO queues must have Content-Based Deduplication enabled.
* This setting is optional.
* No default.

remote_queue.sqs.retry_policy = [max_count|none]
* Currently not supported. This setting is related to a feature that is still
  under development.
* The retry policy to use for remote queue operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a queue operation can be
    retried upon intermittent failure.
  + "none": Do not retry file operations upon failure.
* This setting is optional.
* Default: "max_count"

remote_queue.sqs.max_count.max_retries_per_part = <unsigned integer>
* Currently not supported. This setting is related to a feature that is still
  under development.
* When 'remote_queue.sqs.retry_policy' is set to "max_count", sets the maximum
  number of times a queue operation can be retried upon intermittent failure.
* This setting is optional.
* Default: 9
```

```
remote_queue.sqs.timeout.connect = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The connection timeout, in seconds, when interacting with
  SQS for this queue.
* This setting is optional.
* Default: 5

remote_queue.sqs.timeout.read = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The read timeout, in seconds, when interacting with SQS for
  this queue.
* This setting is optional.
* Default: 60

remote_queue.sqs.timeout.write = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The write timeout, in seconds, when interacting with SQS for
  this queue.
* This setting is optional.
* Default: 60

remote_queue.sqs.timeout.receive_message = <unsigned integer>
* The receive message wait time, in seconds, when interacting with SQS for
  this queue.
* When set to greater than 0, enables "long polling." If there are no messages
  immediately available, the queue waits at most
  'remote_queue.sqs.timeout.receive_message' seconds for a message to
  become available.
* When 0, disables long polling.
* When not set, uses the value configured for the queue via the AWS SQS
  console.
* Maximum value: 20
* This setting is optional.
* Default: 20

remote_queue.sqs.timeout.visibility = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default "visibility timeout," in seconds, to use when
  explicitly changing the visibility of specific messages in the queue.
* NOTE: Changing the value of 'remote_queue.sqs.timeout.visibility'
  does not change the implicit visibility timeout configured for
  the queue in the AWS SQS console.
* This setting is optional.
* Default: 60

remote_queue.sqs.buffer.visibility = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default time, in seconds, before
  'remote_queue.sqs.timeout.visibility' at which visibility of
  specific messages in the queue needs to be changed.
* This setting is optional.
* Default: 15

remote_queue.sqs.executor_max_workers_count = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum number of worker threads that can be used by
```

```
  indexer per pipeline set to execute SQS tasks.
* A value of 0 is equivalent to 1.
* Default: 8

remote_queue.sqs.min_pending_messages = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default "minimum number of pending messages" to use before
  receiving messages off remote queue.
  Messages are only received when the sum of the internal queue message count and
  pending object GET (from large messages storage) count is below
  the set value.
* This setting is optional.
* Default: 10

remote_queue.sqs.large_message_store.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote storage system supporting the S3 API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://s3-<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via 'remote_queue.sqs.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://s3-us-west-2.amazonaws.com/
* This setting is optional.
* No default.

remote_queue.sqs.large_message_store.path = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The remote storage location where messages that are larger than the
  underlying queue maximum message size will reside.
* The format for this attribute is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string for
    identifying a location inside the storage system.
* These external systems are supported:
  - Object stores that support the AWS S3 protocol. These use the scheme "s3".
    For example, "path=s3://mybucket/some/path".
* If not specified, messages exceeding the underlying queue's maximum message
  size are dropped.
* This setting is optional.
* No default.
```

Kinesis specific settings

```
remote_queue.kinesis.access_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the access key to use when authenticating with the remote queue
  system supporting the Kinesis API.
* If not specified, the forwarder will look for these environment variables:
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the forwarder
  attempts to use the secret key from the IAM role.
* This setting is optional.
```

```
* No default.

remote_queue.kinesis.secret_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the secret key to use when authenticating with the remote queue
  system supporting the Kinesis API.
* If not specified, the forwarder will look for these environment variables:
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the forwarder
  attempts to use the secret key from the IAM role.
* This setting is optional.
* No default.

remote_queue.kinesis.auth_region = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The authentication region to use when signing the requests when interacting
  with the remote queue system supporting the Kinesis API.
* If not specified and the forwarder is running on EC2, the auth_region will be
  constructed automatically based on the EC2 region of the instance where the
  the forwarder is running.
* This setting is optional.
* No default.

remote_queue.kinesis.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote queue system supporting the Kinesis API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://kinesis.<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via 'remote_queue.kinesis.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://kinesis.us-west-2.amazonaws.com/
* This setting is optional.
* No default.

remote_queue.kinesis.retry_policy = [max_count|none]
* The retry policy to use for remote queue operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a queue operation will be
    retried upon intermittent failure.
  + "none": Do not retry file operations upon failure.
* This setting is optional.
* Default: "max_count"

remote_queue.kinesis.max_count.max_retries_per_part = <unsigned integer>
* When 'remote_queue.kinesis.retry_policy' is "max_count", sets the
  maximum number of times a queue operation is retried upon intermittent
  failure.
* This setting is optional.
* Default: 9

remote_queue.kinesis.timeout.connect = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The connection timeout, in milliseconds, when interacting with
```

```
  Kinesis for this queue.
* This setting is optional.
* Default: 5000

remote_queue.kinesis.timeout.read = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The read timeout, in milliseconds, when interacting with Kinesis
  for this queue.
* This setting is optional.
* Default: 60000

remote_queue.kinesis.timeout.write = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The write timeout, in milliseconds, when interacting with Kinesis
  for this queue.
* This setting is optional.
* Default: 60000

remote_queue.kinesis.executor_max_workers_count = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum number of worker threads that can be used by
  indexer per pipeline set to execute kinesis queue tasks.
* A value of 0 is equivalent to 1.
* Default: 8

remote_queue.kinesis.max_messages = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default "maximum number of messages" (that are received from
  remote_queue endpoint) to store in kinesis in-memory message queue.
* This setting is optional.
* Default: 10000

remote_queue.kinesis.min_pending_messages = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default "minimum number of pending messages" to use before
  receiving messages off kinesis in-memory message queue.
  Messages are only received when sum of internal queue message count and
  pending object GET (from large messages storage) count is below
  the set value.
* This setting is optional.
* Default: 50

remote_queue.kinesis.max_checkpoints = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default "maximum number of messages" (that have been received from
  remote_queue endpoint and completely consumed) to store in
  the Kinesis in-memory checkpoint queue.
* This setting is optional.
* Default: 100000

remote_queue.kinesis.roll_remote_buckets_interval = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default interval, in seconds, that the Kinesis remote queue
  input worker waits before it rolls the remote storage enabled buckets.
* This setting is optional.
```

```
* Default: 30

remote_queue.kinesis.large_message_store.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote storage system supporting the S3 API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint will be constructed automatically based on the
  auth_region as follows: https://s3-<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via 'remote_queue.kinesis.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://s3-us-west-2.amazonaws.com/
* This setting is optional.
* No default.

remote_queue.kinesis.large_message_store.path = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The remote storage location where messages larger than the
  underlying queue maximum message size will reside.
* The format for this attribute is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string for
    identifying a location inside the storage system.
* These external systems are supported:
   - Object stores that support AWS's S3 protocol. These use the scheme "s3".
     For example, "path=s3://mybucket/some/path".
* If not specified, messages exceeding the underlying queue maximum message
  size are dropped.
* This setting is optional.
* No default.
```

SQS Smartbus specific settings

```
remote_queue.sqs_smartbus.access_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The access key to use when authenticating with the remote queue
  system supporting the SQS API.
* If not specified, the indexer looks for these environment variables:
  'AWS_ACCESS_KEY_ID' or 'AWS_ACCESS_KEY' (in that order). If the environment
  variables are not set and the indexer is running on Elastic Compute Cloud
  (EC2), the indexer attempts to use the secret key from the Identity and
  Access Management (IAM) role.
* This setting is optional.
* No default.

remote_queue.sqs_smartbus.secret_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The secret key to use when authenticating with the remote queue
  system supporting the SQS API.
* If not specified, the indexer looks for these environment variables:
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order). If the environment
  variables are not set and the indexer is running on EC2, the indexer attempts
  to use the secret key from the IAM role.
* This setting is optional.
```

```
* No default.

remote_queue.sqs_smartbus.auth_region = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The authentication region to use when signing the requests when interacting
  with the remote queue system supporting the SQS API.
* If not specified and the indexer is running on EC2, the auth_region is
  constructed automatically based on the EC2 region of the instance where the
  the indexer is running.
* This setting is optional.
* No default.

remote_queue.sqs_smartbus.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote queue system supporting the SQS API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://sqs.<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified in 'remote_queue.sqs.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://sqs.us-west-2.amazonaws.com/
* This setting is optional.
* No default.

remote_queue.sqs_smartbus.max_connections = <unsigned integer>
* Currently not supported. This setting is related to a feature that is still
  under development.
* The maximum number of HTTP connections that can be simultaneously in progress for
  certain queue operations.
* A value of 0 means unlimited.
* Default: 8

remote_queue.sqs_smartbus.message_group_id = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The Message Group ID for Amazon Web Services Simple Queue Service
  (SQS) First-In, First-Out (FIFO) queues.
* Setting a Message Group ID controls how messages within an AWS SQS queue are
  processed.
* For information on SQS FIFO queues and how messages in those queues are
  processed, see "Recommendations for FIFO queues" in the AWS SQS Developer
  Guide.
* If you configure this setting, Splunk software assumes that the SQS queue is
  a FIFO queue, and that messages in the queue should be processed first-in,
  first-out.
* Otherwise, Splunk software assumes that the SQS queue is a standard queue.
* Can be between 1-128 alphanumeric or punctuation characters.
* NOTE: FIFO queues must have Content-Based Deduplication enabled.
* This setting is optional.
* No default.

remote_queue.sqs_smartbus.retry_policy = [max_count|none]
* Currently not supported. This setting is related to a feature that is still
  under development.
* The retry policy to use for remote queue operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
```

```
  + "max_count": Imposes a maximum number of times a queue operation can be
    retried upon intermittent failure.
  + "none": Do not retry file operations upon failure.
* This setting is optional.
* Default: "max_count"

remote_queue.sqs_smartbus.max_count.max_retries_per_part = <unsigned integer>
* Currently not supported. This setting is related to a feature that is still
  under development.
* When 'remote_queue.sqs_smartbus.retry_policy' is set to "max_count", sets the
  maximum number of times a queue operation can be retried upon
  intermittent failure.
* This setting is optional.
* Default: 3

remote_queue.sqs_smartbus.timeout.connect = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The connection timeout, in seconds, when interacting with
  SQS for this queue.
* This setting is optional.
* Default: 5

remote_queue.sqs_smartbus.timeout.read = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The read timeout, in seconds, when interacting with SQS for
  this queue.
* This setting is optional.
* Default: 60

remote_queue.sqs_smartbus.timeout.write = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The write timeout, in seconds, when interacting with SQS for
  this queue.
* This setting is optional.
* Default: 60

remote_queue.sqs_smartbus.timeout.receive_message = <unsigned integer>
* The receive message wait time, in seconds, when interacting with SQS for
  this queue.
* When set to greater than 0, enables "long polling." If there are no messages
  immediately available, the queue waits at most
  'remote_queue.sqs.timeout.receive_message' seconds for a message to
  become available.
* When 0, disables long polling.
* When not set, uses the value configured for the queue via the AWS SQS
  console.
* Maximum value: 20
* This setting is optional.
* Default: 20

remote_queue.sqs_smartbus.timeout.visibility = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default "visibility timeout," in seconds, to use when
  explicitly changing the visibility of specific messages in the queue.
* NOTE: Changing the value of 'remote_queue.sqs.timeout.visibility'
  does not change the implicit visibility timeout configured for
  the queue in the AWS SQS console.
* This setting is optional.
```

```
* Default: 300

remote_queue.sqs_smartbus.buffer.visibility = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default time, in seconds, before
  'remote_queue.sqs.timeout.visibility' at which visibility of
  specific messages in the queue needs to be changed.
* This setting is optional.
* Default: 15

remote_queue.sqs_smartbus.executor_max_workers_count = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum number of worker threads that can be used by
  indexer per pipeline set to execute SQS tasks.
* A value of 0 is equivalent to 1.
* Default: 4

remote_queue.sqs_smartbus.min_pending_messages = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The default "minimum number of pending messages" to use before
  receiving messages off remote queue.
  Messages are only received when the sum of internal queue message count and
  pending object GET (from large messages storage) count is below
  the set value.
* This setting is optional.
* Default: 10

remote_queue.sqs_smartbus.renew_retries = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The number of retries for a particular message on a given indexer after
  being received from the remote queue, before it is proactively moved to the DLQ folder.
* Default: 50

remote_queue.sqs_smartbus.large_message_store.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote storage system supporting the S3 API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://s3-<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via 'remote_queue.sqs_smartbus.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://s3-us-west-2.amazonaws.com/
* This setting is optional.
* No default.

remote_queue.sqs_smartbus.large_message_store.path = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The remote storage location where messages that are larger than the
  underlying queue maximum message size will reside.
* The format for this attribute is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string for
    identifying a location inside the storage system.
* These external systems are supported:
```

```
  - Object stores that support the AWS S3 protocol. These use the scheme "s3".
    For example, "path=s3://mybucket/some/path".
* If not specified, messages exceeding the underlying queue's maximum message
  size are dropped.
* This setting is optional.
* No default.

remote_queue.sqs_smartbus.large_message_store.sslVerifyServerCert = <boolean>
* If set to true, the Splunk platform verifies the certificate presented by the S3
  server and checks that the common name and alternate name match the ones
  specified in 'remote_queue.sqs_smartbus.large_message_store.sslCommonNameToCheck' and
  'remote_queue.sqs_smartbus.large_message_store.sslAltNameToCheck'.
* Default: false

remote_queue.sqs_smartbus.large_message_store.sslVersions = <versions_list>
* Comma-separated list of SSL versions to connect to 'remote.sqs_smartbus.large_message_store.endpoint'.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list
  but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Default: tls1.2

remote_queue.sqs_smartbus.large_message_store.sslCommonNameToCheck = <commonName1>, <commonName2>, ..
* If this value is set, and 'remote_queue.sqs_smartbus.large_message_store.sslVerifyServerCert' is set to
true,
  the Splunk platform instance checks the common name of the certificate presented by
  the remote server (specified in 'remote_queue.sqs_smartbus.large_message_store.endpoint') against this
list
  of common names.
* Default: not set

remote_queue.sqs_smartbus.large_message_store.sslAltNameToCheck = <alternateName1>, <alternateName2>, ..
* If this value is set, and 'remote_queue.sqs_smartbus.large_message_store.sslVerifyServerCert' is set to
true,
  the Splunk platform instance checks the alternate name(s) of the certificate presented by
  the remote server (specified in 'remote_queue.sqs_smartbus.large_message_store.endpoint') against this
list of
  subject alternate names.
* Default: not set

remote_queue.sqs_smartbus.large_message_store.sslRootCAPath = <path>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together. S3 certificate
  will be validated against the CAs present in this file.
* Default: [sslConfig/caCertFile] in server.conf

remote_queue.sqs_smartbus.large_message_store.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Default: TLSv1+HIGH:TLSv1.2+HIGH:@STRENGTH

remote_queue.sqs_smartbus.large_message_store.ecdhCurves = <comma-separated list>
* ECDH curves to use for ECDH key negotiation.
* Specify the curves in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only supports named curves specified
  by their short names.
```

```
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: not set

remote_queue.sqs_smartbus.large_message_store.dhFile = <path>
* PEM format Diffie-Hellman parameter file name.
* DH group size must be no less than 2048bits.
* This file is required in order to enable any Diffie-Hellman ciphers.
* Optional
* Default: not set

remote_queue.sqs_smartbus.dead_letter_queue.name = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The name of the dead letter queue.

remote_queue.sqs_smartbus.dead_letter_queue.process_interval = <number><unit>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The frequency of processing messages that have landed in the dead letter queue.
* Examples: 30s, 6h
* Default: 1d

remote_queue.sqs_smartbus.large_message_store.encryption_scheme = [sse-s3|sse-c|none]
* Currently not supported. This setting is related to a feature that is
  still under development.
* The encryption scheme used by remote storage
* Default: none.

remote_queue.sqs_smartbus.large_message_store.kms_endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The endpoint to connect to for generating KMS keys.
* This setting is required if 'large_message_store.encryption_scheme' is
  set to sse-c.
* Examples: https://kms.us-east-2.amazonaws.com
* No default.

remote_queue.sqs_smartbus.large_message_store.key_id = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The ID for the primary key that KMS uses to generate a data key pair. The
  primary key is stored in AWS.
* This setting is required if 'large_message_store.encryption_scheme' is
  set to sse-c.
* Examples: alias/sqsssekeytrial, 23456789-abcd-1234-11aa-c50f99011223
* No default.

remote_queue.sqs_smartbus.large_message_store.key_refresh_interval = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The time interval to refresh primary key.
* Default: 24h
```

Modular Inputs

```
python.version = [default|python|python2|python3]
```

```
* For Python scripts only, selects which Python version to use.
* Either "default" or "python" select the system-wide default Python version.
* Optional.
* Default: Not set; uses the system-wide Python version.

run_introspection = <boolean>
* Whether or not Splunk software runs introspection on a modular input
  scheme when you have disabled all of its associated scripts by using
  the 'disabled = 1' setting.
* This setting applies only for modular inputs. It takes effect only if you
  specify it under a default stanza of a modular input scheme.
* A default stanza of a modular input scheme begins with the notation
  [<scheme name>]
* If set to "true", Splunk software runs introspection on a modular input
  scheme even when you have disabled all the input scripts for the scheme.
* If set to "false", Splunk software does not run introspection on a modular
  input scheme where you have disabled all scripts for the scheme.
* If introspection does not run for a scheme, then Splunk software does not
  register the modular input scripts that are associated with the scheme
  for execution and it is disabled completely.
* Use the 'disabled' setting to enable or disable individual modular input scripts.
* For example, to turn introspection off for the modular input scheme "myScheme":

  [myScheme]
  run_introspection = false
* Default: true
```

LOGD (logd input for macOS)

```
[logd://<name>]
* This is the macOS logd input component for the Splunk platform.

logd-backtrace = <boolean>
* Whether or not the logd input includes backtraces.
* A value of "true" means that the logd input includes backtraces
  in its events.
* Default: false

logd-debug = <boolean>
* Whether or not the logd input includes "Debug" events.
* A value of "true" means that the logd input includes "Debug" level
  events.
* Default: false

logd-info = <boolean>
* Whether or not the logd input includes "Info" events.
* A value of "true" means that the logd input includes "Info"
  level events.
* Default: true

logd-loss = <boolean>
* Whether or not the logd input includes message loss events.
* A value of "true" means that the logd input includes message loss events.
* Default: false

logd-signpost = <boolean>
* Whether or not the logd input includes signposts.
* A value of "true" means that the logd input includes signpost events.
* Default: false
```

```
logd-predicate = <string>
* Filters messages using the provided predicate, or filter expression,
  that is based on the NSPredicate definition.
* The input supports a single predicate, but the predicate can be a
  compound one.
* Default: none

logd-process = <comma-separated list>
* The process ID on which to operate.
* You can supply multiple process IDs with commas, for example "220,221,223".
* Default: none

logd-source = <boolean>
* Whether or not to include symbol names and source line numbers for
  messages, if available.
* Default: false

logd-include-fields = <comma-separated list>
* The fields to retrieve from a logD record.
* Default:  all

logd-exclude-fields = <comma-separated list>
* The fields to ignore when parsing a logD record
* Example setting: logd-exclude-fields = bootUUID,formatString
* Default: formatString,timestamp,timezoneName

logd-interval = <unsigned integer>
* How often, in seconds, that the input is to query logd for events,
* Default: 30

logd-starttime = <string>
* The earliest acceptable time for the input to query logd for events.
* Use the format "YYYY-MM-DD HH:MM:SS" to specify the timestamp.
* No default.

logd-freetext = <string>
* reserved for future use
```

JOURNALD (journald input for Linux)

```
[journald://<name>]
* This is the systemd-journald input component for Splunk

journalctl-include-fields = <string>
* This setting and the "journalctl-exclude-fields" setting control the fields
  that the journald input retrieves.
* The input selects most of the fields if they are in
  "one of "journalctl-include-fields" and not in 'journalctl-exclude-fields'.
* The exceptions are MESSAGE, CURSOR, and _REALTIME_TIMESTAMP. The system
  treats these fields specially.
* An empty 'journalctl-include-fields' value means to output all fields.
* If you want all fields except XYZ, leave 'journalctl-include-fields' empty,
  and set journalctl-include-fields empty, and set
  journalctl-exclude-fields=XYZ
* The input always retrieves the MESSAGE, __REALTIME_TIMESTAMP, and __CURSOR
  fields, but uses the __REALTIME_TIMESTAMP and __CURSOR fields internally and
  does not send them to the Splunk platform.
* Fields __MONOTONIC_TIMESTAMP and __SOURCE_REALTIME_TIMESTAMP should always
```

```
  be suppressed to decrease cardinality of data. Use Splunk event time instead.
* Default: PRIORITY,_SYSTEMD_UNIT,_SYSTEMD_CGROUP,_TRANSPORT,_PID,_UID,
  _MACHINE_ID,_GID,_COMM,_EXE


journalctl-exclude-fields = <comma-separated list>
* The fields to exclude. use this setting to filter which fields
  to send to the Splunk platform.
* This filter is more computationally expensive than journalctl-output-fields,
  as it is not natively supported by API and requires post-processing
* Default: __MONOTONIC_TIMESTAMP,__SOURCE_REALTIME_TIMESTAMP

journalctl-filter = <string>
* These settings map directly to the arguments for the journalctl command.
  See the documentation for journalctl.
* Default: none

journalctl-unit = <string>
* Equivalent to '-u' parameter of journalctl; show messages for the
  specified systemd unit
* Default: none

journalctl-identifier = <string>
* Equivalent to '-t' parameter of journalctl; show messages for the
  specified syslog identifier SYSLOG_IDENTIFIER
* Default: none

journalctl-priority = <string>
* equivalent to '-p' parameter of journalctl; filter output by message
  priorities or priority ranges.
* Default: 7

journalctl-boot = <string>
* Equivalent to '-b' parameter of journalctl; messages from a specific boot
* Default: none

journalctl-facility = <string>
* Equivalent to '--facility' parameter of journalctl, syslog facility
* Default: none

journalctl-grep = <string>
* Equivalent to '-g' parameter of journalctl; filter output to entries
  where the MESSAGE= field matches the specified regular expression.
  PERL-compatible regular expressions are used
* Default: none

journalctl-user-unit = <string>
* Equivalent to '--user-unit' parameter of journalctl; show messages
  for the specified user session unit.
* Default: none

journalctl-dmesg = <boolean>
* Equivalent to '-k' parameter of journalctl; show only kernel messages.
* Default:  false

journalctl-quiet = <boolean>
* Equivalent to '-q' parameter of journalctl; suppress all informational
  messages
* Default: false

journalctl-freetext = <string>
* reserved for future use
```

inputs.conf.example

```
#   Version 9.2.2
#
# This is an example inputs.conf. Use this file to configure data inputs.
#
# To use one or more of these configurations, copy the configuration block into
# inputs.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# The following configuration reads all the files in the directory /var/log.

[monitor:///var/log]


# The following configuration reads all the files under /var/log/httpd and
# classifies them as sourcetype::access_common.
#
# When checking a file for new data, if the file's modification time is from
# before seven days ago, the file will no longer be checked for changes
# until you restart the software.

[monitor:///var/log/httpd]
sourcetype = access_common
ignoreOlderThan = 7d


# The following configuration reads all the
# files under /mnt/logs. When the path is /mnt/logs/<host>/... it
# sets the hostname (by file) to <host>.

[monitor:///mnt/logs]
host_segment = 3


# The following configuration listens on TCP port 9997 for raw
# data from ANY remote server (not just a Splunk instance). The host of the
# data is set to the IP address of the remote server.

[tcp://:9997]


# The following configuration listens on TCP port 9995 for raw
# data from ANY remote server. The host of the data is set as the host name of
# the remote server.  All data will also be assigned the sourcetype "log4j" and
# the source "tcp:9995".

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

```
# The following configuration listens on TCP port 9995 for raw
# data from 10.1.1.10.
# All data is assigned the host "webhead-1", the sourcetype "access_common" and
# the the source "//10.1.1.10/var/log/apache/access.log".

[tcp://192.0.2.10:9995]
host = webhead-1
sourcetype = access_common
source = //192.0.2.10/var/log/apache/access.log


# The following configuration listens on TCP port 9996 for
# Splunk cooked event data from ANY splunk forwarder.
# The host of the data is set to the host name of the remote server ONLY IF the
# remote data has no host set, or if it is set to "localhost".

[splunktcp://:9996]
connection_host = dns


# The following configuration listens on TCP port 9996 for
# distributed search data from 10.1.1.100. The data is processed the same as
# locally indexed data.

[splunktcp://192.0.2.100:9996]


# The following configuration listens on TCP port 514 for data
# from syslog.corp.company.net. The data is assigned the sourcetype "syslog"
# and the host is set to the host name of the remote server.

[tcp://syslog.corp.example.net:514]
sourcetype = syslog
connection_host = dns

# Following configuration limits the acceptance of data to forwarders
# that have been configured with the token value specified in 'token' field.
# NOTE: The token value is encrypted. The REST endpoint encrypts the token
# while saving it.

[splunktcptoken://tok1]
token = $7$ifQTPTzHD/BA8VgKvVcgO1KQAtr3N1C8S/1uK3nAKIE9dd9e9g==

# Set up Secure Sockets Layer (SSL):

[SSL]
serverCert=$SPLUNK_HOME/etc/auth/server.pem
password=password
requireClientCert=false

[splunktcp-ssl:9996]

# Use file system change monitor:

[fschange:/etc/]
fullEvent=true
pollPeriod=60
recurse=true
sendEventMaxSize=100000
index=main
```

```
# Monitor the Security Windows Event Log channel, getting the most recent
# events first, then older, and finally continuing to gather newly arriving events

[WinEventLog://Security]
disabled = 0
start_from = newest
evt_dc_name =
evt_dns_name =
evt_resolve_ad_ds =
evt_resolve_ad_obj = 1
checkpointInterval = 5

# Monitor the ForwardedEvents Windows Event Log channel, only gathering the
# events that arrive after monitoring starts, going forward in time.

[WinEventLog://ForwardedEvents]
disabled = 0
start_from = oldest
current_only = 1
batch_size = 10
checkpointInterval = 5

[tcp://9994]
queueSize=50KB
persistentQueueSize=100MB

# Perfmon: Windows performance monitoring examples

# You must specify the names of objects, counters and instances
# exactly as they are shown in the Performance Monitor application. Splunk Web
# is the recommended interface to use to configure performance monitor inputs.

# These stanzas gather performance data from the local system only.
# Use wmi.conf for performance monitor metrics on remote systems.

# 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.

[perfmon://LocalPhysicalDisk]
interval = 10
object = PhysicalDisk
counters = Disk Bytes/sec; % Disk Read Time; % Disk Write Time; % Disk Time
instances = *
disabled = 0
index = PerfMon

# Gather common memory statistics using the Memory performance object, every
# 5 seconds. Store the data in the "main" index. Since none of the counters
# specified have applicable instances, the instances attribute is not required.

[perfmon://LocalMainMemory]
interval = 5
object = Memory
counters = Committed Bytes; Available Bytes; % Committed Bytes In Use
disabled = 0
index = main

# Gather data on USB activity levels every 10 seconds. Store this data in the
# default index.

[perfmon://USBChanges]
```

```
interval = 10
object = USB
counters = Usb Control Data Bytes/Sec
instances = *
disabled = 0

# Admon: Windows Active Directory monitoring examples

# Monitor the default domain controller (DC) for the domain that the computer
# running Splunk belongs to. Start monitoring at the root node of Active
# Directory.
[admon://NearestDC]
targetDc =
startingNode =

# Monitor a specific DC, with a specific starting node. Store the events in
# the "admon" Splunk index. Do not print Active Directory schema. Do not
# index baseline events.

[admon://DefaultTargetDC]
targetDc = pri01.eng.ad.splunk.com
startingNode = OU=Computers,DC=eng,DC=ad,DC=splunk,DC=com
index = admon
printSchema = 0
baseline = 0

# Monitor two different DCs with different starting nodes.
[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=hr,DC=ad,DC=splunk,DC=com

# logD
[logd://example]
logd-backtrace = false
logd-debug = false
logd-info = false
logd-loss = false
logd-signpost = false
logd-predicate = 'subsystem == "com.apple.TimeMachine" && eventMessage CONTAINS[c] "backup"'
logd-process = 220,221,223
logd-source = false
journalctl-include-fields = PRIORITY,CMD,EXE
logd-exclude-fields = bootUUID,formatString
logd-interval = 60
logd-starttime = "2015-01-10 17:15:00"

#journald
[journald://example]
journalctl-include-fields = MESSAGE
journalctl-exclude-fields = _UID,_MACHINE_ID,_GID,_COMM,_EXE
journalctl-filter = _SYSTEMD_UNIT=avahi-daemon.service _PID=28097 +  _SYSTEMD_UNIT=dbus.service
journalctl-unit = systemd-modules-load.service
journalctl-identifier = SYSLOG_IDENTIFIER
journalctl-priority = 0
journalctl-boot = 2
journalctl-facility = help
journalctl-grep =^WARN.*disk,.*errno=\d+\S+restarting
journalctl-user-unit = SERVICENAME
```

```
journalctl-dmesg = true
journalctl-quiet = true
```

instance.cfg.conf

The following are the spec and example files for `instance.cfg.conf`.

instance.cfg.conf.spec

```
#   Version 9.2.2
#
# This file contains the set of attributes and values you can expect to find in
# the SPLUNK_HOME/etc/instance.cfg file; the instance.cfg file is not to be
# modified or removed by user.  LEAVE THE instance.cfg FILE ALONE.
#

#
```

GLOBAL SETTINGS

```
# The [general] stanza defines global settings.
#
```

[general]

```
guid = <GUID in all-uppercase>
* This setting formerly (before 5.0) belonged in the [general] stanza of
  server.conf file.

* Splunk expects that every Splunk instance will have a unique string for this
  value, independent of all other Splunk instances.  By default, Splunk will
  arrange for this without user intervention.

* Currently used by (not exhaustive):
  * Clustering environments, to identify participating nodes.
  * Splunk introspective searches (Splunk on Splunk, Deployment Monitor,
    etc.), to identify forwarders.

* At startup, the following happens:

  * If server.conf has a value of 'guid' AND instance.cfg has no value of
    'guid', then the value will be erased from server.conf and moved to
    instance.cfg file.

  * If server.conf has a value of 'guid' AND instance.cfg has a value of
    'guid' AND these values are the same, the value is erased from
    server.conf file.

  * If server.conf has a value of 'guid' AND instance.cfg has a value of 'guid'
    AND these values are different, startup halts and error is shown.  Operator
    must resolve this error.  We recommend erasing the value from server.conf
    file, and then restarting.

  * If you are hitting this error while trying to mass-clone Splunk installs,
    please look into the command 'splunk clone-prep-clear-config';
```

```
    'splunk help' has help.

* See http://www.ietf.org/rfc/rfc4122.txt for how a GUID (a.k.a. UUID) is
  constructed.

* The standard regexp to match an all-uppercase GUID is
  "[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}".
```

instance.cfg.conf.example

```
#   Version 9.2.2
#
# This file contains an example SPLUNK_HOME/etc/instance.cfg file; the
# instance.cfg file is not to be modified or removed by user.  LEAVE THE
# instance.cfg FILE ALONE.
#

[general]
guid = B58A86D9-DF3D-4BF8-A426-DB85C231B699
```

limits.conf

The following are the spec and example files for `limits.conf`.

limits.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure limitations for the search commands.
#
# Each stanza controls different search commands settings.
#
# There is a limits.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name limits.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see limits.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# About Distributed Search
#   Unlike most settings which affect searches, limits.conf settings are not
#   provided by the search head to be used by the search peers.  This means
```

```
#   that if you need to alter search-affecting limits in a distributed
#   environment, typically you will need to modify these settings on the
#   relevant peers and search head for consistent results.
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file takes precedence.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
#
# CAUTION: Do not alter the settings in the limits.conf file unless you know
#     what you are doing. Improperly configured limits might result in
#     splunkd crashes, memory overuse, or both.
```

[default]

```
DelayArchiveProcessorShutdown = <boolean>
* Specifies whether during splunk shutdown archive processor should finish
  processing archive file under process.
* When set to "false": The archive processor abandons further processing of
  the archive file and will process again from start again.
* When set to "true": The archive processor will complete processing of
  the archive file. Shutdown will be delayed.
* Default: false

maxArchiveNestLevel = <non-negative integer>
* Specifies the maximum number of nested levels for an archive file for processing.
* If an archive file exceeds the maximum specified nested level, Splunk software ignores the archive file.
* Maximum value: 10
* Default: 4

max_mem_usage_mb = <non-negative integer>
* Provides a limitation to the amount of RAM, in megabytes (MB), a batch of
  events or results will use in the memory of a search process.
* Operates on an estimation of memory use which is not exact. The estimation can
  deviate by an order of magnitude or so to both the smaller and larger sides.
* The limitation is applied in an unusual way; if the number of results or
  events exceeds maxresultrows, AND the estimated memory exceeds this limit, the
  data is spilled to disk.
* This means, as a general rule, lower limits will cause a search to use more
  disk I/O and less RAM, and be somewhat slower, but should cause the same
  results to typically come out of the search in the end.
* This limit is applied currently to a number, but not all search processors.
  However, more will likely be added as it proves necessary.
* The number is thus effectively a ceiling on batch size for many components of
  search for all searches run on this system.
* When set to "0": Specifies that the size is unbounded. Searches might be
  allowed to grow to arbitrary sizes.
* NOTE:
  * The mvexpand command uses the 'max_mem_usage_mb' value in a different way.
    * The mvexpand command has no combined logic with 'maxresultrows'.
```

```
    * If the memory limit is exceeded, output is truncated, not spilled to disk.
  * The 'stats' and 'sdselect' command processors use the 'max_mem_usage_mb'
    value in the following way.
    * If the estimated memory usage exceeds the specified limit, the results
      are cached to the disk. This means that when a large volume of data
      exceeds the 'max_mem_usage_mb' setting, the search processor doesn't
      store all the data in memory. Instead, the search processor puts some
      data into temporary data files on disk, so that it can do further
      processing on that data later as needed.
    * If 0 is specified, the results are cached to the disk when the number of
      results exceeds the 'maxresultrows' setting.
  * The eventstats command processor uses the 'max_mem_usage_mb' value in the
    following way.
    * Both the 'max_mem_usage_mb' and the 'maxresultrows' settings are used to
      determine the maximum number of results to return.  If the limit for one
      setting is reached, the eventstats processor continues to return results
      until the limit for the other setting is reached. When both limits are
      reached, the eventstats command processor stops adding the requested
      fields to the search results.
    * If you set 'max_mem_usage_mb' to 0, the eventstats command processor uses
      only the 'maxresultrows' setting as the threshold. When the number of
      results exceeds the 'maxresultrows' setting, the eventstats command
      processor stops adding the requested fields to the search results.
* Default: 200

min_batch_size_bytes = <integer>
* Specifies the size, in bytes, of the file/tar after which the
  file is handled by the batch reader instead of the trailing processor.
* Global setting, cannot be configured per input.
* NOTE: Configuring this to a very small value could lead to backing up of jobs
  at the tailing processor.
* Default: 20971520

regex_cpu_profiling = <boolean>
* Enable CPU time metrics for RegexProcessor. Output will be in the
  metrics.log file.
  Entries in metrics.log will appear per_host_regex_cpu, per_source_regex_cpu,
  per_sourcetype_regex_cpu, per_index_regex_cpu.
* Default: true

agg_cpu_profiling = <boolean>
* Enable CPU time metrics for AggregatorProcessor. Output will be in the
  metrics.log file.
  Entries in metrics.log will appear per_host_agg_cpu, per_source_agg_cpu,
  per_sourcetype_agg_cpu, per_index_agg_cpu.
* Default: true

msp_cpu_profiling = <boolean>
* Enable CPU time metrics for MetricSchemaProcessor. Output will be in the
  metrics.log file.
  Entries in metrics.log will appear per_host_msp_cpu, per_source_msp_cpu,
  per_sourcetype_msp_cpu, per_index_msp_cpu.
* Default: true

mp_cpu_profiling = <boolean>
* Enable CPU time metrics for MetricsProcessor. Output will be in the
  metrics.log file.
  Entries in metrics.log will appear per_host_mp_cpu, per_source_mp_cpu,
  per_sourcetype_mp_cpu, per_index_mp_cpu.
* Default: true

lb_cpu_profiling = <boolean>
```

```
* Enable CPU time metrics for LineBreakingProcessor. Output will be in the
  metrics.log file.
  Entries in metrics.log will appear per_host_lb_cpu, per_source_lb_cpu,
  per_sourcetype_lb_cpu, per_index_lb_cpu.
* Default: true

clb_cpu_profiling = <boolean>
* Enable CPU time metrics for ChunkedLBProcessor. Output will be in the
  metrics.log file.
  Entries in metrics.log will appear per_host_clb_cpu, per_source_clb_cpu,
  per_sourcetype_clb_cpu, per_index_clb_cpu.
* Default: false

file_and_directory_eliminator_reaper_interval = <integer>
* Specifies how often, in seconds, to run the FileAndDirectoryEliminator reaping
  process.
* The FileAndDirectoryEliminator eliminates files and directories by moving them
  to a location that is reaped periodically. This reduces the chance of
  encountering issues due to files being in use.
* On Windows, the FileAndDirectoryEliminator is used by the deployment client
  to delete apps that have been removed or that are being redeployed.
* A value of 0 disables the FileAndDirectoryEliminator.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default (on Windows): 60
* Default (otherwise): 0

interval = <integer>
* Number of seconds between logging splunkd metrics to metrics.log
  for different subgroups.
* Check metrics.log for the list of configurable "metrics_modules".
* Set "interval" under the desired "metrics_module" stanza.
* Example:
  * If you want 60 seconds metrics logging interval for "thruput:thruput",
    * [thruput:thruput]
    * interval = 60
* Minimum value is 10 seconds.
* Valid value is multiple of 10.
* If value is not exact multiple of 10, it will be adjusted to nearest
  downward multiple.
* Recommended value multiple of 30. Splunk will decide how often to check for
  metrics reporting based on greatest common divisor across different values.
  If "interval" is set 30, 40 for two different components, then
  greatest common divisor for 30, 40 and 60(default) is 10. It's expensive
  for metrics reporting thread to log every 10 sec.
  If "interval" is set 30, 900 for two different components, then
  greatest common divisor for 30, 90 and 60(default) is 30. It's less
  expensive for metrics reporting thread to log every 30 sec.
* Default : "interval" config value set under [metrics] stanza.
```

[searchresults]

```
* This stanza controls search results for a variety of Splunk search commands.

compression_level = <integer>
* Compression level to use when writing search results to .csv.gz files.
* Default: 1

maxresultrows = <integer>
* Configures the maximum number of events generated by search commands
  which grow the size of your result set (such as multikv) or that create
  events. Other search commands are explicitly controlled in specific stanzas
```

```
  that follow.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 50000

tocsv_maxretry = <integer>
* Maximum number of times to retry the atomic write operation.
* When set to "1": Specifies that there will be no retries.
* Default: 5

tocsv_retryperiod_ms = <integer>
* Period of time to wait before each retry.
* Default: 500

* These setting control logging of error messages to the info.csv file.
  All messages will be logged to the search.log file regardless of
  these settings.
```

[search_info]

```
* This stanza controls logging of messages to the info.csv file.
* Messages logged to the info.csv file are available to REST API clients
  and Splunk Web. Limiting the messages added to info.csv will mean
  that these messages will not be available in the UI and/or the REST API.

filteredindexes_log_level = [DEBUG|INFO|WARN|ERROR]
* Log level of messages when search returns no results because
  user has no permissions to search on queried indexes.
* Default: DEBUG

infocsv_log_level = [DEBUG|INFO|WARN|ERROR]
* Limits the messages which are added to the info.csv file to the stated
  level and above.
* For example, if "infocsv_log_level" is WARN, messages of type WARN
  and higher will be added to the info.csv file.
* Default: INFO

max_infocsv_messages  = <positive integer>
* Limits the number of messages which are added to the info.csv file,
  per log level.
* If more than max_infocsv_messages log entries are generated, additional
  entries will not be logged in the info.csv file. All entries will still be
  logged in the search.log file.
* Default: 20

show_warn_on_filtered_indexes = <boolean>
* Log warnings if search returns no results because user has
  no permissions to search on queried indexes.
* Default: false
```

[subsearch]

```
* This stanza controls subsearch results.
* NOTE: This stanza DOES NOT control subsearch results when a subsearch is
  called by commands such as join, append, or appendcols.
* Read more about subsearches in the online documentation:
  http://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutsubsearches


maxout = <integer>
```

```
* Maximum number of results to return from a subsearch.
* This value cannot be greater than or equal to 10500.
* Default: 10000

maxtime = <integer>
* Maximum number of seconds to run a subsearch before finalizing
* Default: 60

ttl = <integer>
* The time to live (ttl), in seconds, of the cache for the results of a given
  subsearch.
* Do not set this below 120 seconds.
* See the definition in the [search] stanza under the "TTL" section for more
  details on how the ttl is computed.
* Default: 300 (5 minutes)

subsearch_artifacts_delete_policy = [immediate|ttl]
* How subsearch artifacts are deleted after a sub search completes.
* Set to `immediate` to have subsearch artifacts remove immediately after a
  subsearch completes.
* Set to 'ttl' to have subsearch artifacts delete after the time-to-live of
  the subsearch has been reached.
* For example, you could use '|noop subsearch_artifacts_delete_policy = [immediate|ttl]'
  to overwrite the setting for a particular search.
* Default: ttl
```

SEARCH COMMAND

```
# This section contains the limitation settings for the search command.
# The settings are organized by type of setting.
```

[search]

```
# The settings under the [search] stanza are organized by type of setting.
```

Batch search

```
# This section contains settings for batch search.

allow_batch_mode = <boolean>
* Specifies whether or not to allow the use of batch mode which searches
  in disk based batches in a time insensitive manner.
* In distributed search environments, this setting is used on the search head.
* Default: true

batch_search_max_index_values = <integer>
* When using batch mode, this limits the number of event entries read from the
  index file. These entries are small, approximately 72 bytes. However batch
  mode is more efficient when it can read more entries at one time.
* Setting this value to a smaller number can lead to slower search performance.
* A balance needs to be struck between more efficient searching in batch mode
* and running out of memory on the system with concurrently running searches.
* Default: 10000000
```

```
batch_search_max_pipeline = <integer>
* This setting controls the number of search pipelines that are launched on the
  indexer during batch search.
* Increasing the number of search pipelines can improve search performance.
  However, this can also result in increased thread and memory usage.
* This setting applies only to searches that run on remote indexers.
* The value for this setting should be >=1. When this setting is >1 on the
  search head, the setting is applied to all remote indexers. Otherwise, remote
  indexers use their local 'batch_search_max_pipeline' setting.
* You can override this setting on a per-search basis by appending
  '|noop batch_search_max_pipeline=<integer>' to the search string. The
  <integer> should be >1.
* Default: 1

batch_search_max_results_aggregator_queue_size = <integer>
* Controls the size, in bytes, of the search results queue to which all
  the search pipelines dump the processed search results.
* Increasing the size can lead to search performance gains.
  Decreasing the size can reduce search performance.
* Do not specify zero for this setting.
* Default: 100000000

batch_search_max_serialized_results_queue_size = <integer>
* Controls the size, in bytes, of the serialized results queue from which
  the serialized search results are transmitted.
* Increasing the size can lead to search performance gains.
  Decreasing the size can reduce search performance.
* Do not specify zero for this setting.
* Default: 100000000

NOTE: The following batch search settings control the periodicity of retries
      to search peers in the event of failure (Connection errors, and others).
      The interval exists between failure and first retry, as well as
      successive retries in the event of further failures.

batch_retry_min_interval = <integer>
* When batch mode attempts to retry the search on a peer that failed,
  specifies the minimum time, in seconds, to wait to retry the search.
* Default: 5

batch_retry_max_interval = <integer>
* When batch mode attempts to retry the search on a peer that failed,
  specifies the maximum time, in seconds, to wait to retry the search.
* Default: 300 (5 minutes)

batch_retry_scaling = <double>
* After a batch retry attempt fails, uses this scaling factor to increase
  the time to wait before trying the search again.
* The value should be > 1.0.
* Default: 1.5
```

Bundles

```
# This section contains settings for bundles and bundle replication.

load_remote_bundles = <boolean>
* On a search peer, allow remote (search head) bundles to be loaded in splunkd.
* Default: false.

replication_file_ttl = <integer>
```

```
* The time to live (ttl), in seconds, of bundle replication tarballs,
  for example: *.bundle files.
* Default: 600 (10 minutes)

replication_period_sec  = <integer>
* The minimum amount of time, in seconds, between two successive bundle
  replications.
* Default: 60

sync_bundle_replication = [0|1|auto]
* Indicates whether configuration file replication blocks
  searches or is run asynchronously.
* When set to "auto": The Splunk software uses asynchronous
  replication only if all of the peers support asynchronous bundle
  replication.
  Otherwise synchronous replication is used.
* Default: auto

bundle_status_expiry_time = <interval>
* The amount of time the search head waits before purging the status of a knowledge bundle
  push request to the indexer.
* The status is purged either when it is not queried for a period greater than
  this setting or when its associated bundle is deleted by the reaper.
* The interval can be specified as a string for minutes, seconds, hours, days.
  For example; 60s, 1m, 1h, 1d etc.
* Default: 1h
```

Concurrency

```
# This section contains settings for search concurrency limits.

total_search_concurrency_limit = auto | <integer>
* Specifies the maximum search concurrency limit for a search head cluster
  or standalone search head.
* When set to "auto", the search head cluster or standalone search head
  calculates the historical search concurrency limit using
  max_hist_searches = max_searches_per_cpu x number_of_cpus + base_max_searches.
* The real-time search concurrency limit is calculated based on the historical search
  concurrency limit.
* When set to an integer, the setting specifies the maximum search concurrency limit.
  For a search head cluster, the number specifies the maximum search limit across
  the cluster. For a standalone search head, the number specifies the maximum
  search limit for the search head. The value must be in the range of 1 to 8192.
* Default: auto

base_max_searches = <integer>
* A constant to add to the maximum number of searches, computed as a
  multiplier of the CPUs.
* Default: 6

max_rt_search_multiplier = <decimal number>
* A number by which the maximum number of historical searches is multiplied
  to determine the maximum number of concurrent real-time searches.
* NOTE: The maximum number of real-time searches is computed as:
  max_rt_searches = max_rt_search_multiplier x max_hist_searches
* Default: 1

max_searches_per_cpu = <integer>
* The maximum number of concurrent historical searches for each CPU.
  The system-wide limit of historical searches is computed as:
```

```
  max_hist_searches =  max_searches_per_cpu x number_of_cpus + base_max_searches
* NOTE: The maximum number of real-time searches is computed as:
  max_rt_searches = max_rt_search_multiplier x max_hist_searches
* Default: 1

shc_adhoc_quota_enforcement = on | off | overflow
* Determines the way in which the cluster enforces limits on the number of concurrent searches.
  Since concurrent searches include both scheduled and ad hoc searches, this setting effectively
  determines the  enforcement method for admitting new ad hoc searches.
* "on" means the ad hoc search admission process is managed cluster-wide by the captain.
* "off" means the ad hoc search admission process is managed locally, by each
   search head that receives an ad hoc search request.
* "overflow" means the local search head checks its local capacity first
  when admitting an ad hoc search. If the search head has capacity (that is,
  if the search head is below the local limit on number of concurrent searches),
  it runs the search locally. If the search head has reached its limit on concurrent
  searches, it defers to the captain for permission to run the search. The captain will
  check which search head has the capacity, and tell the local search head to proxy the search
  to the remote search head to run it.
* Default: off
```

Distributed search

```
# This section contains settings for distributed search connection
# information.

addpeer_skew_limit = <positive integer>
* Absolute value of the largest time skew, in seconds, that is allowed when
  configuring a search peer from a search head, independent of time.
* If the difference in time (skew) between the search head and the peer is
  greater than "addpeer_skew_limit", the search peer is not added.
* This is only relevant to manually added peers. This setting has no effect
  on index cluster search peers.
* Default: 600 (10 minutes)

fetch_remote_search_log = [enabled|disabledSavedSearches|disabled]
* When set to "enabled": All remote search logs are downloaded barring
  the oneshot search.
* When set to "disabledSavedSearches": Downloads all remote logs other
  than saved search logs and oneshot search logs.
* When set to "disabled": Irrespective of the search type, all remote
  search log download functionality is disabled.
* NOTE:
  * The previous Boolean values:[true|false] are still
    supported, but are not recommended.
  * The previous value of "true" maps to the current value of "enabled".
  * The previous value of "false" maps to the current value of "disabled".
* Default: disabledSavedSearches

max_chunk_queue_size = <integer>
* The maximum size of the chunk queue.
* default: 10000000

max_combiner_memevents = <integer>
* Maximum size of the in-memory buffer for the search results combiner.
  The <integer> is the number of events.
* Default: 50000

max_tolerable_skew = <positive integer>
* Absolute value of the largest time skew, in seconds, that is tolerated
```

```
  between the native clock on the search head and the native clock on the peer
  (independent of time zone).
* If this time skew is exceeded, a warning is logged. This estimate is
  approximate and tries to account for network delays.
* Default: 60

max_workers_searchparser = <integer>
* The number of worker threads in processing search result when using round
  robin policy.
* default: 5

results_queue_min_size = <integer>
* The minimum size, of search result chunks, that will be kept from peers
  for processing on the search head before throttling the rate that data
  is accepted.
* The minimum queue size in chunks is the "results_queue_min_size" value
  and the number of peers providing results, which ever is greater.
* Default: 10

result_queue_max_size = <integer>
* The maximum size, in bytes, that will be kept from peers for processing on
  the search head before throttling the rate that data is accepted.
* The "results_queue_min_size" value takes precedence. The number of search
  results chunks specified by "results_queue_min_size" will always be
  retained in the queue even if the combined size in MB exceeds the
  "result_queue_max_size" value.
* Default: 100

results_queue_read_timeout_sec = <integer>
* The amount of time, in seconds, to wait when the search executing on the
  search head has not received new results from any of the peers.
* Cannot be less than the 'receiveTimeout' setting in the distsearch.conf
  file.
* Default: 900

batch_wait_after_end = <integer>
* DEPRECATED: Use the 'results_queue_read_timeout_sec' setting instead.

remote_search_requests_throttling_type = disabled | per_cpu | physical_ram
* Sets the way remote searches are throttled on remote peers. Search request
  that is throttled is rejected with 429 HTTP code.
* "disabled" simply disables any throttling.
* "per_cpu" sets the throttling based on available CPU number.
* "physical_ram" sets the throttling based on available system memory.
* Multiple, comma-separated, throttling types can be set. For example:
  'remote_search_requests_throttling_type = per_cpu, physical_ram'
  enables both "per_cpu" and "physical_ram".
* Does not apply to real-time searches.
* Do not use this feature in conjunction with workload management.
* Default: disabled

remote_search_requests_send_capabilities_list = <boolean>
* When turned on, the search head sends the list of all capabilities of the
  user running the search to every search peer participating in the search.
* This makes it possible to uniformly enforce user-level role-based access
  control (RBAC).
* Default: false

remote_search_requests_reject_if_capabilities_list_absent = <boolean>
* When turned on for a search peer, the search peer rejects search requests that
  do not also specify the full capability list for the user running the search.
* The search head sends the full capability list for users running the
```

```
  search when 'send_capabilities_list_to_indexer' is set to true.
* Turn this on only if all search heads have already set
  'send_capabilities_list_to_indexers' to true.
* Default: false
```

Field stats

```
# This section contains settings for field statistics.

fieldstats_update_freq = <number>
* How often to update the field summary statistics, as a ratio to the elapsed
  run time so far.
* Smaller values means update more frequently.
* When set to "0": Specifies to update as frequently as possible.
* Default: 0

fieldstats_update_maxperiod = <number>
* The maximum period, in seconds, for updating field summary statistics.
* When set to "0": Specifies that there is not maximum period. The period
  is dictated by the calculation:
  current_run_time x fieldstats_update_freq
* Fractional seconds are allowed.
* Default: 60

min_freq = <number>
* Minimum frequency of a field that is required for the field to be included
  in the /summary endpoint.
* The frequency must be a fraction >=0 and <=1.
* Default: 0.01 (1%)
```

History

```
# This section contains settings for search history.

enable_history = <boolean>
* Specifies whether to keep a history of the searches that are run.
* Default: true

max_history_length = <integer>
* Maximum number of searches to store in history for each user and application.
* When 'search_history_storage_mode' has a value of "kvstore", this value is
  applicable per user only, and not per user and application combination.
* Default: 500

max_history_storage_retention_time = <integer>[s|m|h|d]
* The maximum time to store search history records for each user and
  application.
* This setting and the 'max_history_length' setting determine how many search
  history records appear in persistent storage.
* Search stops storing search history records when either the retention time or
  the number of search history records exceeds the values you configure with
  these settings.
* A value of 0 means that search only uses 'max_history_length' to retain
  search history to persistent storage.
* The time units you can specify for this setting are:
  s, sec, second, secs, seconds, m, min, minute, mins, minutes,
  h, hr, hour, hrs, hours, d, day, days.
```

```
* This setting is only applicable when 'search_history_storage_mode' has
  a value of "kvstore".
* Default: 90d

search_history_storage_mode = <string>
* The storage mode by which a search head cluster saves search history.
* Valid storage modes include "csv" and "kvstore".
* This setting is valid only when the 'enable_history' setting has a value of "true".
* A value of "kvstore" means that the cluster can replicate search history across
  all its members using the App Key Value Store service.
* A value of "csv" means that search history is saved to CSV files only on the
  local search head.
* When you initially give this setting a value of "kvstore", the search head
  migrates the existing search history records, if they are present in existing
  CSV files, into the App Key Value Store service. This migration of search
  history can only happen once. If you later change the storage mode to "csv",
  then back to "kvstore", subsequent migrations do not occur.
* NOTE: In the "kvstore" storage mode, the 'max_history_length' is the maximum
  number of searches that the SHC can store for each user. In this case, the
  maximum acceptable value of 'max_history_length' cannot exceed 1000.
* Default: csv
```

Memory tracker

```
# This section contains settings for the memory tracker.

enable_memory_tracker = <boolean>
* Specifies if the memory tracker is enabled.
* When set to "false" (disabled): The search is not terminated even if
  the search exceeds the memory limit.
* When set to "true": Enables the memory tracker.
* Must be set to "true" to enable the "search_process_memory_usage_threshold"
  setting or the "search_process_memory_usage_percentage_threshold" setting.
* Default: false

search_process_memory_usage_threshold = <double>
* To use this setting, the "enable_memory_tracker" setting must be set
  to "true".
* Specifies the maximum memory, in MB, that the search process can consume
  in RAM.
* Search processes that violate the threshold are terminated.
* If the value is set to 0, then search processes are allowed to grow
  unbounded in terms of in memory usage.
* Default: 4000 (4GB)

search_process_memory_usage_percentage_threshold = <decimal>
* To use this setting, the 'enable_memory_tracker' setting must be set
  to "true".
* Specifies the percent of the total memory that the search process is
  entitled to consume.
* Search processes that violate the threshold percentage are terminated.
* If the value is set to zero, then splunk search processes are allowed to
  grow unbounded in terms of percentage memory usage.
* Any setting larger than 100 or less than 0 is discarded and the default
  value is used.
* Default: 25%
```

Meta search

```
# This section contains settings for meta search.

allow_inexact_metasearch = <boolean>
* Specifies if a metasearch that is inexact be allowed.
* When set to "true": An INFO message is added to the inexact metasearches.
* When set to "false": A fatal exception occurs at search parsing time.
* Default: false

indexed_as_exact_metasearch = <boolean>
* Specifies if a metasearch can process <field>=<value> the same as
  <field>::<value>, if <field> is an indexed field.
* When set to "true": Allows a larger set of metasearches when the
  'allow_inexact_metasearch' setting is "false". However, some of the
  metasearches might be inconsistent with the results of doing a normal
  search.
* Default: false
```

Misc

```
# This section contains miscellaneous search settings.



async_quota_update = <boolean>
* When set to 'true', this setting enables a thread that periodically checks
  the disk quota cache for searches.
  * Because it moves disk quota checking to an async function, this setting
    improves search performance.
  * However, this thread can cause the number of in-process searches to
    slightly exceed concurrent search quotas.
* Set this setting to 'false' if you require strict maintenance of user disk
  quotas.
* Default: false

async_quota_update_freq = <number>
* The frequency, in seconds, at which the disk quota cache for searches is
  updated.
* Applies only when 'async_quota_update=true'.
* Default: 30

use_removable_search_cache = <boolean>
* Determines if the /saved/searches handler will use a cache that
  lets it emit <removable> tags on a list call.
* This slightly changes the appearance of the delete option
  on saved search knowledge objects in Splunk Web, but results
  in a performance boost.

disk_usage_update_period = <number>
* Specifies how frequently, in seconds, should the search process estimate the
  artifact disk usage.
* The quota for the amount of disk space that a search job can use is
  controlled by the 'srchDiskQuota' setting in the authorize.conf file.
* Exceeding this quota causes the search to be auto-finalized immediately,
  even if there are results that have not yet been returned.
* Fractional seconds are allowed.
* Default: 10
```

```
dispatch_dir_warning_size = <integer>
* Specifies the number of jobs in the dispatch directory that triggers when
  to issue a bulletin message. The message warns that performance might
  be impacted.
* Default: 5000

do_not_use_summaries = <boolean>
* Do not use this setting without working in tandem with Splunk support.
* This setting is a very narrow subset of 'summary_mode=none'.
* When set to "true": Disables some functionality that is necessary for
  report acceleration.
  * In particular, when set to "true", search processes will no longer query
    the main splunkd's /admin/summarization endpoint for report acceleration
     summary IDs.
* In certain narrow use-cases this might improve performance if report
  acceleration (savedsearches.conf:auto_summarize) is not in use, by lowering
  the main splunkd's process overhead.
* Default: false

enable_createrss_command = <boolean>
* Enables the deprecated 'createrss' search command. Enabling 'createrss'
  does not affect the behavior of the 'rss' alert action.
* This deprecated command is now disabled by default.
* default: false

enable_datamodel_meval = <boolean>
* Enable concatenation of successively occurring evals into a single
  comma-separated eval during the generation of datamodel searches.
* default: true

enable_file_command = <boolean>
* Enables the deprecated 'file' search command.
* This deprecated command is now disabled by default.
* default: false

enable_conditional_expansion = <boolean>
* Determines whether or not scoped conditional expansion of knowledge
* objects occurs during search string expansion. This only applies on
* the search head.
* NOTE: Do not change unless instructed to do so by Splunk Support.
* Default: true

force_saved_search_dispatch_as_user = <boolean>
* Specifies whether to overwrite the "dispatchAs" value.
* When set to "true": The "dispatchAs" value is overwritten by "user"
  regardless of the [user|owner] value in the savedsearches.conf file.
* When set to "false": The value in the savedsearches.conf file is used.
* You might want to set this to "true" to effectively disable
  "dispatchAs = owner" for the entire install, if that more closely aligns
  with security goals.
* Default: false

get_summary_id_connection_timeout = <integer>
* The connection timeout, in seconds, for a search to check for
  available summaries using the admin/summarization REST endpoint.
* This setting does not apply if 'do_not_use_summaries' is "true", or
  if 'summary_mode' is set to "none".
* Default: 5

get_summary_id_rcv_timeout = <integer>
* The timeout, in seconds, for a search to receive data from the
```

```
  admin/summarization REST endpoint when checking for available summaries.
* This setting does not apply if 'do_not_use_summaries' is "true", or
  if 'summary_mode' is set to "none".
* Default: 5

get_summary_id_send_timeout = <integer>
* The timeout, in seconds, for a search to send a query to the
  admin/summarization REST endpoint when checking for available
  summaries.
* This setting does not apply if 'do_not_use_summaries' is "true", or
  if 'summary_mode' is set to "none".
* Default: 5

max_id_length = <integer>
* Maximum length of the custom search job ID when spawned by using
  REST API argument "id".
* Default: 150

max_id_length_before_hash = <integer>
* Specifies the maximum length of a generated or custom search job ID before
  the Splunk software shortens the directory name. The search job ID itself
  remains the same.
* If set to 0, the Splunk software never hashes the ID. In this case, IDs that
  are too long cause the search to fail.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 230

search_keepalive_frequency = <integer>
* Specifies how often, in milliseconds, a keepalive is sent while a search
  is running.
* Default: 30000 (30 seconds)

search_keepalive_max = <integer>
* The maximum number of uninterupted keepalives before the connection is closed.
* This counter is reset if the search returns results.
* Default: 100

search_retry = <Boolean>
* Specifies whether the Splunk software reruns all or elements of a currently
  running search process when the search process is affected by indexer
  failures in an indexer clustering environment.
  * Indexers can fail during rolling restart or indexer upgrade when indexer
    clustering is enabled. Indexer reboots can also result in failures.
* When set to 'true', the Splunk software attempts to rerun search processes
  that are affected by indexer failures. The Splunk software can rerun entire
  searches and it can rerun searches from the indexer fail point. Do not set
  the value to "1" to indicate "true", because some systems might not parse
  this value correctly.
  * NOTE: Splunk software performs search reruns on a best effort basis. When
    you enable this setting it is possible for Splunk software to return
    partial results for searches without warning.
* When set to 'false', search processes stop returning results from specific
  indexers when those indexers fail, and the Splunk software does not rerun
  those searches.
* Default: false

search_retry_max_historical = <integer>
* Specifies the maximum number of attempts that the Splunk software makes to
  rerun a historical search as described by 'search_retry'.
* This setting is applied only when 'search_retry = true'.
* This setting applies only to historical searches.
* When the number of attempts exceeds 'search_retry_max_historical', the search
```

```
  fails with an error stating that results are incomplete.
* Default: 1


search_retry_waiting_time = <integer>
* Sets how long, in seconds, 'search_retry' waits to get updated indexer
  information.
* The wait time required for recovery after indexer failure can vary depending
  on your indexer environment.
* Increase this value if your environment needs more recovery time to get
  updated indexer information.
* The value should be >= 1
* Default: 70

stack_size = <integer>
* The stack size, in bytes, of the thread that executes the search.
* Default: 4194304 (4MB)

summary_mode = [all|only|none]
* Specifies if precomputed summary data are to be used.
* When set to "all": Use summary data if possible, otherwise use raw data.
* When set to "only": Use summary data if possible, otherwise do not use
  any data.
* When set to "none": Never use precomputed summary data.
* Default: all

track_indextime_range = <boolean>
* Specifies if the system should track the _indextime range of returned
  search results.
* Default: true

use_bloomfilter = <boolean>
* Specifies whether the Splunk software uses Bloom filters to optimize searches.
* When set to 'true', the Splunk software consults 'bloomfilter' files that may
  be present in index buckets to determine whether those buckets contain
  relevant search terms, thereby enabling the software to skip search of tsidx
  files that do not have relevant search terms. In this way, Bloom filter usage
  can improve search performance.
* When set to 'false', the Splunk software searches tsidx summary files without
  filtering out tsidx files that do not have relevant terms.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true

use_metadata_elimination = <boolean>
* Control whether to use metadata to rule out buckets.
* Default: true

results_serial_format = [csv|srs]
* The internal format used for storing serialized results on disk.
* Options:
*    csv: Comma-separated values format
*    srs: Splunk binary format
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: srs

results_compression_algorithm = [gzip|zstd|none]
* The compression algorithm used for storing serialized results on disk.
* Options:
*    gzip: gzip
*    zstd: zstd
*    none: No compression
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
```

```
* Default: zstd

record_search_telemetry = <boolean>
* Controls whether to record search related metrics in search_telemetry.json
  in the dispatch dir. It also indexes this file to the _introspection index.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true


search_telemetry_file_limit = <integer>
* Sets a limit to the number of telemetry files that the Splunk software can
  copy to the var/run/splunk/search_telemetry/ directory, so that it may index
  them in the _introspection index.
* Once this limit is reached, the Splunk software stops adding telemetry files
  to the directory for indexing.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 500


search_telemetry_component_limit = <integer>
* Sets a limit to the size (in bytes) of each of the constituent components in
  the search telemetry json representation.
* Once this limit is reached, the Splunk software will replace the constituent
  component with a simple value: "trimmed".
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 10000

use_dispatchtmp_dir = <boolean>
* DEPRECATED. This setting has been deprecated and has no effect.

auto_cancel_after_pause = <integer>
* Specifies the amount of time, in seconds, that a search must be paused before
  the search is automatically cancelled.
* If set to 0, a paused search is never automatically cancelled.
* Default: 0

always_include_indexedfield_lispy = <boolean>
* Whether or not search always looks for a field that does not have
  "INDEXED = true" set in fields.conf using both the indexed and non-
  indexed forms.
* If set to "true", when searching for <field>=<value>, the lexicon is
  searched for both "<field>::<value>" and "<value>".
* If set to "false", when searching for <field>=<val>, the lexicon is
  searched only for "<value>".
* Set to "true" if you have fields that are sometimes indexed and
  sometimes not indexed.
* For field names that are always indexed, it is much better
  for performance to set "INDEXED = true" in fields.conf for
  that field instead.
* Default: true

indexed_fields_expansion = <boolean>
* Specifies whether search scopes known indexed fields with the source types
  that they are known to be indexed with.
* When set to 'true', for every field known to be indexed, the Splunk software
  converts every known field=val statement to field::val, scoped with the
  applicable sourcetypes.
* Default: true

max_searchinfo_map_size = <integer>
* Maximum number of entries in each SearchResultsInfo data structure map that
  are used to track information about search behavior
```

```
* Default: 50000

track_matching_sourcetypes = <boolean>
* if true, keeps track of the number of events of each sourcetype that match a
  search, and store that information in info.csv
* Default: true

search_launch_timeout_seconds = <positive integer>
* The maximum amount of time, in seconds, to wait before a search job
  is launched successfully.
* If a search job does not launch after the timeout interval elapses,
  the job terminates as a search failure.
* If search jobs time out frequently before successfully launching, check
  whether the server running Splunk software is overloaded. Alternatively,
  change this setting to a number greater than 180.
* For most deployments, 180 seconds is sufficient.
* Default: 180

search_startup_config_timeout_ms = <positive integer>
* The amount of time allowed in milliseconds to initialize a search job's
  configuration, including the knowledge bundle.
* If initializing the search configuration takes longer than the time allowed
  by this setting, the 'DISPATCH_RUNNER:SLOW_CONFIG_INITIAL' warning message
  is displayed in Splunk Web.
* This setting is used only to monitor search performance.
* Default: 3000

max_audit_sourcetypes = <integer>
* if track_matching_sourcetypes = true, the matching sourcetypes
  for a search will be written to the info=completed audit.log message
  upon completion of the search, up to max_audit_sourcetypes.
* If max_audit_sourcetypes is set to 0, sourcetype information
  will not be added to audit.log.
* If the number of matching sourcetypes exceeds the max_audit_sourcetypes
  setting, the sourcetypes with the greatest number of matching
  events will be included.
* Default: 100

use_search_evaluator_v2 = <boolean>
* If true, search evaluator v2 is used.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true

execute_postprocess_in_search = <boolean>
* If true, try to run postprocess searches ahead of time in the search process
  instead of the main splunkd process.
* Default: true

max_fieldmeta_cnt_ui = <number>
* The maximum number of field metadata displayed in the /jobs/fieldmeta endpoint.
* When viewing the search job status for searches with a large
  number of field metadata, decreasing this value will reduce the memory load on
  splunkd mothership, but show less field metadata in the web UI.
* Default: 1000
```

Parsing

```
# This section contains settings related to parsing searches.

max_macro_depth = <integer>
```

```
* Maximum recursion depth for macros. Specifies the maximum levels for macro
  expansion.
* It is considered a search exception if macro expansion does not stop after
  this many levels.
* Value must be greater than or equal to 1.
* Default: 100

max_subsearch_depth = <integer>
* Maximum recursion depth for subsearches. Specifies the maximum levels for
  subsearches.
* It is considered a search exception if a subsearch does not stop after
  this many levels.
* Default: 8

min_prefix_len = <integer>
* The minimum length of a prefix before a wildcard (*) to use in the query
  to the index.
* Default: 1

use_directives = <boolean>
* Specifies whether a search can take directives and interpret them
  into arguments.
* This is used in conjunction with the search optimizer in order to
  improve search performance.
* Default: true
```

Phased execution settings

```
# This section contains settings for multi-phased execution

phased_execution = <boolean>
* DEPRECATED: This setting has been deprecated.

phased_execution_mode = [multithreaded|auto|singlethreaded]
* DEPRECATED: This setting has been deprecated.
* Controls whether searches use the multiple-phase method of search execution,
  which is required for parallel reduce functionality as of Splunk Enterprise
  7.1.0.
* When set to 'multithreaded' the Splunk platform uses the multiple-phase
  search execution method. Allows usage of the 'prjob' command
  and the 'redistribute' command.
* When set to 'auto', the Splunk platform uses the multiple-phase search
  execution method when the 'prjob' command or the 'redistribute' command
  are used in the search string. If neither the 'prjob' command nor the
  'redistribute' command are present in the search string, the single-phase
  search execution method is used.
* When set to 'singlethreaded' the Splunk platform uses the single-threaded
  search execution method, which does not allow usage of the 'prjob' command
  or the 'redistribute' command.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: multithreaded
```

Preview

```
# This section contains settings for previews.

max_preview_period = <integer>
```

```
* The maximum time, in seconds, between previews.
* Used with the preview interval that is calculated with the
  'preview_duty_cycle' setting.
* When set to "0": Specifies unlimited time between previews.
* Default: 0

min_preview_period = <integer>
* The minimum time, in seconds, required between previews. When the calculated
  interval using 'preview_duty_cycle' indicates previews should be run
  frequently. This setting is used to limit the frequency with which previews
  run.
* Default: 1

preview_duty_cycle = <number>
* The maximum time to spend generating previews, as a fraction of the total
  search time.
* Must be > 0.0 and < 1.0
* Default: 0.25

preview_freq = <timespan> or <decimal>
* The minimum amount of time between results preview updates.
* You can specify values for this setting in one of two formats:
  * As a span of time. In this format, you specify an integer and a character
    that represents a time unit, for example, "10s" means 10 seconds. The
    preview updates every period of 'preview_freq'.
  * As a ratio of the amount of time that the search has been running. In this
    format, you specify a decimal. The preview updates as a ratio of the amount
    of time that the search has ben running, or as a ratio of the length of the
    time window for real-time windowed searches.
* If you use the ratio format, you must specify a decimal number above 0 and
  less than 1.
* A setting of 0 disables preview_freq, meaning that there are no limits to the
  minimum time between previews.
* NOTE: Change this setting only when instructed to do so by Splunk Support.
* Default: 0.05
```

Quota or queued searches

```
# This section contains settings for quota or queued searches.

default_allow_queue = <boolean>
* Unless otherwise specified by using a REST API argument, specifies if an
  asynchronous job spawning request should be queued on quota violation.
  If not, an http error of server too busy is returned.
* Default: 1 (true)

dispatch_quota_retry = <integer>
* The maximum number of times to retry to dispatch a search when the quota has
  been reached.
* Default: 4

dispatch_quota_sleep_ms = <integer>
* The time, in milliseconds, between retrying to dispatch a search when a
  quota is reached.
* Retries the given number of times, with each successive wait 2x longer than
  the previous wait time.
* Default: 100

enable_cumulative_quota = <boolean>
```

```
* Specifies whether to enforce cumulative role based quotas.
* Default: false

queued_job_check_freq = <number>
* Frequency, in seconds, to check queued jobs to determine if the jobs can
  be started.
* Fractional seconds are allowed.
* Default: 1.
```

Reading chunk controls

```
# This section contains settings for reading chunk controls.

chunk_multiplier = <integer>
* A multiplier that the 'max_results_perchunk', 'min_results_perchunk', and
  'target_time_perchunk' settings are multiplied by for a long running search.
* Default: 5

long_search_threshold = <integer>
* The time, in seconds, until a search is considered "long running".
* Default: 2

max_rawsize_perchunk = <integer>
* The maximum raw size, in bytes, of results for each call to search
  (in dispatch).
* When set to "0": Specifies that there is no size limit.
* This setting is not affected by the 'chunk_multiplier' setting.
* Default: 100000000 (100MB)

max_results_perchunk = <integer>
* Maximum results for each call to search (in dispatch).
* Must be less than or equal to the 'maxresultrows' setting.
* Default: 2500

min_results_perchunk = <integer>
* The minimum results for each call to search (in dispatch).
* Must be less than or equal to the 'max_results_perchunk' setting.
* Default: 100

target_time_perchunk = <integer>
* The target duration, in milliseconds, of a particular call to fetch
  search results.
* Default: 2000 (2 seconds)
```

Real-time

```
# This section contains settings for real-time searches.

check_splunkd_period = <number>
* Amount of time, in seconds, that determines how frequently the search process
  (when running a real-time search) checks whether the parent process
  (splunkd) is running or not.
* Fractional seconds are allowed.
* Default: 60 (1 minute)

realtime_buffer = <integer>
* Maximum number of accessible events to keep for real-time searches in
```

```
  Splunk Web.
* Acts as circular buffer after this buffer limit is reached.
* Must be greater than or equal to 1.
* Default: 10000
```

Remote storage

```
# This section contains settings for remote storage.

bucket_localize_acquire_lock_timeout_sec = <integer>
* The maximum amount of time, in seconds, to wait when attempting to acquire a
  lock for a localized bucket.
* When set to 0, waits indefinitely.
* This setting is only relevant when using remote storage.
* Default: 60 (1 minute)

bucket_localize_connect_timeout_max_retries = <integer>
* The maximum number of times to retry when getting connect timeouts
  while trying to localize a bucket.
* When set to 0, do not retry
* This setting is only relevant when using remote storage.
* Default: 5

bucket_localize_max_timeout_sec = <integer>
* The maximum amount of time, in seconds, to spend localizing a bucket stored
  in remote storage.
* If the bucket contents (what is required for the search) cannot be localized
  in that timeframe, the bucket will not be searched.
* When set to "0": Specifies an unlimited amount of time.
* This setting is only relevant when using remote storage.
* Default: 300 (5 minutes)

bucket_localize_status_check_period_ms = <integer>
* The amount of time, in milliseconds, between consecutive status checks to see
  if the needed bucket contents required by the search have been localized.
* This setting is only relevant when using remote storage.
* The minimum and maximum values are 10 and 60000, respectively.  If the
  specified value falls outside this range, it is effectively set to the
  nearest value within the range.  For example, if you set the value to
  70000, the effective value will be 60000.
* Default: 50 (.05 seconds)

bucket_localize_status_check_backoff_start_ms = <integer>
* When explicitly set, and different from bucket_localize_status_check_period_ms,
  enables exponential backoff between consecutive status checks for bucket
  localization. Starting from the specified amount of time, in milliseconds, up to
  bucket_localize_status_check_period_ms.
* This setting is only relevant when using remote storage.
* Setting this option is beneficial when bucket contents localize quickly (e.g., in
  less time than the minimal allowed value for bucket_localize_status_check_period_ms),
  or with high variability.
* The minimum and maximum values are 1 and bucket_localize_status_check_period_ms,
  respectively. If the specified value falls outside this range, it is effectively
  set to the nearest value within the range.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 0 (no backoff)

bucket_localize_max_lookahead = <integer>
* Specifies the maximum number of buckets the search command localizes
  for look-ahead purposes, in addition to the required bucket.
```

```
* Increasing this value can improve performance, at the cost of additional
  network/io/disk utilization.
* Valid values are 0-64. Any value larger than 64 will be set to 64. Other
  invalid values will be discarded and the default will be substituted.
* This setting is only relevant when using remote storage.
* Default: 5

bucket_localize_lookahead_priority_ratio = <integer>
* A value of N means that lookahead localizations will occur only 1 out of N
  search localizations, if any.
* Default: 5

bucket_predictor = [consec_not_needed|everything]
* Specifies which bucket file prediction algorithm to use.
* Do not change this unless you know what you are doing.
* Default: consec_not_needed
```

Results storage

```
# This section contains settings for storing final search results.

max_count = <integer>
* The number of events that can be accessible in any given status bucket
  (when status_buckets = 0).
* The last accessible event in a call that takes a base and count.
* NOTE: This value does not reflect the number of events displayed in the
  UI after the search is evaluated or computed.
* Default: 500000

max_events_per_bucket = <integer>
* For searches with 'status_buckets>0', this setting limits the number of
  events retrieved for each timeline bucket.
* Default: 1000 in code.

status_buckets = <integer>
* The approximate maximum number buckets to generate and maintain in the
  timeline.
* Default: 0, which means do not generate timeline information

read_final_results_from_timeliner = <boolean>
* When you run a search of event data where 'status_buckets > 0', this setting
  controls the contents of the results.csv.gz and results.srs.zstd files in the
  search artifact.
* When set to "true", the final results saved to disk by the search process on
  the search head are a sample of events ready from the timeliner. Do not set
  the value to "1" to indicate "true", because some systems might not parse
  this value correctly.
* When set to "false", the final results saved to disk by the search process on
  the search head are all events produced by the last SPL command, up to a
  limit of 'max_count' events.
* The 'read_final_results_from_timeliner' setting affects the output of
  subsequent 'loadjob' searches.
    * When set to "true" the 'loadjob' search returns the sample of the final
      results, not the full result set. For example, if the full result set is
      10k results, it might return only 1000 results.
    * When set to "false" the 'loadjob' search returns the full set of search
      results. For example, if the full result set is 10k results, it returns 10k
      results.
* Default: true
```

```
role_based_field_filtering = <boolean>
* Enable the role-based field filtering feature.
* Default: false

truncate_report = [1|0]
* Specifies whether or not to apply the 'max_count' setting to report output.
* Default: 0 (false)

write_multifile_results_out = <boolean>
* At the end of the search, if results are in multiple files, write out the
  multiple files to the results_dir directory, under the search results
  directory.
* This setting speeds up post-processing search, since the results will
  already be split into appropriate size files.
* Default: true
```

Search process

```
# This section contains settings for search process configurations.

idle_process_cache_search_count = <integer>
* The number of searches that the search process must reach, before purging
  older data from the cache. The purge is performed even if the
  'idle_process_cache_timeout' has not been reached.
* When a search process is allowed to run more than one search, the search
  process can cache some data between searches.
* When set to a negative value: No purge occurs, no matter how many
  searches are run.
* Has no effect on Windows if 'search_process_mode' is not set to "auto"
  or if 'max_searches_per_process' is set to "0" or "1".
* Default: 8

idle_process_cache_timeout = <number>
* The amount of time, in seconds, that a search process must be idle before
  the system purges some older data from these caches.
* When a search process is allowed to run more than one search, the search
  process can cache some data between searches.
* When set to a negative value: No purge occurs, no matter on how long the
  search process is idle.
* When set to "0": Purging always occurs, regardless of whether the process
  has been idle or not.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* Default: 0.5 (seconds)

idle_process_regex_cache_hiwater = <integer>
* A threshold for the number of entries in the regex cache. If the regex cache
  grows to larger than this number of entries, the systems attempts to
  purge some of the older entries.
* When a search process is allowed to run more than one search, the search
  process can cache compiled regex artifacts.
* Usually the 'idle_process_cache_search_count' and the
  'idle_process_cache_timeout' settings keep the regex cache at a
  reasonable size.  This setting prevents the cache from growing
  extremely large during a single large search.
* When set to a negative value: No purge occurs, not matter how large
  the cache.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* Default: 2500
```

```
idle_process_reaper_period = auto | <number>
* The amount of time, in seconds, between checks to determine if there are
  too many idle search processes.
* When a search process is allowed to run more than one search, the system
  checks if there are too many idle search processes.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* Default: 30

launcher_max_idle_checks = auto | <integer>
* Specifies the number of idle processes that are inspected before giving up
  and starting a new search process.
* When allowing more than one search to run for each process, the system
  attempts to find an appropriate idle process to use.
* When set to a negative value: Every eligible idle process is inspected.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process is set to "0" or "1".
* Default: 5

launcher_threads = <integer>
* The number of server thread to run to manage the search processes.
* Valid only when more than one search is allowed to run for each process.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* Default: -1 (a value is selected automatically)

max_idle_process_count = auto | <integer>
* The maximum number of preforked search processes that are
  allowed to be idle and reused by later search execution.
* The setting is valid if the 'enable_search_process_long_lifespan'
  setting in the server.conf file is set to "true". Otherwise, it is
  set to zero when "enable_search_process_long_lifespan = false".
* If the total number of idle search processes exceeds this setting,
  some processes are reaped until the number meets the setting.
* Adjust this setting to control how the Splunk server memory is used by
  idle search processes. The Splunk server can consume more system memory
  when the number of idle search processes is higher.
* When set to "auto": the initial value is set to 64.
* When set to "-1" or another negative value: the setting has no limit.
* This setting is not applied on computers running Windows or when
  'search_process_mode' is not "auto".
* Default: auto

max_idle_process_memory = auto | <integer>
* The maximum amount of memory (RSS) in KB used by a search process that is
  allowed to be idle and reused later after running successfully.
* The setting is valid if the 'enable_search_process_long_lifespan'
  setting in the server.conf file is set to "true".
* If the memory used by a search process when it runs exceeds this setting,
  the process is not reusable. As a result, the process exits after it runs.
* Use this setting to prevent a search process from potential OOM issues
  due to the increase in memory usage after the process has been reused
  too many times.
* When set to "auto": The initial default value is set to "(1024*1024)" KB
  and subsequently adjusted automatically depending on the amount of system
  memory that is available to Splunk software.
* When set to "-1" or another negative value: The memory size is not limited.
* Has no effect on Windows or if "search_process_mode" is not "auto".
* Default: auto

max_search_process_pool = auto | <integer>
```

```
* The maximum number of search processes that can be launched to run searches
  in the pool of preforked search processes.
* The setting is valid if the 'enable_search_process_long_lifespan' setting
  in the server.conf file is set to "true".
* Use this setting to limit the total number of running search processes on
  a search head or peer that is prevented from being overloaded or using
  high system resources (CPU, Memory, etc).
* When set to "auto": Splunk server determines the pool size by multiplying
  the number of CPU cores and the allowed number of search processes (16).
  The pool size is 64 at minimum.
* When set to "-1" or another negative value: The pool size is not limited.
* Has no effect on Windows or if "search_process_mode" is not "auto".
* Default: 2048

max_old_bundle_idle_time = auto | <number>
* The amount of time, in seconds, that a process bundle must be idle before
  the process bundle is considered for reaping.
* Used when reaping idle search processes and the process is not configured
  with the most recent configuration bundle.
* When set to "auto": Splunk software uses the internal default value.
  If the 'enable_search_process_long_lifespan' setting in the server.conf
  file is set to "true", the default value is "300". Otherwise, it is "5".
* When set to "-1" or negative value: The idle processes are not reaped sooner
  than normal if the processes are using an older configuration bundle.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* Default: 5

max_searches_per_process = <integer>
* Specifies the maximum number of searches that each search process can run
  before exiting.
* After a search completes, the search process can wait for another search to
  start and the search process can be reused.
* When set to "0" or "1": The process is never reused.
* When set to a negative value: There is no limit to the number of searches
  that a process can run.
* Has no effect on Windows if 'search_process_mode' is not set to "auto".
* Default: 500 (Linux)
* Default: 1 (Windows)

max_searches_started_per_cycle = <integer>
* Specifies the number of new, concurrent searches started by the search
  launcher in a cycle.
* This limits the contention between running searches and new searches,
  improving search process reuse, and efficiency.
* Note: Do not change this setting unless instructed to do so by Splunk
  Support.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or if
  'max_searches_per_process' is set to "0" or "1".
* Default: 30

max_time_per_process = auto | <number>
* Specifies the maximum time, in seconds, that a process can spend running
  searches.
* When a search process is allowed to run more than one search, limits how
  much time a process can accumulate running searches before the process
  must exit.
* When set to a negative value: There is no limit on the amount of time a
  search process can spend running.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* NOTE: A search can run longer than the value set for 'max_time_per_process'
```

```
  without being terminated. This setting ONLY prevents the process from
  being used to run additional searches after the maximum time is reached.
* Default: auto

process_max_age = <number>
* Specifies the maximum age, in seconds, for a search process.
* When a search process is allowed to run more than one search, a process
  is not reused if the process is older than the value specified.
* When set to a negative value: There is no limit on the age of the
  search process.
* This setting includes the time that the process spends idle, which is
  different than "max_time_per_process" setting.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* NOTE: A search can run longer than the time set for 'process_max_age'
  without being terminated. This setting ONLY prevents that process from
  being used to run more searches after the search completes.
* Default: 7200 (120 minutes or 2 hours)

process_min_age_before_user_change = auto | <number>
* The minimum age, in seconds, of an idle process before using a process
  from a different user.
* When a search process is allowed to run more than one search, the system
  tries to reuse an idle process that last ran a search by the same Splunk
  user.
* If no such idle process exists, the system tries to use an idle process
  from a different user. The idle process from a different user must be
  idle for at least the value specified for the
  'process_min_age_before_user_change' setting.
* When set to "0": Any idle process by any Splunk user can be reused.
* When set to a negative value: Only a search process by same Splunk user
  can be reused.
* Has no effect on Windows if 'search_process_mode' is not set to "auto" or
  if 'max_searches_per_process' is set to "0" or "1".
* Default: 4

search_process_mode = [auto|traditional|debug <debugging-command> <debugging-args>]
* Controls how search processes are started.
* When set to "traditional": Each search process is initialized completely
  from scratch.
* When set to "debug": When set to a string beginning with "debug",
  searches are routed through the <debugging-command>, where the user can
  "plug in" debugging tools.
  * The <debugging-command> must reside in one of the following locations:
    * $SPLUNK_HOME/etc/system/bin/
    * $SPLUNK_HOME/etc/apps/$YOUR_APP/bin/
    * $SPLUNK_HOME/bin/scripts/
  * The <debugging-args> are passed, followed by the search command it
    would normally run, to <debugging-command>
    * For example, given the following setting:
        search_process_mode = debug $SPLUNK_HOME/bin/scripts/search-debugger.sh 5
      A command similar to the following is run:
        $SPLUNK_HOME/bin/scripts/search-debugger.sh 5 splunkd search \
        --id=... --maxbuckets=... --ttl=... [...]
* Default: auto

search_process_configure_oom_score_adj = <boolean>
* Determines whether to increase the value of the oom_score (Out of Memory
  Score) for search processes.
* The oom_score is proportional to the amount of memory used by the process,
  and shows how likely the system is to terminate the process due to low
  available memory. When memory runs low, the system kills the process with the
```

```
  highest oom_score to free the most memory.
* If set to true, when system runs out of memory, the kernel preferentially
  kills search processes to protect the main splunkd process and make the
  overall service more stable.
* Applies to Linux operating system only.
* Default: true.

search_process_set_oom_score_adj = <integer>
* Specifies the value added to the existing oom_score for search processes.
* Applies only when 'search_process_configure_oom_score_adj' is set to true.
* The higher the value, the more likely the system is to kill search processes
  before the main splunkd process, decreasing the risk of a Splunk software
  crash.
* Supports integers between 0 and 1000. If set to 0, this setting has no
  effect on searches.
* Generally, the highest oom_score of main splunkd process is less than 700.
  Thus, by adding the default value, in most cases the system is likely to kill
  search processes before it kills the main splunkd process.
* Default: 700.
```

search_messages.log

```
log_search_messages = <boolean>
* Specifies whether splunkd promotes user-facing search messages
  from $SPLUNK_HOME/var/run/splunk/dispatch/<sid>/info.csv to
  $SPLUNK_HOME/var/log/splunk/search_messages.log.
* Splunkd does not promote messages with a severity that is ranked
  lower than the value of search_messages_severity.
* Splunkd promotes messages only after search has been audited.
* The search_messages.log file follows this format when it logs messages:
  orig_component="..." sid="..." peer_name="..." message=...
* Default: true

search_messages_severity = <string>
* When 'log_search_messages = true', this setting specifies the lowest
  severity of message that splunkd logs to search_messages.log.
  The processor ignores all messages with a lower severity.
* Possible values in ascending order: DEBUG, INFO, WARN, ERROR
  * For example, when 'search_messages_severity = WARN', splunkd logs
    only messages with 'WARN' and 'ERROR' severities.
* Default: WARN
```

Search reuse

```
# This section contains settings for search reuse.

allow_reuse = <boolean>
* Specifies whether to allow normally executed historical searches to be
  implicitly re-used for newer requests if the newer request allows it.
* Default: true

reuse_map_maxsize = <integer>
* Maximum number of jobs to store in the reuse map.
* Default: 1000
```

Splunk Analytics for Hadoop

```
# This section contains settings for use with Splunk Analytics for Hadoop.

reduce_duty_cycle = <number>
* The maximum time to spend performing the reduce, as a fraction of total
  search time.
* Must be > 0.0 and < 1.0.
* Default: 0.25

reduce_freq = <integer>
* When the specified number of chunks is reached, attempt to reduce
  the intermediate results.
* When set to "0": Specifies that there is never an attempt to reduce the
  intermediate result.
* Default: 10

remote_reduce_limit = <unsigned long>
* The number of results processed by a streaming search before a reduce
  is forced.
* NOTE: this option applies only if the search is run with --runReduce=true
  (currently only Splunk Analytics for Hadoop does this)
* When set to "0": Specifies that there is no limit.
* Default: 1000000

unified_search = <boolean>
* Specifies if unified search is turned on for hunk archiving.
* Default: false
```

Status

```
# This section contains settings for search status.

status_cache_size = <integer>
* The number of status data for search jobs that splunkd can cache in RAM.
  This cache improves performance of the jobs endpoint.
* Default: 10000

status_period_ms = <integer>
* The minimum amount of time, in milliseconds, between successive
  status/info.csv file updates.
* This setting ensures that search does not spend significant time just
  updating these files.
  * This is typically important for very large number of search peers.
  * It could also be important for extremely rapid responses from search
    peers, when the search peers have very little work to do.
* Default: 1000 (1 second)
```

Timelines

```
# This section contains settings for timelines.

remote_event_download_finalize_pool = <integer>
* Size of the pool, in threads, responsible for writing out the full
  remote events.
* Default: 5
```

```
remote_event_download_initialize_pool = <integer>
* Size of the pool, in threads, responsible for initiating the remote
  event fetch.
* Default: 5

remote_event_download_local_pool = <integer>
* Size of the pool, in threads, responsible for reading full local events.
* Default: 5

remote_timeline = <boolean>
* Specifies if the timeline can be computed remotely to enable better
  map/reduce scalability.
* Default: 1 (true)

remote_timeline_connection_timeout = <integer>
* Connection timeout, in seconds, for fetching events processed by remote
  peer timeliner.
* Default: 5.

remote_timeline_fetchall = <boolean>
* When set to "1" (true): Splunk fetches all events accessible through the
  timeline from the remote peers before the job is considered done.
  * Fetching of all events might delay the finalization of some searches,
    typically those running in verbose mode from the main Search view in
    Splunk Web.
  * This potential performance impact can be mitigated by lowering the
    'max_events_per_bucket' settings.
* When set to "0" (false): The search peers might not ship all matching
  events to the search head, particularly if there is a very large number
  of them.
    * Skipping the complete fetching of events back to the search head will
      result in prompt search finalization.
    * Some events may not be available to browse in the UI.
* This setting does NOT affect the accuracy of search results computed by
  reporting searches.
* Default: 1 (true)

remote_timeline_max_count = <integer>
* Maximum number of events to be stored per timeline bucket on each search
  peer.
* Default: 10000

remote_timeline_max_size_mb = <integer>
* Maximum size of disk, in MB, that remote timeline events should take
  on each peer.
* If the limit is reached, a DEBUG message is emitted and should be
  visible in the job inspector or in messages.
* Default: 100

remote_timeline_min_peers = <integer>
* Minimum number of search peers for enabling remote computation of
  timelines.
* Default: 1

remote_timeline_parallel_fetch = <boolean>
* Specifies whether to connect to multiple peers at the same time when
  fetching remote events.
* Default: true

remote_timeline_prefetch = <integer>
* Specifies the maximum number of full eventuate that each peer should
```

```
  proactively send at the beginning.
* Default: 100

remote_timeline_receive_timeout = <integer>
* Receive timeout, in seconds, for fetching events processed by remote peer
  timeliner.
* Default: 10

remote_timeline_send_timeout = <integer>
* Send timeout, in seconds, for fetching events processed by remote peer
  timeliner.
* Default: 10

remote_timeline_thread = <boolean>
* Specifies whether to use a separate thread to read the full events from
  remote peers if 'remote_timeline' is used and 'remote_timeline_fetchall'
  is set to "true".
  Has no effect if 'remote_timeline' or 'remote_timeline_fetchall' is set to
  "false".
* Default: 1 (true)

remote_timeline_touchperiod = <number>
* How often, in seconds, while a search is running to touch remote timeline
  artifacts to keep the artifacts from being deleted by the remote peer.
* When set to "0": The remote timelines are never touched.
* Fractional seconds are allowed.
* Default: 300 (5 minutes)

timeline_events_preview = <boolean>
* When set to "true": Display events in the Search app as the events are
  scanned, including events that are in-memory and not yet committed, instead
  of waiting until all of the events are scanned to see the search results.
  You will not be able to expand the event information in the event viewer
  until events are committed.
* When set to "false": Events are displayed only after the events are
  committed (the events are written to the disk).
* This setting might increase disk usage to temporarily save uncommitted
  events while the search is running. Additionally, search performance might
  be impacted.
* Default: false

timeline_freq = <timespan> or <ratio>
* The minimum amount of time, in seconds, between timeline commits.
* If specified as a number < 1 (and > 0), minimum time between commits is
  computed as a ratio of the amount of time that the search has been running.
* Default: 0
```

TTL

```
# This section contains time to live (ttl) settings.

cache_ttl = <integer>
* The length of time, in seconds, to persist search cache entries.
* Default: 300 (5 minutes)

default_save_ttl = <integer>
* How long, in seconds, the ttl for a search artifact should be extended in
  response to the save control action.
* When set to 0, the system waits indefinitely.
```

```
* Default: 604800 (1 week)

failed_job_ttl = <integer>
* How long, in seconds, the search artifacts should be stored on disk after
  a job has failed. The ttl is computed relative to the modtime of the
  status.csv file of the job, if the file exists, or the modtime of the
  artifact directory for the search job.
* If a job is being actively viewed in the Splunk UI then the modtime of
  the status.csv file is constantly updated such that the reaper does not
  remove the job from underneath.
* Default: 86400 (24 hours)

remote_ttl = <integer>
* How long, in seconds, the search artifacts from searches run in behalf of
  a search head should be stored on the indexer after completion.
* Default: 600 (10 minutes)


ttl = <integer>
* How long, in seconds, the search artifacts should be stored on disk after
  the job completes. The ttl is computed relative to the modtime of the
  status.csv file of the job, if the file exists, or the modtime of the
  artifact directory for the search job.
* If a job is being actively viewed in the Splunk UI then the modtime of
  the status.csv file is constantly updated such that the reaper does not
  remove the job from underneath.
* Default: 600 (10 minutes)

check_search_marker_done_interval = <integer>
* The amount of time, in seconds, that elapses between checks of search marker
  files, such as hot bucket markers and backfill complete markers.
* This setting is used to identify when the remote search process on the
  indexer completes processing all hot bucket and backfill portions of
  the search.
* Default: 60

check_search_marker_sleep_interval = <integer>
* The amount of time, in seconds, that the process will sleep between
  subsequent search marker file checks.
* This setting is used to put the process into sleep mode periodically on the
  indexer, then wake up and check whether hot buckets and backfill portions
  of the search are complete.
* Default: 1

srtemp_dir_ttl = <integer>
* The time to live, in seconds, for the temporary files and directories
  within the intermediate search results directory tree.
* These files and directories are located in $SPLUNK_HOME/var/run/splunk/srtemp.
* Every 'srtemp_dir_ttl' seconds, the reaper removes files and directories
  within this tree to reclaim disk space.
* The reaper measures the time to live through the newest file modification time
  within the directory.
* When set to 0, the reaper does not remove any files or directories in this
  tree.
* Default: 86400 (24 hours)
```

Unsupported settings

```
# This section contains settings that are no longer supported.
```

```
enable_status_cache = <boolean>
* This is not a user tunable setting.  Do not use this setting without
  working in tandem with Splunk personnel.  This setting is not tested at
  non-default.
* This controls whether the status cache is used, which caches information
  about search jobs (and job artifacts) in memory in main splunkd.
* Normally this cacheing is enabled and assists performance. However, when
  using Search Head Pooling, artifacts in the shared storage location will be
  changed by other search heads, so this cacheing is disabled.
* Explicit requests to jobs endpoints , eg /services/search/jobs/<sid> are
  always satisfied from disk, regardless of this setting.
* Default (when search head pooling is not enabled): true
* Default (when search head pooling is enabled): false

status_cache_in_memory_ttl = <positive integer>
* This is not a user tunable setting. Do not use this setting without working
  in tandem with Splunk personnel. This setting is not tested at non-default.
* This setting has no effect unless search head pooling is enabled, AND
  enable_status_cache has been set to true.
* If set, controls the number of milliseconds which a status cache entry may be
  used before it expires.
* Default: 60000 (60 seconds)
```

Unused settings

```
# This section contains settings that have been deprecated. These settings
# remain listed in this file for backwards compatibility.

max_bucket_bytes = <integer>
* This setting has been deprecated and has no effect.

rr_min_sleep_ms = <integer>
* REMOVED.  This setting is no longer used.

rr_max_sleep_ms = <integer>
* REMOVED.  This setting is no longer used.

rr_sleep_factor = <integer>
* REMOVED.  This setting is no longer used.
```

Distributed search throttling

```
# This section describes peer-side settings for distributed search throttling.
```

[search_throttling::per_cpu]

```
max_concurrent = <unsigned integer>
* Sets the maximum number of remote searches for each available CPU.
  The total number of searches for this throttling type is thus calculated as:
  max_searches = max_concurrent x number_of_cpus
* When the calculated value is exceeded, search requests are rejected until the number
  of concurrent searches falls below the limit.
* A value of 0 disables throttling.
* This setting is relevant only when used with 'remote_search_requests_throttling_type'.
```

```
* Default: 12
```

[search_throttling::physical_ram]

```
min_memory_per_search = <unsigned integer>[KB|MB|GB]
* Sets the minimum memory requirement per search instance.
  The total number of searches for this throttling type is thus calculated as:
  max_searches = available_system_memory / min_memory_per_search
* When the calculated value is exceeded, search requests are rejected until the number
  of concurrent searches falls below the limit.
* A value of 0 disables throttling.
* This setting is relevant only when used with 'remote_search_requests_throttling_type'.
* Specify this value as an integer followed by KB, MB, or GB (for example,
  10MB is 10 megabytes)
* Default: 64MB
```

OTHER COMMAND SETTINGS

```
# This section contains the stanzas for the SPL commands, except for the
# search command, which is in separate section.
```

[anomalousvalue]

```
maxresultrows = <integer>
* Configures the maximum number of events that can be present in memory at one
  time.
* Default: The value set for 'maxresultrows' in the [searchresults] stanza,
  which is 50000 by default.

maxvalues = <integer>
* Maximum number of distinct values for a field.
* Default: 0

maxvaluesize = <integer>
* Maximum size, in bytes, of any single value (truncated to this size if
  larger).
* Default: 0
```

[associate]

```
maxfields = <integer>
* Maximum number of fields to analyze.
* Default: 10000

maxvalues = <integer>
* Maximum number of values for any field to keep track of.
* Default: 0

maxvaluesize = <integer>
* Maximum length of a single value to consider.
* Default: 0
```

[autoregress]

```
maxp = <integer>
* Maximum number of events for auto regression.
* Default: 10000

maxrange = <integer>
* Maximum magnitude of range for p values when given a range.
* Default: 1000
```

[collect]

```
format_multivalue_collect = <boolean>
* Specifies whether the 'collect' processor should format multivalued fields
  specially when it collects them into a summary index.
* A setting of 'true' means that the 'collect' processor will break each
  value of a multivalue field out into a discrete key/value pair.
  * For example, when this setting is 'true' and the 'collect' processor is
    given the field 'alphabet' with values 'a, b, c', the 'collect' processor
    adds the following fields to the summary index:
    alphabet="a", alphabet="b", alphabet="c"
* A setting of 'false' means that the 'collect' processor will collect
  each multivalued field as a single key with values listed and
  newline-separated.
  * For example, when this setting is 'false' and the 'collect' processor is
    given the field 'alphabet' with values 'a, b, c', the 'collect' processor
    adds the following field to the summary index:
    alphabet="a
              b
              c"
* Default: false

collect_ignore_minor_breakers = <boolean>
* Specifies whether the 'collect' command adds quotation marks around field
  values containing major or minor breakers when the command collects those
  values into a summary index.
* A setting of 'true' means that the 'collect' command checks for major
  breakers in field values, such as spaces, square or curly brackets,
  parentheses, semicolons, or exclamation points. If 'collect' finds major
  breakers in a field value, it adds quotation marks to that field value. This
  enables the use of 'tstats' with the PREFIX() directive on fields that do not
  contain major breakers.
* A setting of 'false' means that the 'collect' command adds quotation marks
  when it finds either a minor breaker or a major breaker in a field value.
* For example, say you have the field-value pair 'user_name = name@spl.com'. In
  this case both '@' and '.' are minor breakers.
  * When 'collect_ignore_minor_breakers = true', the 'collect' command does not
    enclose the value of 'user_name' in quotation marks when it adds the
    field-value pair to the summary index: user_name = name@spl.com
  * When 'collect_ignore_minor_breakers = false', the 'collect' command encloses
    the value of 'user_name' in quotation marks because 'collect' detects that
    the value contains minor breakers. In this case, this is what 'collect'
    adds to the summary index: user_name = "name@spl.com"
* Default: false
```

[concurrency]

```
max_count = <integer>
* Maximum number of detected concurrencies.
* Default: 10000000
```

[correlate]

```
maxfields = <integer>
* Maximum number of fields to correlate.
* Default: 1000
```

[ctable]

```
* This stanza controls settings for the contingency command.
* Aliases for the contingency command are: ctable and counttable.

maxvalues = <integer>
* Maximum number of columns/rows to generate (the maximum number of distinct
  values for the row field and column field).
* Default: 1000
```

[dbinspect]

```
maxresultrows = <integer>
* The maximum number of result rows that the dbinspect command can fetch
  at one time.
* A smaller value uses less search head memory in scenarios with large
  number of buckets. However, setting the value too small decreases
  search performance.
* Note: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 50000
```

[discretize]

```
* This stanza contains the settings for the bin command.
* Aliases for the bin command are: bucket and discretize.

default_time_bins = <integer>
* When discretizing time for timechart or explicitly via bin, the default bins
  to use if no span or bins is specified.
* Default: 100

maxbins = <integer>
* Maximum number of bins to discretize into.
* If 'maxbins' is not specified or = 0, 'maxbins' uses the value set for
  'maxresultrows' in the [searchresults] stanza, which is 50000 by default.
* Default: 50000
```

[eval]

```
printf_max_precision = <non-negative integer>
* The maximum usable precision for 'printf' format strings.
```

```
* Default: 1000000
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.

printf_max_width = <integer>
* The maximum usable width for 'printf' format strings.
* Default: 1000000
* NOTE: Do not change this setting unless instructed to do so by Splunk
  Support.
```

[findkeywords]

```
maxevents = <integer>
* Maximum number of events used by the findkeywords command and the
  Patterns tab.
* Default: 50000
```

[geomfilter]

```
enable_clipping = <boolean>
* Whether or not polygons are clipped to the viewport provided by the
  render client.
* Default: true

enable_generalization = <boolean>
* Whether or not generalization is applied to polygon boundaries to reduce
  point count for rendering.
* Default: true
```

[geostats]

```
filterstrategy = <integer>
* Controls the selection strategy on the geoviz map.
* Valid values are 1 and 2.

maxzoomlevel = <integer>
* Controls the number of zoom levels that geostats will cluster events on.

zl_0_gridcell_latspan = <decimal>
* Controls what is the grid spacing in terms of latitude degrees at the
  lowest zoom level, which is zoom-level 0.
* Grid-spacing at other zoom levels are auto created from this value by
  reducing by a factor of 2 at each zoom-level.

zl_0_gridcell_longspan = <decimal>
* Controls what is the grid spacing in terms of longitude degrees at the
  lowest zoom level, which is zoom-level 0
* Grid-spacing at other zoom levels are auto created from this value by
  reducing by a factor of 2 at each zoom-level.
```

[inputcsv]

```
mkdir_max_retries = <integer>
* Maximum number of retries for creating a tmp directory (with random name as
  subdir of SPLUNK_HOME/var/run/splunk)
* Default: 100
```

[iplocation]

```
db_path = <path>
* The absolute path to the GeoIP database in the MMDB format.
* The 'db_path' setting does not support standard Splunk environment
  variables such as SPLUNK_HOME.
* Default: The database that is included with the Splunk platform.
```

[join]

```
subsearch_maxout = <integer>
* The maximum number of result rows to output from subsearch to join against
* The join command subsearch results are restricted by two settings, 'subsearch_maxout'
setting in this stanza and 'maxresultrows' setting in the [searchresults] stanza.
* Default: 50000

subsearch_maxtime = <integer>
* Maximum search time, in seconds, before auto-finalization of subsearch.
* Default: 60

subsearch_timeout = <integer>
* Maximum time, in seconds, to wait for subsearch to fully finish.
* Default: 120
* DEPRECATED
```

[kmeans]

```
maxdatapoints = <integer>
* Maximum data points to do kmeans clusterings for.
* Default: 100000000 (100 million)

maxkrange = <integer>
* Maximum number of k values to iterate over when specifying a range.
* Default: 100

maxkvalue = <integer>
* Maximum number of clusters to attempt to solve for.
* Default: 1000
```

[lookup]

```
batch_index_query = <boolean>
* Should non-memory file lookups (files that are too large) use batched queries
  to possibly improve performance?
* Default: true

batch_response_limit = <integer>
* When doing batch requests, the maximum number of matches to retrieve.
* If more than this limit of matches would otherwise be retrieved, the lookup
  falls back to non-batch mode matching.
* Default: 5000000

max_lookup_messages = <positive integer>
* If more than "max_lookup_messages" log entries are generated, additional
  entries will not be logged in info.csv. All entries will still be logged in
  search.log.
```

```
max_matches = <integer>
* DEPRECATED: Use this setting in transforms.conf for lookup definitions.

max_memtable_bytes = <integer>
* Maximum size, in bytes, of static lookup file to use an in-memory index for.
* Lookup files with size above max_memtable_bytes will be indexed on disk
* NOTE: This setting also applies to lookup files loaded through the lookup()
  eval function *which runs at search time*. The same function if called through
  the ingest-eval functionality, uses ingest_max_memtable_bytes instead.
* CAUTION: Setting this to a large value results in loading large lookup
  files in memory. This leads to a bigger process memory footprint.
* Default: 26214400 (25MB)

ingest_max_memtable_bytes = <integer>
* Maximum size, in bytes, of static lookup file to use for a lookup when
  used in the ingest context. (i.e when used with the lookup() eval function
  at ingest time).
* Lookup files with size above ingest_max_memtable_bytes cannot be used for
  the lookup() eval function when used with the ingest-eval functionality.
* CAUTION: Setting this to a large value results in loading large lookup
  files in memory. This leads to a bigger process (splunkd) memory footprint.
* Default: 10485760 (10MB)

ingest_lookup_refresh_period_secs = <integer>
* Period of time, in seconds, after which the in-memory lookup tables that are used
  with the lookup() eval function at ingest time are refreshed.
* This does not apply if the lookup() function is used at search time.
* Default: 60 (1 minute).

indexed_csv_ttl = <positive integer>
* Specifies the amount of time, in seconds, that a indexed CSV lookup table
  can exist without update before it is removed by Splunk software.
* On a period set by 'indexed_csv_keep_alive_timeout', Splunk software checks
  the CSV lookup table to see if it has been updated. If it has been updated,
  Splunk software modifies a special token file.
* At the end of the 'indexed_csv_ttl' period Splunk software looks at the token
  file. If the token file shows that its CSV lookup table has been updated,
  Splunk software does not delete that CSV lookup table.
* Default: 300

indexed_csv_keep_alive_timeout = <positive integer>
* Sets the period, in seconds, for an activity check that Splunk software
  performs on indexed CSV lookup tables.
* When Splunk software performs a CSV lookup table check and finds that the
  table has been updated, it marks this activity on a token file. The token
  file update prevents the CSV lookup table from being deleted after
  'indexed_csv_ttl' seconds of inactivity have passed.
* Default: 30

indexed_csv_inprogress_max_timeout = <positive integer>
* Sets the maximum time, in seconds, for Splunk software to wait for ongoing
  indexing of a CSV lookup table to finish before failing any search that is
  awaiting the lookup table.
* Default: 300

max_reverse_matches = <integer>
* maximum reverse lookup matches (for search expansion)
* Default: 50

shared_provider_cache_size = <integer>
* Sets the cache size in bytes that the Splunk software uses when it shares CSV lookups
```

```
  across multiple lookup commands.
* The <integer> represents the size of the cache in bytes. This is incremented by the
  size of each in-memory file (in bytes) inserted into the shared cache.
* Set this to 0 to disable lookup sharing, defaults to 200MB (209715200 bytes).
* Do not change this value unless you are advised to do so by Splunk Support or
  a similar authority.
* Default: 209715200

input_errors_fatal = <boolean>
* This setting determines whether certain inputlookup or inputcsv command
  errors cause searches to fail or return a warning message.
* When set to true, this setting causes inputlookup and inputcsv errors to make
  an entire search fail. This happens even when the errors take place in a
  subsearch.
* When set to false, this setting returns a warning message for many
  inputlookup and inputcsv error conditions.
* Certain kinds of errors cause searches to fail no matter how this setting is
  set.
* Default: false

enable_splunkd_kv_lookup_indexing = <boolean>
* This setting determines whether KV Store lookup indexing is performed
  during bundle replication.
* When set to true, KVStore lookup indexing occurs on the main splunkd process,
  asynchronous to searches.
* When set to false, KV Store lookup indexing is triggered by the search
  process, potentially slowing search performance.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true

enforce_auto_lookup_order = <boolean>
* true: LOOKUP-<name>s in props.conf are looked up in ASCII order by <name>.
* false: LOOKUP-<name>s in props.conf are looked up in random order.
* Default : false
```

[metadata]

```
bucket_localize_max_lookahead = <integer>
* This setting is only relevant when using remote storage.
* Specifies the maximum number of buckets the metadata command localizes
  for look-ahead purposes, in addition to the required bucket.
* Increasing this value can improve performance, at the cost of additional
  network/io/disk utilization.
* Valid values are 0-64. Any value larger than 64 will be set to 64. Other
  invalid values will be discarded and the default will be substituted.
* Default: 10

maxcount = <integer>
* The total number of metadata search results returned by the search head;
  after the 'maxcount' is reached, any additional metadata results received from
  the search peers will be ignored (not returned).
* A larger number incurs additional memory usage on the search head.
* Default: 100000

maxresultrows = <integer>
* The maximum number of results in a single chunk fetched by the metadata
  command
* A smaller value will require less memory on the search head in setups with
  large number of peers and many metadata results, though, setting this too
  small will decrease the search performance.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
```

```
* Default: 10000
```

[metric_alerts]

```
* This stanza provides global settings for metric alerts.

condition_evaluation_interval = <integer>
* This setting provides the alert condition evaluation interval in minutes.
* Must be a number from 1 to 60.
* Default: 1

search_delay = <time specifier>
* Specifies a delay time for metric alert searches. It can be passed to
  the 'allow_skew' setting for the search.
* The search delay allows the search to wait for the latest indexed data.
* For example,
**  15s+ means search delay is at least 15s after the minute determined by
    `condition_evaluation_interval`.
**  15s+30s means search delay is a random number from 15s to 45s after the minute.
* Only change this setting if you are experiencing significant data latency
  issues.
* Default: 15s+

search_ttl = <positive integer>p
* Specifies the default life span of metric alert search jobs.
* The time to live is defined as "at least until the Nth periodic run of the
  search, where the period is defined by the 'condition_evaluation_interval'
  setting".
* Default: 2p

honor_action = <boolean>
* Specifies whether the Splunk software should change the 'search_ttl' to the
  action ttl when an action is triggered.
* If there are multiple actions, the largest action ttl wins.
* Default: false
```

[msearch]

```
chunk_size = <unsigned integer>
* Specifies the default value of the 'chunk_size' argument for the 'msearch'
  command.
* When you run an 'msearch' search, the search head returns batches of metric
  time series until the search result set is complete.
* This argument sets a limit for the number of metric time series that the
  search head can gather in a single batch from a single MSIDX file. For
  example, when 'chunk_size=100', the search head can return 100 metric time
  series worth of metric data points in batches until the search is complete.
* Lower this value when 'msearch' searches use too much memory, or when they
  infrequently return events.
* Larger 'chunk_size' values can improve search performance, with the tradeoff
  of using more memory per search.
* Smaller 'chunk_size' values can reduce search performance, with the tradeoff
  of using less memory per search.
* This setting cannot be set lower than 10.
* Default: 1000

target_per_timeseries = <unsigned integer>
* Specifies the maximum number of metric data points to retrieve per tsidx file
  associated with an 'msearch' query.
```

```
* When set to 0, this setting returns all data points available within the given
  time range for each time series.
* Default: 5
```

[mvexpand]

```
* This stanza allows for fine tuning of mvexpand search command.

max_mem_usage_mb = <non-negative integer>
* Overrides the default value for 'max_mem_usage_mb'.
* Limits the amount of RAM, in megabytes (MB), a batch of events or results will
  use in the memory of a search process.
* See definition in the [default] stanza for 'max_mem_usage_mb'
  for more details.
* Default: 500
```

[mvcombine]

```
* This stanza allows for fine tuning of mvcombine search command.

max_mem_usage_mb = <non-negative integer>
* Overrides the default value for 'max_mem_usage_mb'
* Limits the amount of RAM, in megabytes (MB), a batch of events or results
  use in the memory of a search process.
* See definition in the [default] stanza for 'max_mem_usage_mb'
  for more details.
* Default: 500
```

[outputlookup]

```
outputlookup_check_permission = <boolean>
* Specifies whether the outputlookup command should verify that users
  have write permissions to CSV lookup table files.
* outputlookup_check_permission is used in conjunction with the
  transforms.conf setting check_permission.
* The system only applies outputlookup_check_permission to .csv lookup
  configurations in transforms.conf that have check_permission=true.
* You can set lookup table file permissions in the .meta file for each lookup
  file, or through the Lookup Table Files page in Settings. By default, only
  users who have the admin or power role can write to a shared CSV lookup
  file.
* Default: false

create_context = [app|user|system]
* Specifies the context where the lookup file will be created for the first time.
  If there is a current application context and the following options,
  file will be created under:
  * app    : etc/apps/<app>/lookups
  * user   : etc/users/<user>/<app>/lookups
  Otherwise, file will be created under:
  * system : etc/system/local/lookups
* Default: app
```

[rare]

```
maxresultrows = <integer>
* Maximum number of result rows to create.
* If not specified, defaults to the value set for 'maxresultrows' in the
  [searchresults] stanza, which is 50000 by default.
* Default: 50000

maxvalues = <integer>
* Maximum number of distinct field vector values to keep track of.
* Default: 0

maxvaluesize = <integer>
* Maximum length of a single value to consider.
* Default: 0
```

[rest]

```
allow_reload = <boolean>
* Whether or not the '_reload' action is allowed for the
  'rest' search command.
* If you must use '_reload' with the 'rest' search command,
  set 'allow_reload' to "true".
* Use of '_reload' with the 'rest' search command is deprecated.
* Default: true
```

[set]

```
maxresultrows = <integer>
* The maximum number of results the set command will use from each result
  set to compute the required set operation.
* Default: 50000
```

[sort]

```
maxfiles = <integer>
* Maximum files to open at once.  Multiple passes are made if the number of
  result chunks exceeds this threshold.
* Default: 64.
```

[spath]

```
extract_all = <boolean>
* Controls whether to respect automatic field extraction when spath is
  invoked manually.
* If set to "true", all fields are extracted regardless of settings.
* If set to "false", only fields used by later search commands are extracted.
* Default: true

extraction_cutoff = <integer>
* For 'extract-all' spath extraction mode, this setting applies extraction only
  to the first <integer> number of bytes. This setting applies both the auto kv
  extraction and the spath command, when explicitly extracting fields.
* Default: 5000
```

[stats|sistats]

```
approx_dc_threshold = <unsigned integer>
* Applies specifically to the estdc(x) function (approximate distinct count).
* When the Splunk software uses estdc(x) for commands such as stats, chart, and
  timechart, it does not use approximated results if the actual number of
  distinct values is below this threshold.
* To always use estimation, set 'approx_dc_threshold=1'.
* Note: When 'approx_dc_threshold=0' the Splunk software uses the default value
  for this setting (1000)
* Default: 1000

dc_digest_bits = <integer>
* The size of the digest used for approximating distinct count.
* The digest is configured to be 2 ^ 'dc_digest_bits' bytes in size.
* Must be >= 8 (128B) and <= 16 (64KB)
* Default: 10 (equivalent to 1KB)

default_partitions = <integer>
* Number of partitions to split incoming data into for parallel/multithreaded
  reduce.
* Default: 1

check_for_invalid_time = <boolean>
* Specifies whether the stats processor returns results for searches with
  time-sensitive aggregations such as 'latest', 'latest_time', and 'rate' when
  the '_time' or '_origtime' field is missing from input events.
* When you run a search that fits this description:
  * A setting of 'true' means that the stats processor does not return results
    for that search.
  * A setting of 'false' means that the stats processor returns results for
    that search that are likely incorrect or random.
  * In either case, the stats processor displays an info message that tells you
    what has gone wrong and how it can be corrected.
* Default: false

list_maxsize = <integer>
* Maximum number of list items to emit when using the list() function
  stats/sistats
* Default: 100

max_keymap_rows = <integer>
* Limits the number of result rows that the search head stores in the key map
  during the map phase of a 'stats' operation. The Splunk software looks up
  rows stored in the map and combines them greedily prior to final reduce.
* 'Stats' performance is nonlinear with respect to the number of rows in
  the key map. Limiting the number of rows held can improve performance.
* Excess rows expunged from the key map remain in memory, subject to
  max_mem_usage_mb.
* A key map maps vectors of group-by keys (field values) to their associated
  rows. It is a feature of the 'stats' family of search commands.
* This setting applies particularly to high cardinality searches.
* This setting does not apply to 'streamstats' or 'eventstats' searches.
* Default: 1000000

maxmem_check_freq = <integer>
* How frequently, in number of rows, to check if the in-memory data
  structure size limit is exceeded, as specified by the
  'max_mem_usage_mb' setting.
* Default: 50000
```

```
maxresultrows = <integer>
* Maximum number of rows allowed in the process memory.
* When the search process exceeds 'max_mem_usage_mb' and 'maxresultrows',
  data is sent to the disk.
* If not specified, uses the value set for 'maxresultrows' in the
  [searchresults] stanza, which is 50000 by default.
* Default: 50000

max_stream_window = <integer>
* For the streamstats command, the maximum allow window size.
* Default: 10000

maxvalues = <integer>
* Maximum number of values for any field to keep track of.
* When set to "0": Specifies an unlimited number of values.
* Default: 0

maxvaluesize = <integer>
* Maximum length of a single value to consider.
* When set to "0": Specifies an unlimited number of values.
* Default: 0

max_valuemap_bytes = <integer>
* For the sistats command, the maximum encoded length of the valuemap,
  per result written out.
* If limit is exceeded, extra result rows are written out as needed.
* 0 = no limit per row
* Default: 100000

natural_sort_output = <boolean>
* Whether or not to perform a natural sort on the output of 'stats'
  if the output size is greater than or equal to the 'maxresultrows'
  setting.
* A natural sort means that numbers are sorted numerically and non-numbers
  are sorted lexicographically.
* Default: true

partitions_limit = <integer>
* Maximum number of partitions to split into that can be specified with the
  'partitions' option.
* When exceeded, the number of partitions is reduced to this limit.
* Default: 100

perc_method = nearest-rank|interpolated
* Which method to use for computing percentiles (and medians=50 percentile).
  * nearest-rank picks the number with 0-based rank R =
    floor((percentile/100)*count)
  * interpolated means given F = (percentile/100)*(count-1),
    pick ranks R1 = floor(F) and R2 = ceiling(F).
    Answer = (R2 * (F - R1)) + (R1 * (1 - (F - R1)))
* See wikipedia percentile entries on nearest rank and "alternative methods"
* Default: nearest-rank

perc_digest_type = rdigest|tdigest
* Which digest algorithm to use for computing percentiles
  ( and medians=50 percentile).
  * rdigest picks the rdigest_k, rdigest_maxnodes and perc_method properties.
  * tdigest picks the tdigest_k and tdigest_max_buffer_size properties.
* Default: tdigest

sparkline_maxsize = <integer>
* Maximum number of elements to emit for a sparkline
```

```
* Default: The value of the 'list_maxsize' setting

sparkline_time_steps = <time-step-string>
* Specify a set of time steps in order of decreasing granularity. Use an
  integer and one of the following time units to indicate each step.
  * s = seconds
  * m = minutes
  * h = hours
  * d = days
  * month
* A time step from this list is selected based on the <sparkline_maxsize>
  setting.
* The lowest <sparkline_time_steps> value that does not exceed the maximum number
* of bins is used.
* Example:
  * If you have the following configurations:
  * <sparkline_time_steps> = 1s,5s,10s,30s,1m,5m,10m,30m,1h,1d,1month
  * <sparkline_maxsize> = 100
  * The timespan for 7 days of data is 604,800 seconds.
  * Span = 604,800/<sparkline_maxsize>.
  * If sparkline_maxsize = 100, then
    span = (604,800 / 100) = 60,480 sec == 1.68 hours.
  * The "1d" time step is used because it is the lowest value that does not
    exceed the maximum number of bins.
* Default: 1s,5s,10s,30s,1m,5m,10m,30m,1h,1d,1month


NOTE: The following are rdigest and tdigest settings.
      rdigest is a data structure used to compute approximate order statistics
      (such as median and percentiles) using sublinear space.

rdigest_k = <integer>
* rdigest compression factor
* Lower values mean more compression
* After compression, number of nodes guaranteed to be greater than or equal to
  11 times k.
* Must be greater than or equal to 2.
* Default: 100

rdigest_maxnodes = <integer>
* Maximum rdigest nodes before automatic compression is triggered.
* When set to "1": Specifies to automatically configure based on k value.
* Default: 1

tdigest_k = <integer>
* tdigest compression factor
* Higher values mean less compression, more mem usage, but better accuracy.
* Must be greater than or equal to 1.
* Default: 50

tdigest_max_buffer_size = <integer>
* Maximum number of elements before automatic reallocation of buffer storage
  is triggered.
* Smaller values result in less memory usage but is slower.
* Very small values (<100) are not recommended as they will be very slow.
* Larger values help performance up to a point after which it actually
  hurts performance.
* Recommended range is around 10tdigest_k to 30tdigest_k.
* Default: 1000

tmpfile_compression = <string>
* temporary file compression format, used for stats tmp files only
```

```
* "lz4" indicates use of the lz4 format
* "zstd" indicates use of the zstd format
* "none" indicates use of no compression
* Default: lz4

tmpfile_compression_level = <int>
* Temporary file compression format level.
* If tmpfile_compression is lz4 or zstd, this will indicate the compression level.
* For zstd higher numbers indicate higher speed, and lower compression ratios.
* For lz4 higher numbers indicate lower speed, and higher compression ratios.
* Default: 0

min_chunk_size_kb = <integer>
* Specifies the minimum size of a chunk of intermediate results during 'stats'
  search processing. See 'chunk_size_double_every' for additional details.
* This affects the minimum amount of ram required for low-cardinality 'stats'
  searches as well as the size and number of the files produced when that
  data is spilled to disk due to memory pressure.
* Adjust this value only when such an adjustment is absolutely necessary.
  * If the 'stats' process must use less memory in low cardinality cases,
    reduce this value at the cost of increased filesystem inode usage and
    possibly decreased search performance.
  * If the 'stats' process must use fewer filesystem inodes and create larger
    data chunks even for small searches, increase this value at the cost of
    memory in low cardinality searches.
* Default: 64

max_chunk_size_kb = <integer>
* Specifies the maximum size of a chunk of intermediate results during
  'stats' search processing. See 'chunk_size_double_every' for additional
  details.
* By limiting the maximum chunk size, this setting affects the number of data
  chunks that the 'stats' processor can create when intermediate data is
  spilled to disk due to memory pressure.
* Increase this setting if you need to reduce the filesystem inode usage of your
  'stats' processes.
* This setting should never exceed 1/20th of 'max_mem_usage_mb'.
* Default: 4096

chunk_size_double_every = <integer>
* The 'stats' processor stores intermediate data for 'stats' searches in data
  chunks. These intermediate data chunks must have a size between
  'min_chunk_size_kb' and 'max_chunk_size_kb'.
* At the start of a stats job, the 'stats' processor sets the chunk size
  at the 'min_chunk_size_kb' limit. However, when the number of chunks it
  creates reaches the threshold set by 'chunk_size_double_every', the 'stats'
  processor doubles the size of each chunk it creates thereafter. The 'stats'
  processor continues doubling the chunk size it creates each time it creates
  an additional number of chunks equivalent to 'chunk_size_double_every'. The
  'stats' processor stops doubling the chunk size when it reaches the
  'max_chunk_size_kb' limit.
* This behavior lets the 'stats' processor begin 'stats' processes with small
  data chunks, which reduces ram usage on low cardinality searches. It also
  lets the 'stats' processor increase the chunk size when it spills a lot
  of data to disk, which reduces filesystem inode usage for high cardinality
  searches.
* To minimize allocation of unused memory, increase the
  'chunk_size_double_every' threshold to keep the chunks smaller for a longer
  amount of time.
* To reduce filesystem inode usage, decrease the 'chunk_size_double_every'
  threshold so the 'stats' processor reaches the 'max_chunk_size_kb' limit
  quicker. This lowers the number of temporary files created by the search
```

```
  process.
* Default: 100
```

[top]

```
maxresultrows = <integer>
* Maximum number of result rows to create.
* If not specified, uses the value set for 'maxresultrows' in the
  [searchresults] stanza, which is 50000 by default.
* Default: 50000

maxvalues = <integer>
* Maximum number of distinct field vector values to keep track of.
* Default: 100000

maxvaluesize = <integer>
* Maximum length of a single value to consider.
* Default: 1000
```

[transactions]

```
maxopentxn = <integer>
* Specifies the maximum number of not yet closed transactions to keep in the
  open pool before starting to evict transactions.
* Default: 5000

maxopenevents = <integer>
* Specifies the maximum number of events (which are) part of open transactions
  before transaction eviction starts happening, using LRU policy.
* Default: 100000
```

[tscollect]

```
squashcase = <boolean>
* The default value of the 'squashcase' argument if not specified by the command
* Default: false

keepresults = <boolean>
* The default value of the 'keepresults' argument if not specified by the command
* Default: false

optimize_max_size_mb = <unsigned integer>
* The maximum size in megabytes of files to create with optimize
* Specify 0 for no limit (may create very large tsidx files)
* Default: 1024
```

[tstats]

```
allow_old_summaries = <boolean>
* Whether or not the 'tstats' command, when run on an accelerated datamodel,
  confirms that the datamodel search in each bucket's summary metadata is
  considered to be up to date with the current datamodel search.
* Only bucket summaries that are considered "up to date" are used to
  deliver results.
* This value is the default value of the 'allow_old_summaries' setting,
  if that argument is not specified in the command.
```

```
* When set to "false", 'tstats' always confirms that the datamodel
  search in each bucket's summary metadata is considered up to date with the
  current datamodel search.
* When set to "true", 'tstats' delivers results even from bucket summaries
  that are considered out of date with the current datamodel.
* Default: false

apply_search_filter = <boolean>
* Whether or not 'tstats' applies role-based search filters when users
  run the command on normal index data.
* If set to "true", 'tstats' applies role-based search filters.
* NOTE: Regardless of this setting value, 'tstats' never applies search
  filters to data collected with 'tscollect', or with datamodel acceleration.
* Default: true

bucket_localize_max_lookahead = <integer>
* This setting is only relevant when using remote storage.
* Specifies the maximum number of buckets the tstats command localizes for
  look-ahead purposes, in addition to the required bucket.
* Increasing this value can improve performance, at the cost of additional
  network/io/disk utilization.
* Valid values are 0-64. Any value larger than 64 will be set to 64. Other
  invalid values will be discarded and the default will be substituted.
* Default: 10

chunk_size = <unsigned integer>
* ADVANCED: The default value of 'chunk_size' arg if not specified by
  the command
* This argument controls how many events are retrieved at a time within a
  single TSIDX file when answering queries
* Consider lowering this value if tstats queries are using too much memory
  (cannot be set lower than 10000)
* Larger values will tend to cause more memory to be used (per search) and
  might have performance benefits.
* Smaller values will tend to reduce performance and might reduce memory used
  (per search).
* Altering this value without careful measurement is not advised.
* Default: 10000000

include_events_omitted_when_filtering_numeric_values = <boolean>
* When you run a 'tstats' search that filters numeric values of one or more
  fields, it might omit events where those fields do not exist or have NULL
  values from the search results.
* This setting specifies whether this omission of events takes place when
  'tstats' filters out events based on numeric values of fields.
* A setting of 'true' means that when the 'tstats' command filters out events
  where a field has a specific numeric value, it also matches events where
  that field is not present.
* A setting of 'false' means that when the 'tstats' command filters out events
  where a field has a specific numeric value, it also omits events where that
  field is not present.
* Default: false

summariesonly = <boolean>
* Whether or not 'tstats' employs a mixed mode when running against an
  accelerated datamodel.
* This value is the default value for the 'summariesonly' setting, if that
  argument is not specified in the command.
* In mixed mode, 'tstats' falls back to search if it encounters missing
  tsidx data.
* If set to "true", 'tstats' overrides this mixed mode, and only generates
  results from available tsidx data, which might be incomplete.
```

```
* If set to "false", 'tstats' uses mixed mode, and falls back to search for
  tsidx data that is missing.
* Default: false

warn_on_missing_summaries = <boolean>
* ADVANCED: Only meant for debugging 'summariesonly=true' searches on
  accelerated datamodels.
* When set to "true", search will issue a warning for a tstats 'summariesonly=true'
  search for the following scenarios:
    a) If there is a non-hot bucket that has no corresponding datamodel
    acceleration summary whatsoever.
    b) If the bucket's summary does not match with the current datamodel
    acceleration search.
* Default: false

batch_search_max_pipeline = <integer>
* Controls the number of tstats/mstats search pipelines launched at the
  indexer during batch search.
* Increase the number of search pipelines to improve search performance, at
  the cost of a concurrent increase in thread and memory usage.
* This value applies only to searches that run on remote indexers.
* Default: 1

use_bloomfilter = <boolean>
* Specifies whether the Splunk software uses Bloom filters to optimize searches.
* When set to 'true', the Splunk software consults 'bloomfilter' files that may
  be present in index buckets to determine whether those buckets contain
  relevant search terms, thereby enabling the software to skip search of tsidx
  files that do not have relevant search terms. In this way, Bloom filter usage
  can improve search performance.
* When set to 'false', the Splunk software searches tsidx summary files without
  filtering out tsidx files that do not have relevant terms.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true

update_datamodel_usage_stats = <boolean>
* Specifies whether or not Splunk software can call the summary touch endpoint
  when it detects that it is using summaries from an accelerated data model.
* The summary touch endpoint is an internal endpoint that helps track how
  frequently a summary is being used, if ever.
* When 'update_datamodel_usage_stats' is set to "false", Splunk software skips this
  endpoint call.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true
```

[mstats]

```
time_bin_limit = <unsigned integer>
* Applies only to mstats search jobs.
* Controls how many time bins can be allocated within a single TSIDX file when
  the search head processes mstats search jobs that group results by time (by
  using 'span', for example).
* When this setting is set to 0, there is no time bin limit for qualifying
  mstats search jobs. Removing the time bin limit can cause the Splunk platform
  to run out of memory when you run those jobs.
* Lower this value when your mstats search jobs are using too much memory per
  search.
* Raise this value if your mstats searches return errors when they have wide
  time ranges or their group-by spans are too small.
* The Splunk platform estimates the number of time bins a search requires by
  dividing its time range by its group-by span. If range/span >
```

```
  'time_bin_limit', it outputs an error. This could happen with a search with a
  time range of a year and a span of '1s', for example.
  * The search time range is determined through the 'earliest' and 'latest'
    values for the search.
  * Some types of searches, such as 'all time' searches, do not have 'earliest'
    and 'latest' values. In those cases the Splunk platform checks within each
    single TSIDX file to derive a time range for the search.
* Default: 1000000

use_bloomfilter = <boolean>
* Specifies whether the Splunk software uses Bloom filters to optimize searches.
* When set to 'true', the Splunk software consults 'bloomfilter' files that may
  be present in index buckets to determine whether those buckets contain
  relevant search terms, thereby enabling the software to skip search of tsidx
  files that do not have relevant search terms. In this way, Bloom filter usage
  can improve search performance.
* When set to 'false', the Splunk software searches tsidx summary files without
  filtering out tsidx files that do not have relevant terms.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true
```

[typeahead]

```
cache_ttl_sec = <integer>
* How long, in seconds, the typeahead cached results are valid.
* Default 300

fetch_multiplier = <integer>
* A multiplying factor that determines the number of terms to fetch from the
  index, fetch = fetch_multiplier x count.
* Default: 50

max_concurrent_per_user = <integer>
* The maximum number of concurrent typeahead searches per user. Once this
  maximum is reached only cached typeahead results might be available
* Default: 3

maxcount = <integer>
* Maximum number of typeahead results to find.
* Default: 1000

max_servers = <integer>
* Specifies the maximum number of remote search servers that are used in
  addition to the search head for the purpose of providing typeahead
  functionality.
* When properly set, 'max_servers' minimizes the workload impact of
  running typeahead search jobs in a clustering deployment. If your target
  indexes are evenly distributed among search servers, use the default setting
  or a similarly low number.
* For load balancing, the choice of remote search servers for typeahead
  searches is random.
* When set to "0": There is no limit and all available search servers
  are used for typeahead search jobs.
* Default: 2

min_prefix_length = <integer>
* The minimum length of the string prefix after which to provide typeahead.
* Default: 1

use_cache = <boolean>
* Specifies whether the typeahead cache will be used if use_cache is not
```

```
  specified in the command line or endpoint.
* Default: true or 1

banned_segments = <semicolon-separated-list>
* Specifies a semicolon-separated list of segments. The 'typeahead' search
  processor filters events with these segments out of the results it returns.
* A best practice is to bracket each listed segment with wildcard asterisks
  ('*').
  * For example, if you set 'banned_segments = *password*;*SSN*', the
    'typeahead' processor removes any event that contains the string 'password'
    or 'SSN' from the final result set.
* No default
```

[typer]

```
maxlen = <integer>
* In eventtyping, pay attention to first <integer> characters of any attribute
  (such as _raw), including individual tokens. Can be overridden by supplying
  the typer operator with the argument maxlen (for example,
  "|typer maxlen=300").
* Default: 10000
```

[xyseries]

```
* This stanza allows for fine tuning of xyseries search command.

max_mem_usage_mb = <non-negative integer>
* Overrides the default value for 'max_mem_usage_mb'
* See definition in [default] max_mem_usage_mb for more details
```

GENERAL SETTINGS

```
# This section contains the stanzas for a variety of general settings.
```

[auto_summarizer]

```
allow_event_summarization = <boolean>
* Whether auto summarization of searches whose remote part returns events
  rather than results will be allowed.
* Default: false

cache_timeout = <integer>
* The minimum amount of time, in seconds, to cache auto summary details and
  search hash codes.
* The cached entry expires randomly between 'cache_timeout' and
  2 * "cache_timeout" seconds.
* Default: 600 (10 minutes)

detailed_dashboard = <boolean>
* Turn on/off the display of both normalized and regular summaries in the
  Report Acceleration summary dashboard and details.
* Default: false
```

```
maintenance_period = <integer>
* The period of time, in seconds, that the auto summarization maintenance
  happens
* Default: 1800 (30 minutes)

max_run_stats = <integer>
* Maximum number of summarization run statistics to keep track and expose via
  REST.
* Default: 48

max_verify_buckets = <integer>
* When verifying buckets, stop after verifying this many buckets if no failures
  have been found
* 0 means never
* Default: 100

max_verify_bucket_time = <integer>
* Maximum time, in seconds, to spend verifying each bucket.
* Default: 15

max_verify_ratio = <number>
* Maximum fraction of data in each bucket to verify
* Default: 0.1 (10%)

max_verify_total_time = <integer>
* Maximum total time in seconds to spend doing verification, regardless if any
  buckets have failed or not
* When set to "0": Specifies no limit.
* Default: 0

normalized_summaries = <boolean>
* Turn on/off normalization of report acceleration summaries.
* Default: true

return_actions_with_normalized_ids = [yes|no|fromcontext]
* Report acceleration summaries are stored under a signature/hash which can be
  regular or normalized.
  * Normalization improves the re-use of pre-built summaries but is not
    supported before 5.0. This config will determine the default value of how
    normalization works (regular/normalized)
  * When set to 'fromcontext': Specifies that the end points and summaries
    would be operating based on context.
* Normalization strategy can also be changed via admin/summarization REST calls
  with the 'use_normalization' argument, which can take the values
  "yes"/"no"/"fromcontext"
* Default: fromcontext

search_2_hash_cache_timeout = <integer>
* The amount of time, in seconds, to cache search hash codes
* Default: The value of the 'cache_timeout' setting

shc_accurate_access_counts = <boolean>
* Only relevant if you are using search head clustering
* Turn on/off to make acceleration summary access counts accurate on the
  captain.
* by centralizing

verify_delete = <boolean>
* Should summaries that fail verification be automatically deleted?
* Default: false
```

```
disable_transparent_mode_federation = <boolean>
* Disable fowarding summarization searches to the remote search head for
  federated search transparent mode.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: false
```

[export]

```
add_offset = <boolean>
* Add an offset/row number to JSON streaming output
* Default: true

add_timestamp = <boolean>
* Add a epoch time timestamp to JSON streaming output that reflects the time
  the results were generated/retrieved
* Default: false
```

[extern]

```
perf_warn_limit = <integer>
* Warn when external scripted command is applied to more than this many
  events
* When set to "0": Specifies for no message (message is always INFO level)
* Default: 10000
```

[auth]

```
* Settings for managing auth features.

enable_install_apps = <boolean>
* Whether or not the "install_apps" capability is enabled for app installation,
  uninstallation, creation, and update.
* If set to "true", you must be assigned a role that holds the 'install_apps'
  capability to access the 'apps/local' REST endpoint for app installation,
  uninstallation, creation, and update.
* If set to "false", you must be assigned a role that holds either the
  'admin_all_objects' or 'edit_local_apps' capabilities for app installation,
  uninstallation, creation, and update.
* Default: false
```

[http_input]

```
max_number_of_tokens = <unsigned integer>
* The maximum number of tokens reported by logging input metrics.
* Default: 10000

max_content_length = <integer>
* The maximum length, in bytes, of HTTP request content that is
  accepted by the HTTP Event Collector server.
* Default: 838860800 (~ 800 MB)

max_number_of_ack_channel = <integer>
* The maximum number of ACK channels accepted by HTTP Event Collector
  server.
* Default: 1000000 (~ 1 million)
```

```
max_number_of_acked_requests_pending_query = <integer>
* The maximum number of ACKed requests pending query on HTTP Event
  Collector server.
* Default: 10000000 (~ 10 million)

max_number_of_acked_requests_pending_query_per_ack_channel = <integer>
* The maximum number of ACKed requested pending query per ACK channel on HTTP
  Event Collector server..
* Default: 1000000 (~ 1 million)

metrics_report_interval = <integer>
* The interval, in seconds, of logging input metrics report.
* Default: 60 (1 minute)
```

[indexpreview]

```
max_preview_bytes = <integer>
* Maximum number of bytes to read from each file during preview
* Default: 2000000 (2 MB)

max_results_perchunk = <integer>
* Maximum number of results to emit per call to preview data generator
* Default: 2500

soft_preview_queue_size = <integer>
* Loosely-applied maximum on number of preview data objects held in memory
* Default: 100
```

[inputproc]

```
file_tracking_db_threshold_mb = <integer>
* The size, in megabytes, at which point the file tracking
  database, otherwise known as the "fishbucket" or "btree", rolls over
  to a new file.
* The rollover process is as follows:
  * After the fishbucket reaches 'file_tracking_db_threshold_mb' megabytes
    in size, a new database file is created.
  * From this point forward, the processor writes new entries to the
    new database.
  * Initially, the processor attempts to read entries from the new database,
    but upon failure, falls back to the old database.
  * Successful reads from the old database are written to the new database.
* NOTE: During migration, if this setting doesn't exist, the initialization
  code in splunkd triggers an automatic migration step that reads in the
  current value for "maxDataSize" under the "_thefishbucket" stanza in
  indexes.conf and writes this value into etc/system/local/limits.conf.

learned_sourcetypes_limit = <0 or positive integer>
* Limits the number of entries added to the learned app for performance
  reasons.
* If nonzero, limits two properties of data added to the learned app by the
  file classifier. (Code specific to monitor:: stanzas that auto-determines
  sourcetypes from content.)
  * The number of sourcetypes added to the learned app's props.conf file will
    be limited to approximately this number.
  * The number of file-content fingerprints added to the learned app's
    sourcetypes.conf file will be limited to approximately this number.
* The tracking for uncompressed and compressed files is done separately, so in
  some cases this value may be exceeded.
```

```
* This limit is not the recommended solution for auto-identifying sourcetypes.
  The usual  best practices are to set sourcetypes in input stanzas, or
  alternatively to apply them based on filename pattern in props.conf
  [source::<pattern>] stanzas.
* Default: 1000

max_fd = <integer>
* Maximum number of file descriptors that a ingestion pipeline in Splunk
  will keep open, to capture any trailing data from files that are written
  to very slowly.
* Note that this limit will be applied per ingestion pipeline. For more
  information about multiple ingestion pipelines see parallelIngestionPipelines
  in the server.conf.spec file.
* With N parallel ingestion pipelines the maximum number of file descriptors
  that can be open across all of the ingestion pipelines will be N * max_fd.
* Default: 100

monitornohandle_max_heap_mb = <integer>
* The maximum amount of memory, in megabytes, used by the MonitorNoHandle
  modular input in user mode.
* The memory of this input grows in size when the data being produced
  by applications writing to monitored files comes in faster than the Splunk
  instance can accept it.
* When set to 0, the heap size (memory allocated in the modular input) can grow
  without limit.
* If this size is limited, and the limit is encountered, the input drops
  some data to stay within the limit.
* This setting is valid only on Windows machines.
* Default: 0

tailing_proc_speed = <integer>
* REMOVED.  This setting is no longer used.

monitornohandle_max_driver_mem_mb = <integer>
* The maximum amount of NonPaged memory, in megabytes, used by the kernel
  driver of the MonitorNoHandle modular input.
* The memory of this input grows in size when the data being produced
  by applications writing to monitored files comes in faster than the Splunk
  instance can accept it.
* When set to 0, the NonPaged memory size (memory allocated in the kernel
  driver of the modular input) can grow without limit.
* If this size is limited, and the limit is encountered, the input drops
  some data to stay within the limit.
* This setting is valid only on Windows machines.
* Default: 0

monitornohandle_max_driver_records = <integer>
* The maximum number of in-memory records that the kernel module for
  the MonitorNoHandle modular input stores.
* This setting controls memory growth by limiting the amount of memory
  that the MonitorNoHandle input kernel module uses.
* When 'monitornohandle_max_driver_mem_mb' is set to > 0, this
  setting is ignored.
* The 'monitornohandle_max_driver_mem_mb' and
  'monitornohandle_max_driver_records' settings are mutually exclusive.
* If the limit is encountered, the input drops some data
  to remain within the limit.
* Default: 500.

time_before_close = <integer>
* MOVED.  This setting is now configured per-input in inputs.conf.
* Specifying this setting in limits.conf is DEPRECATED, but overrides
```

```
  the setting for all inputs, for now.
```

[journal_compression]

```
threads = <integer>
* Specifies the maximum number of indexer threads which will be work on
  compressing hot bucket journal data.
* This setting does not typically need to be modified.
* Default: The number of CPU threads of the host machine
```

[kv]

```
avg_extractor_time = <integer>
* Maximum amount of CPU time, in milliseconds, that the average (over search
  results) execution time of a key-value pair extractor will be allowed to take
  before warning. Once the average becomes larger than this amount of time a
  warning will be issued
* Default: 500 (.5 seconds)

limit = <integer>
* The maximum number of fields that an automatic key-value field extraction
  (auto kv) can generate at search time.
* Increase this setting if you want to ensure that the field picker in the Splunk Web
  search page displays all fields.
* Set this value to 0 if you do not want to limit the number of fields
  that can be extracted at search time.
* Default: 100

indexed_kv_limit = <integer>
* The maximum number of fields that can be extracted at index time from a data source.
* This setting does not prevent a search from extracting indexed fields that
  the search needs and explicitly requests.
* The Splunk platform imposes this limit for each index bucket.
* Fields that can be extracted at index time include default fields, custom fields,
  and structured data header fields.
* The summary fields 'host', 'index', 'source', 'sourcetype', 'eventtype', 'linecount',
  'splunk_server', and 'splunk_server_group' do not count against this limit and are
  always returned.
* Increase this setting if, for example, you have indexed data with a large
  number of columns and want to ensure that the field picker in the Splunk Web search
  page displays all fields.
* This setting is different from the 'limit' setting in that it limits field
  extraction in different phases of data processing. Previously, the 'limit'
  setting handled both index-time and search-time field extraction limits, and
  to maintain backward compatibility, both settings work in concert.
* The Splunk platform always uses the higher value for either setting to enforce
  index-time field extraction limits.
  * For example, if you set 'indexed_kv_limit' to "500" and 'limit' to "200",
    then the platform limits indexed-time field extractions to 500 and
    search-time field extractions to 200.
  * If you set 'indexed_kv_limit' to "200" and 'limit' to "500", then the
    platform limits both index-time and search-time field extraction to 500.
* Set this value to 0 if you do not want to limit the number of fields
  that can be extracted at index time.
* Default: 200

maxchars = <integer>
* When non-zero, truncate _raw to this size and then do auto KV.
* Default: 10240 characters
```

```
maxcols = <integer>
* When non-zero, the point at which kv should stop creating new fields.
* Default: 512

max_extractor_time = <integer>
* Maximum amount of CPU time, in milliseconds, that a key-value pair extractor
  will be allowed to take before warning. If the extractor exceeds this
  execution time on any event a warning will be issued
* Default: 1000 (1 second)
```

[kvstore]

```
max_accelerations_per_collection = <unsigned integer>
* The maximum number of accelerations that can be assigned to a single
  collection
* Valid values range from 0 to 50
* Default: 10

max_documents_per_batch_save = <unsigned integer>
* The maximum number of documents that can be saved in a single batch
* Default: 50000

max_fields_per_acceleration = <unsigned integer>
* The maximum number of fields that can be part of a compound acceleration
  (i.e. an acceleration with multiple keys)
* Valid values range from 0 to 50
* Default: 10

max_queries_per_batch = <unsigned integer>
* The maximum number of queries that can be run in a single batch
* Default: 1000

max_rows_in_memory_per_dump = <unsigned integer>
* The maximum number of rows in memory before flushing it to the CSV projection
  of KVStore collection.
* Default: 200

max_rows_per_query = <unsigned integer>
* The maximum number of rows that will be returned for a single query to
  a collection.
* If the query returns more rows than the specified value, then returned
  result set will contain the number of rows specified in this value.
* Default: 50000

max_size_per_batch_result_mb = <unsigned integer>
* The maximum size, in megabytes (MB), of the result set from a set of
  batched queries
* Default: 100

max_size_per_batch_save_mb = <unsigned integer>
* The maximum size, in megabytes (MB), of a batch save query.
* Default: 50

max_size_per_result_mb = <unsigned integer>
* The maximum size, in megabytes (MB), of the result that will be
  returned for a single query to a collection.
* Default: 50

max_threads_per_outputlookup = <unsigned integer>
* The maximum number of threads to use during outputlookup commands on KVStore
```

```
* If the value is 0 the thread count will be determined by CPU count
* Default: 1
```

[kvstore_migration]

```
periodic_timer_interval = <integer>
* The interval, in seconds, at which a search head cluster member polls
  the status of a KV Store migration or upgrade after the start of that migration or upgrade.
* The minimum accepted value is 1.
* The maximum accepted value is 60.
* Default: 10

max_failed_status_unchanged_count = <integer>
* The maximum number of intervals, as determined by
  the 'periodic_timer_interval' setting, that a search head cluster member's
  status can remain in a failed state during a KV Store migration or upgrade before the member retries
  that migration or upgrade. If the number of intervals has been exceeded,
  then the member is marked as aborted.
* Once this limit is reached, the member aborts the migration or upgrade.
* Default: 30
```

[input_channels]

```
max_inactive = <integer>
* The Maximum number of inactive input channel configurations to keep in cache.
* Each source/sourcetype/host combination requires an independent input
  channel, which contains all relevant settings for ingestion.
* When set to 'auto', the Splunk platform will tune this setting based on the
  physical RAM present in the server at startup.
* Increasing this number might help with low ingestion throughput when there
  are no blocked queues (i.e., no 'blocked=true' events for 'group=queue' in
  metrics.log), and splunkd is creating a very high number of new input
  channels (see the value of 'new_channels' in
  'group=map, name=pipelineinputchannel', also in metrics.log), usually in the
  order of thousands. However, this action is only effective when those input
  channels could have been reused: for example, the source, sourcetype, and
  host fields are not generated randomly and tend to be reused within the
  lifetime of cached channel entries.
* Default: auto

lowater_inactive = <integer>
* Size of the inactive input channel cache after which entries will be
  considered for recycling: having its memory reused for storing settings
  for a different input channel.
* When set to 'auto', the Splunk platform will tune this setting value based
  on the value of 'max_inactive'.
* Default: auto

inactive_eligibility_age_seconds = <integer>
* Time, in seconds, after which an inactive input channel will be removed from
  the cache to free up memory.
* Default: 330
```

[ldap]

```
allow_multiple_matching_users = <boolean>
* Whether or not Splunk Enterprise allows login when it finds multiple
```

```
  entries in LDAP with the same value for the 'username' attribute.
* When multiple entries are found, it chooses the first Distinguished Name
  (DN) lexicographically.
* Setting this to false is more secure as it does not allow any ambiguous
  login, but users with duplicate entries will be unable to login.
* Default: true

max_users_to_precache = <unsigned integer>
* The maximum number of users that are pre-cached from LDAP after
  reloading auth.
* Set this to 0 to turn off pre-caching.
```

[metrics]

```
interval = <integer>
* Number of seconds between logging splunkd metrics to metrics.log.
* Minimum of 10.
* Default (Splunk Enterprise): 60
* Default (Splunk Universal Forwarder): 60

maxseries = <integer>
* The number of series to include in the per_x_thruput reports in metrics.log.
* Default: 10
```

[metrics:tcpin_connections]

```
aggregate_metrics = <boolean>
* For each splunktcp connection from forwarder, splunk logs metrics information
  every metrics interval.
* When there are large number of forwarders connected to indexer, the amount of
  information logged can take lot of space in metrics.log. When set to true, it
  will aggregate information across each connection and report only once per
  metrics interval.
* Default: false

suppress_derived_info = <boolean>
* For each forwarder connection, _tcp_Bps, _tcp_KBps, _tcp_avg_thruput,
  _tcp_Kprocessed is logged in metrics.log.
* This can be derived from kb. When set to true, the above derived info will
  not be emitted.
* Default: false

idle_connections_log_frequency =  <integer>
* For each splunktcp connection from forwarder, splunk logs metrics information
  every metrics interval "[metrics]->interval". There may be large number of
  idle connections. Idle connection received zero bytes during last
  "[metrics]->interval"*idle_connections_log_frequency seconds.
* Setting to skip logging idle connection metrics to metrics.log.
* A value of 1 means always log idle connection metrics to metrics.log.
* Default: 1
```

[pdf]

```
max_rows_per_table = <unsigned integer>
* The maximum number of rows that will be rendered for a table within
  integrated PDF rendering.
* Default: 1000
```

```
render_endpoint_timeout = <unsigned integer>
* The number of seconds after which the pdfgen render endpoint will timeout if
  it has not yet finished rendering the PDF output.
* Default: 3600 (60 minutes)
```

[realtime]

```
# Default options for indexer support of real-time searches
# These can all be overridden for a single search via REST API arguments

alerting_period_ms = <integer>
* The time, in milliseconds, to wait between triggering alerts during a
  realtime search.
* This setting limits the frequency at which alerts are triggered during
  realtime search.
* A value of 0 means that alerts are triggered for every batch of events
  that are read. In dense realtime searches with expensive alerts, this
  can overwhelm the alerting system.
* Precedence: Searchhead
* Default: 0

blocking = <boolean>
* Whether or not the indexer should block if a queue is full.
* Default: false

default_backfill = <boolean>
* Whether or not windowed real-time searches should backfill events.
* Default: true

enforce_time_order = <boolean>
* Whether or not real-time searches should ensure that events are sorted in
  ascending time order.
* Splunk Web automatically reverses the order that it displays events for
  real-time searches. If set to "true", the latest events will be shown first.
* Default: true

indexfilter = <boolean>
* Whether or not the indexer should pre-filter events for efficiency.
* Default: 1 (true)

indexed_realtime_update_interval = <integer>
* When you run an indexed realtime search, the list of searchable buckets
  needs to be updated. If the Splunk software is installed on a cluster,
  the list of allowed primary buckets is refreshed. If not installed on
  a cluster, the list of buckets, including any new hot buckets are refreshed.
  This setting controls the interval for the refresh. The setting must be
  less than the "indexed_realtime_disk_sync_delay" setting. If your realtime
  buckets transition from new to warm in less time than the value specified
  for the "indexed_realtime_update_interval" setting, data will be skipped
  by the realtime search in a clustered environment.
* Precedence: Indexers
* Default: 30

indexed_realtime_cluster_update_interval = <integer>
* This setting is deprecated. Use the "indexed_realtime_update_interval"
  setting instead.
* While running an indexed realtime search on a cluster, the list of allowed
  primary buckets is updated. This controls the interval at which the list
  is updated. This value must be less than the
  'indexed_realtime_disk_sync_delay' setting. If your buckets transition from
```

```
  Brand New to warm in less than the interval time specified, indexed
  realtime will lose data in a clustered environment.
* Precedence: Indexers
* Default: 30

indexed_realtime_default_span = <integer>
* An indexed realtime search is made up of many component historical searches
  that by default will span this many seconds. If a component search is not
  completed in this many seconds the next historical search will span the extra
  seconds. To reduce the overhead of running an indexed realtime search you can
  change this span to delay longer before starting the next component
  historical search.
* Precedence: Indexers
* Default: 1

indexed_realtime_disk_sync_delay = <integer>
* The number of seconds to wait for disk flushes to finish when using
  indexed/continuous/pseudo realtime search, so that all data can be seen.
* After indexing there is a non-deterministic period where the files on disk,
  when opened by other programs, might not reflect the latest flush to disk,
  particularly when a system is under heavy load.
* Precedence: SearchHead overrides Indexers
* Default: 60

indexed_realtime_maximum_span = <integer>
* While running an indexed realtime search, if the component searches regularly
  take longer than 'indexed_realtime_default_span' seconds,
  then indexed realtime search can fall more than
  'indexed_realtime_disk_sync_delay' seconds behind realtime.
* Use this setting to set a limit after which search drops data to
  catch back up to the specified delay from realtime, and only
  search the default span of seconds.
* Precedence: API overrides SearchHead overrides Indexers
* Default: 0 (unlimited)

indexed_realtime_use_by_default = <boolean>
* Whether or not the indexedRealtime mode should be used by default.
* Precedence: SearchHead
* This is an app/user level configuration setting, and cannot be set as global.
* Default: false

local_connect_timeout = <integer>
* Connection timeout, in seconds, for an indexer's search process when
  connecting to that indexer's splunkd.
* Default: 5

local_receive_timeout = <integer>
* Receive timeout, in seconds, for an indexer's search process when
  connecting to that indexer's splunkd.
* Default: 5

local_send_timeout = <integer>
* Send timeout, in seconds, for an indexer's search process when connecting
  to that indexer's splunkd.
* Default: 5

max_blocking_secs = <integer>
* Maximum time, in seconds, to block if the queue is full (meaningless
  if blocking = false)
* 0 means no limit
* Default: 60
```

```
queue_size = <integer>
* Size of queue for each real-time search (must be >0).
* Default: 10000
```

[restapi]

```
maxresultrows = <integer>
* Maximum result rows to be returned by /events or /results getters from REST
  API.
* Default: 50000

jobscontentmaxcount = <integer>
* Maximum length of a property in the contents dictionary of an entry from
  /jobs getter from REST API
* Value of 0 disables truncation
* Default: 0

time_format_reject = <regular expression>
* HTTP arguments for time_format and output_time_format that match
  this regex will be rejected.
* The regex will be satisfied by a substring match anywhere in the argument.
* Intended as defense-in-depth against XSS style attacks against browser users
  by crafting specially encoded URLS for them to access splunkd.
* If unset, all argument strings are accepted.
* To disable this check entirely, set the value to empty.
  * Example of disabling: time_format_reject =
* Default: [<>!] , which means that the less-than '<', greater-than '>', and
  exclamation point '!' are not allowed.

restprocessor_errors_fatal = <boolean>
* Determines whether to return a hard error for REST command usages that are
  invalid.
* An invalid REST command usage is a REST request that returns an HTTP status
  outside the range of [200, 300].
* Default: false

max_persistent_connections = <integer>
* The maximum number of persistent processes that EAI custom REST handlers can create to serve REST API
calls in persistent mode.
* A value of "0" means that there is no limit to the number of processes that the handlers can create.
* Default: 3000
```

[reversedns]

```
rdnsMaxDutyCycle = <integer>
* Generate diagnostic WARN in splunkd.log if reverse dns lookups are taking
  more than this percent of time
* Range 0-100
* Default: 10
```

[sample]

```
maxsamples = <integer>
* Default: 10000

maxtotalsamples = <integer>
* Default: 100000
```

[scheduler]

```
action_execution_threads = <integer>
* Number of threads to use to execute alert actions, change this number if your
  alert actions take a long time to execute.
* This number is capped at 100.
* Default: 10

actions_queue_size = <integer>
* The number of alert notifications to queue before the scheduler starts
  blocking, set to 0 for infinite size.
* Default: 500

actions_queue_timeout = <integer>
* The maximum amount of time, in seconds, to block when the action queue size is
  full.
* Default: 30

alerts_expire_period = <integer>
* The amount of time, in seconds, between expired alert removal
* This period controls how frequently the alerts list is scanned, the only
  benefit from reducing this is better resolution in the number of alerts fired
  at the savedsearch level.
* Change not recommended.
* Default: 120

alerts_max_count = <integer>
* Maximum number of unexpired alerts information to keep for the alerts
  manager, when this number is reached Splunk will start discarding the oldest
  alerts.
* Default: 50000

alerts_max_history = <integer>[s|m|h|d]
* Maximum time to search in the past for previously triggered alerts.
* splunkd uses this property to populate the Activity -> Triggered Alerts
  page at startup.
* Values greater than the default may cause slowdown.
* Relevant units are: s, sec, second, secs, seconds, m, min, minute, mins,
  minutes, h, hr, hour, hrs, hours, d, day, days.
* Default: 7d

alerts_scoping = host|splunk_server|all
* Determines the scoping to use on the search to populate the triggered alerts
  page. Choosing splunk_server will result in the search query
  using splunk_server=local, host will result in the search query using
  host=<search-head-host-name>, and all will have no scoping added to the
  search query.
* Default: splunk_server

async_saved_search_fetch = <boolean>
* Enables a separate thread that will fetch scheduled or auto-summarized saved
  searches asynchronously.
* Do not change this setting unless instructed to do so by Splunk support.
* Default: true

async_saved_search_interval = <integer>
* The interval, in seconds, that scheduled or auto-summarized saved searches
  will be fetched asynchronously.
* Has no effect if async_saved_search_fetch is set to false.
* Default: 30
```

```
async_admission_eval_interval = <integer>
* The interval, in seconds, that scheduled saved searches will be evaluated
  for admission rules asynchronously.
* Has no effect if async_saved_search_fetch is set to false.
* If async_saved_search_fetch is false, admission rule evaluation for saved
  searches is done on the scheduler thread.
* Default: 600

auto_summary_perc = <integer>
* The maximum number of concurrent searches to be allocated for auto
  summarization, as a percentage of the concurrent searches that the scheduler
  can run.
* Auto summary searches include:
  * Searches which generate the data for the Report Acceleration feature.
  * Searches which generate the data for Data Model acceleration.
* NOTE: user scheduled searches take precedence over auto summary searches.
* Default: 50

auto_summary_perc.<n> = <integer>
auto_summary_perc.<n>.when = <cron string>
* The same as auto_summary_perc but the value is applied only when the cron
  string matches the current time.  This allows 'auto_summary_perc' to have
  different values at different times of day, week, month, etc.
* There may be any number of non-negative <n> that progress from least specific
  to most specific with increasing <n>.
* The scheduler looks in reverse-<n> order looking for the first match.
* If either these settings aren't provided at all or no "when" matches the
  current time, the value falls back to the non-<n> value of 'auto_summary_perc'.

concurrency_message_throttle_time = <integer>[s|m|h|d]
* Amount of time controlling throttling between messages warning about scheduler
  concurrency limits.
* Relevant units are: s, sec, second, secs, seconds, m, min, minute, mins,
  minutes, h, hr, hour, hrs, hours, d, day, days.
* Default: 10m

introspection_lookback = <duration-specifier>
* The amount of time to "look back" when reporting introspection statistics.
* For example: what is the number of dispatched searches in the last 60 minutes?
* Use [<integer>]<unit> to specify a duration;
  a missing <integer> defaults to 1.
* Relevant units are: m, min, minute, mins, minutes, h, hr, hour, hrs, hours,
  d, day, days, w, week, weeks.
* For example: "5m" = 5 minutes, "1h" = 1 hour.
* Default: 1h

max_action_results = <integer>
* The maximum number of results to load when triggering an alert action.
* Default: 50000

max_continuous_scheduled_search_lookback = <duration-specifier>
* The maximum amount of time to run missed continuous scheduled searches for
  once Splunk Enterprise comes back up, in the event it was down.
* Use [<integer>]<unit> to specify a duration;
  a missing <integer> defaults to 1.
* Relevant units are: m, min, minute, mins, minutes, h, hr, hour, hrs, hours,
  d, day, days, w, week, weeks, mon, month, months.
* For example: "5m" = 5 minutes, "1h" = 1 hour.
* A value of 0 means no lookback.
* Default: 24h

max_lock_files = <integer>
```

```
* The number of most recent lock files to keep around.
* This setting only applies in search head pooling.

max_lock_file_ttl = <integer>
* Time, in seconds, that must pass before reaping a stale lock file.
* Only applies in search head pooling.

max_per_result_alerts = <integer>
* Maximum number of alerts to trigger for each saved search instance (or
  real-time results preview for RT alerts)
* Only applies in non-digest mode alerting. Use 0 to disable this limit
* Default: 500

max_per_result_alerts_time = <integer>
* Maximum amount of time, in seconds, to spend triggering alerts for each
  saved search instance (or real-time results preview for RT alerts)
* Only applies in non-digest mode alerting. Use 0 to disable this limit.
* Default: 300 (5 minutes)

max_searches_perc = <integer>
* The maximum number of searches the scheduler can run, as a percentage of the
  maximum number of concurrent searches, see [search] max_searches_per_cpu for
  how to set the system wide maximum number of searches.
* Default: 50

max_searches_perc.<n> = <integer>
max_searches_perc.<n>.when = <cron string>
* The same as max_searches_perc but the value is applied only when the cron
  string matches the current time.  This allows 'max_searches_perc' to have
  different values at different times of day, week, month, etc.
* There may be any number of non-negative <n> that progress from least specific
  to most specific with increasing <n>.
* The scheduler looks in reverse-<n> order looking for the first match.
* If either these settings aren't provided at all or no "when" matches the
  current time, the value falls back to the non-<n> value of 'max_searches_perc'.

persistence_period = <integer>
* The period, in seconds, between scheduler state persistence to disk. The
  scheduler currently persists the suppression and fired-unexpired alerts to
  disk.
* This is relevant only in search head pooling mode.
* Default: 30

persistance_period = <integer>
* DEPRECATED: Use the 'persistence_period' setting instead.

priority_runtime_factor = <double>
* The amount to scale the priority runtime adjustment by.
* Every search's priority is made higher (worse) by its typical running time.
  Since many searches run in fractions of a second and the priority is
  integral, adjusting by a raw runtime wouldn't change the result; therefore,
  it's scaled by this value.
* Default: 10

priority_skipped_factor = <double>
* The amount to scale the skipped adjustment by.
* A potential issue with the priority_runtime_factor is that now longer-running
  searches may get starved.  To balance this out, make a search's priority
  lower (better) the more times it's been skipped.  Eventually, this adjustment
  will outweigh any worse priority due to a long runtime. This value controls
  how quickly this happens.
* Default: 1
```

```
dispatch_retry_delay = <unsigned integer>
* The amount of time, in seconds, to delay retrying a scheduled search that
  failed to dispatch (usually due to hitting concurrency limits).
* Maximum value: 30
* Default: 0

saved_searches_disabled = <boolean>
* Whether saved search jobs are disabled by the scheduler.
* Default: false


scheduled_view_timeout = <integer>[s|m|h|d]
* The maximum amount of time that a scheduled view (pdf delivery) would be
  allowed to render
* Relevant units are: s, sec, second, secs, seconds, m, min, minute, mins,
  minutes, h, hr, hour, hrs, hours, d, day, days.
* Default: 60m

shc_role_quota_enforcement = <boolean>
* When this attribute is enabled, the search head cluster captain enforces
  user-role quotas for scheduled searches globally (cluster-wide).
* A given role can have (n *number_of_members) searches running cluster-wide,
  where n is the quota for that role as defined by srchJobsQuota and
  rtSrchJobsQuota on the captain and number_of_members include the members
  capable of running scheduled searches.
* Scheduled searches will therefore not have an enforcement of user role
  quota on a per-member basis.
* Role-based disk quota checks (srchDiskQuota in authorize.conf) can be
  enforced only on a per-member basis.
  These checks are skipped when shc_role_quota_enforcement is enabled.
* Quota information is conveyed from the members to the captain. Network delays
  can cause the quota calculation on the captain to vary from the actual values
  in the members and may cause search limit warnings. This should clear up as
  the information is synced.
* Default: false

shc_syswide_quota_enforcement = <boolean>
* When this is enabled, Maximum number of concurrent searches is enforced
  globally (cluster-wide) by the captain for scheduled searches.
  Concurrent searches include both scheduled searches and ad hoc searches.
* This is (n * number_of_members) where n is the max concurrent searches per
  node (see max_searches_per_cpu for a description of how this is computed) and
  number_of_members include members capable of running scheduled searches.
* Scheduled searches will therefore not have an enforcement of instance-wide
  concurrent search quota on a per-member basis.
* Note that this does not control the enforcement of the scheduler quota.
  For a search head cluster, that is defined as
  (max_searches_perc * number_of_members)
  and is always enforced globally on the captain.
* Quota information is conveyed from the members to the captain. Network delays
  can cause the quota calculation on the captain to vary from the actual values
  in the members and may cause search limit warnings. This should clear up as
  the information is synced.
* Default: false

shc_local_quota_check = <boolean>
* DEPRECATED. Local (per-member) quota check is enforced by default.
* To disable per-member quota checking, enable one of the cluster-wide quota
  checks (shc_role_quota_enforcement or shc_syswide_quota_enforcement).
* For example, setting 'shc_role_quota_enforcement=true' turns off local role
  quota enforcement for all nodes in the cluster and is enforced cluster-wide
```

```
  by the captain.

shp_dispatch_to_slave = <boolean>
* DEPRECATED; use shp_dispatch_to_member instead.

shp_dispatch_to_member = <boolean>
* By default the scheduler should distribute jobs throughout the pool.
* Default: true

search_history_load_timeout = <duration-specifier>
* The maximum amount of time to defer running continuous scheduled searches
  while waiting for the KV Store to come up in order to load historical data.
  This is used to prevent gaps in continuous scheduled searches when splunkd
  was down.
* Use [<integer>]<unit> to specify a duration; a missing <integer> defaults to 1.
* Relevant units are: s, sec, second, secs, seconds, m, min, minute, mins,
  minutes.
* For example: "60s" = 60 seconds, "5m" = 5 minutes.
* Default: 2m

search_history_max_runtimes = <unsigned integer>
* The number of runtimes kept for each search.
* Used to calculate historical typical runtime during search prioritization.
* Default: 10
```

[search_metrics]

```
debug_metrics = <boolean>
* This indicates whether to output more detailed search metrics for
  debugging.
* This will do things like break out where the time was spent by peer, and might
  add additional deeper levels of metrics.
* This is NOT related to "metrics.log" but to the "Execution Costs" and
  "Performance" fields in the Search inspector, or the count_map in the
  info.csv file.
* Default: false
```

[show_source]

```
distributed = <boolean>
* Whether or not a distributed search is performed to get events from all
  servers and indexes.
* Turning this off results in better performance for show source, but events
  will only come from the initial server and index.
* Default: true

distributed_search_limit = <unsigned integer>
* The maximum number of events that are requested when performing a search
  for distributed show source.
* As this is used for a larger search than the initial non-distributed show
  source, it is larger than max_count
* Splunk software rarely returns anywhere near this number of results,
  as excess results are pruned.
* The point is to ensure the distributed search captures the target event in an
  environment with many events.
* Default: 30000

max_count = <integer>
* Maximum number of events accessible by show_source.
```

```
* The show source command will fail when more than this many events are in the
  same second as the requested event.
* Default: 10000

max_timeafter = <timespan>
* Maximum time after requested event to show.
* Default: '1day' (86400 seconds)

max_timebefore = <timespan>
* Maximum time before requested event to show.
* Default: '1day' (86400 seconds)
```

[rex]

```
match_limit = <integer>
* Limits the amount of resources that are spent by PCRE
  when running patterns that will not match.
* Use this to set an upper bound on how many times PCRE calls an internal
  function, match(). If set too low, PCRE might fail to correctly match
  a pattern.
* Default: 100000

depth_limit = <integer>
* Limits the amount of resources that are spent by PCRE
  when running patterns that will not match.
* Use this to limit the depth of nested backtracking in an internal PCRE
  function, match(). If set too low, PCRE might fail to correctly match
  a pattern.
* Default: 1000
```

[slc]

```
maxclusters = <integer>
* Maximum number of clusters to create.
* Default: 10000.
```

[slow_peer_disconnect]

```
# This stanza contains settings for the heuristic that will detect and
# disconnect slow peers towards the end of a search that has returned a
# large volume of data.

batch_search_activation_fraction = <decimal>
* The fraction of peers that must have completed before disconnection begins.
* This is only applicable to batch search because the slow peers will
  not hold back the fast peers.
* Default: 0.9

bound_on_disconnect_threshold_as_fraction_of_mean = <decimal>
* The maximum value of the threshold data rate that is used to determine
  if a peer is slow.
* The actual threshold is computed dynamically at search time but never exceeds
  (100*maximum_threshold_as_fraction_of_mean)% on either side of the mean.
* Default: 0.2

disabled = <boolean>
* Whether or not this feature is enabled.
```

```
* Default: true

grace_period_before_disconnect = <decimal>
* How long, in seconds, when multiplied by life_time_of_collector, to wait
  while the heuristic claims that a peer is slow, before disconnecting the
  peer.
* If the heuristic consistently claims that the peer is slow for at least
  <grace_period_before_disconnect>*life_time_of_collector seconds, then the
  peer is disconnected.
* Default: 0.1

packets_per_data_point = <unsigned integer>
* Rate statistics will be sampled once every packets_per_data_point packets.
* Default: 500

sensitivity = <decimal>
* Sensitivity of the heuristic to newer values. For larger values of
  sensitivity the heuristic will give more weight to newer statistic.
* Default: 0.3

threshold_connection_life_time = <unsigned integer>
* All peers will be given an initial grace period of at least these many
  seconds before they are considered in the heuristic.
* Default: 5

threshold_data_volume = <unsigned integer>
* The volume of uncompressed data that must have accumulated, in
  kilobytes (KB), from a peer before it is considered in the heuristic.
* Default: 1024
```

[summarize]

```
bucket_refresh_interval = <integer>
* When poll_buckets_until_maxtime is enabled in a non-clustered
  environment, this is the minimum amount of time (in seconds)
  between bucket refreshes.
* Default: 30

bucket_refresh_interval_cluster = <integer>
* When poll_buckets_until_maxtime is enabled in a clustered
  environment, this is the minimum amount of time (in seconds)
  between bucket refreshes.
* Default: 120

hot_bucket_min_new_events = <integer>
* The minimum number of new events that need to be added to the hot bucket
  (since last summarization)  before a new summarization can take place.
  To disable hot bucket summarization set this value to a * large positive
  number.
* Default: 100000

indextime_lag = <unsigned integer>
* The amount of lag time, in seconds, to give indexing to ensure that
  it has synced any received events to disk.
* Effectively, the data that has been received in the past 'indextime_lag'
  seconds is NOT summarized.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 90

max_hot_bucket_summarization_idle_time = <unsigned integer>
* Maximum amount of time, in seconds, a hot bucket can be idle. When the
```

```
  time exceeds the maximum, all of the events are summarized even if there
  are not enough events (determined by the hot_bucket_min_new_events
  attribute).
* Default: 900 (15 minutes)

max_replicated_hot_bucket_idle_time = <unsigned integer>
* The maximum amount of time, in seconds, that a replicated hot bucket
  can remain idle before 'indextime_lag' is no longer applied to it.
* This applies only to idle replicated hot buckets. When new events arrive,
  the default behavior of applying 'indextime_lag' resumes.
* Default: 150

max_summary_ratio = <decimal>
* A number in the [0-1] range that indicates the maximum ratio of
  summary data / bucket size at which point the summarization of that
  bucket, for the particular search, will be disabled.
* Set to 0 to disable.
* Default: 0

max_summary_size = <integer>
* Size of summary, in bytes, at which point we'll start applying the
  max_summary_ratio.
* Set to 0 to disable.
* Default: 0

max_time = <integer>
* The maximum amount of time, seconds, that a summary search process is
  allowed to run.
* Set to 0 to disable.
* Default: 0

poll_buckets_until_maxtime = <boolean>
* Only modify this setting when you are directed to do so by Support.
* Use the datamodels.conf setting 'acceleration.poll_buckets_until_maxtime'
  for individual data models that are sensitive to summarization latency delays.
* Default: false

auto_finalize_secs_after_maxtime = <integer>
* The maximum amount of time, in seconds, that a summary search process is
  allowed to run after having exceeded max_time before it is auto-finalized.
* The Splunk software auto-finalizes searches after a period of time that is
  the combination of this setting with the 'acceleration.max_time' setting.
  * For example, if you set 'acceleration.max_time' to 3600 and you set
    'auto_finalize_secs_after_maxtime' to 300, the Splunk software finalizes
    the search after 3900 seconds.
* An 'acceleration.max_time' setting of "0" indicates that there is no time
  limit for a summary search and causes the Splunk software to ignore the
  'auto_finalize_secs_after_maxtime' setting.
* Default: 300

sleep_seconds = <integer>
* The amount of time, in seconds, to sleep between polling the summarization
  complete status.
* Default: 5

sleep_rebuild_deletion_seconds = <integer>
* The maximum amount of time, in seconds, for Splunk software to wait for data
  model acceleration summary deletion to occur during an automatic summary rebuild.
  When this interval is reached the summary rebuild process moves on to the next
  bucket.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 5
```

```
stale_lock_seconds = <integer>
* The amount of time, in seconds, to have elapse since the mod time of
  a .lock file before summarization considers * that lock file stale
  and removes it.
* Default: 600

tscollect_queue_size = <unsigned integer>
* This setting sets the size (in bytes) of the internal producer-consumer
  queue. Accelerated data model summary creation searches use this queue to
  speed up the summarization task.
* Setting this to a non-zero value reduces the memory usage of the data model
  acceleration search process while accelerating large buckets of events.
* A value of 0 represents no bound on the queue size.
* CAUTION: Do not change this setting without consulting Splunk Support.
  Changing it may slow down the accelerated data model summary creation search.
* Default: 0
```

[system_checks]

```
insufficient_search_capabilities = enabled | disabled
* Enables/disables automatic daily logging of scheduled searches by users
  who have insufficient capabilities to run them as configured.
* Such searches are those that:
  + Have schedule_priority set to a value other than "default" but the
    owner does not have the edit_search_schedule_priority capability.
  + Have schedule_window set to a value other than "auto" but the owner does
    not have the edit_search_schedule_window capability.
* This check and any resulting logging occur on system startup and every 24
  hours thereafter.
* Default: enabled

installed_files_integrity = enabled | log_only | disabled
* Enables/disables automatic verification on every startup that all the
  files that were installed with the running Splunk version are still the
  files that should be present.
  * Effectively this finds cases where files were removed or changed that
    should not be removed or changed, whether by accident or intent.
  * The source of truth for the files that should be present is the manifest
    file in the $SPLUNK_HOME directory that comes with the release, so if
    this file is removed or altered, the check cannot work correctly.
  * Reading of all the files provided with the install has some I/O cost,
    though it is paid out over many seconds and should not be severe.
* When "enabled", detected problems will cause a message to be posted to
  the bulletin board (system UI status message).
* When "enabled" or "log_only", detected problems will cause details to be
  written out to the splunkd.log file.
* When "disabled", no check will be attempted or reported.
* Default: enabled

orphan_searches = enabled|disabled
* Enables/disables automatic UI message notifications to admins for
  scheduled saved searches with invalid owners.
  * Scheduled saved searches with invalid owners are considered "orphaned".
    They cannot be run because Splunk cannot determine the roles to use for
    the search context.
  * Typically, this situation occurs when a user creates scheduled searches
    then departs the organization or company, causing their account to be
    deactivated.
* Currently this check and any resulting notifications occur on system
  startup and every 24 hours thereafter.
```

```
* Default: enabled
```

[thruput]

```
maxKBps = <integer>
* The maximum speed, in kilobytes per second, that incoming data is
  processed through the thruput processor in the ingestion pipeline.
* To control the CPU load while indexing, use this setting to throttle
  the number of events this indexer processes to the rate (in
  kilobytes per second) that you specify.
* NOTE:
  * There is no guarantee that the thruput processor
    will always process less than the number of kilobytes per
    second that you specify with this setting. The status of
    earlier processing queues in the pipeline can cause
    temporary bursts of network activity that exceed what
    is configured in the setting.
  * The setting does not limit the amount of data that is
    written to the network from the tcpoutput processor, such
    as what happens when a universal forwarder sends data to
    an indexer.
  * The thruput processor applies the 'maxKBps' setting for each
    ingestion pipeline. If you configure multiple ingestion
    pipelines, the processor multiplies the 'maxKBps' value
    by the number of ingestion pipelines that you have
    configured.
  * For more information about multiple ingestion pipelines, see
    the 'parallelIngestionPipelines' setting in the
    server.conf.spec file.
* Default (Splunk Enterprise): 0 (unlimited)
* Default (Splunk Universal Forwarder): 256
```

[viewstates]

```
enable_reaper = <boolean>
* Controls whether the viewstate reaper runs.
* Default: true

reaper_freq = <integer>
* Controls how often, in seconds, the viewstate reaper runs.
* Default: 86400 (24 hours)

reaper_soft_warn_level = <integer>
* Controls what the reaper considers an acceptable number of viewstates.
* Default: 1000

ttl = <integer>
* Controls the age, in seconds, at which a viewstate is considered eligible
  for reaping.
* Default: 86400 (24 hours)
```

[scheduled_views]

```
# Scheduled views are hidden [saved searches / reports] that trigger
# PDF generation for a dashboard. When a user enables scheduled PDF delivery
# in the dashboard UI, scheduled views are created.
#
```

```
# The naming pattern for scheduled views is _ScheduledView__<view_name>,
# where <view_name> is the name of the corresponding dashboard.
#
# The scheduled views reaper, if enabled, runs periodically to look for
# scheduled views that have been orphaned. A scheduled view becomes orphaned
# when its corresponding dashboard has been deleted. The scheduled views reaper
# deletes these orphaned scheduled views. The reaper only deletes scheduled
# views if the scheduled views have not been disabled and their permissions
# have not been modified.

enable_reaper = <boolean>
* Controls whether the scheduled views reaper runs, as well as whether
* scheduled views are deleted when the dashboard they reference is deleted.
* Default: true

reaper_freq = <integer>
* Controls how often, in seconds, the scheduled views reaper runs.
* Default: 86400 (24 hours)
```

OPTIMIZATION

```
# This section contains global and specific optimization settings
```

[search_optimization]

```
enabled = <boolean>
* Enables search optimizations
* Default: true
```

[search_optimization::search_expansion]

```
enabled = <boolean>
* Enables optimizer-based search expansion.
* This enables the optimizer to work on pre-expanded searches.
* Default: true

# NOTE: Do not edit the below configurations unless directed by support
```

[search_optimization::replace_append_with_union]

```
enabled = <boolean>
* Enables replace append with union command optimization
* Default: true
```

[search_optimization::merge_union]

```
enabled = <boolean>
* Merge consecutive unions
* Default: true
```

[search_optimization::insert_redistribute_command]

```
enabled = <boolean>
* Enables a search language optimization that inserts a 'redistribute' command.
  This lets you use parallel reduce search processing to shorten the search
  runtime for a set of supported SPL commands.
* This optimization cannot be used by Splunk platform implementations that are
  restricted to the single-threaded search execution method. For more
  information about search execution methods, see the description of the
  'phased_execution_mode' setting in this file.
* Default: true
```

[search_optimization::predicate_merge]

```
enabled = <boolean>
* Enables predicate merge optimization
* Default: true

inputlookup_merge = <boolean>
* Enables predicate merge optimization to merge predicates into inputlookup
* predicate_merge must be enabled for this optimization to be performed
* Default: true

merge_to_base_search = <boolean>
* Enable the predicate merge optimization to merge the predicates into the
  first search in the pipeline.
* Default: true

fields_black_list = <fields_list>
* A comma-separated list of fields that will not be merged into the first
  search in the pipeline.
* If a field contains sub-tokens as values, then the field should be added
  to fields_black_list
* No default.
```

[search_optimization::predicate_push]

```
enabled = <boolean>
* Enables predicate push optimization
* Default: true
```

[search_optimization::predicate_split]

```
enabled = <boolean>
* Enables predicate split optimization
* Default: true
```

[search_optimization::projection_elimination]

```
enabled = <boolean>
* Enables projection elimination optimization
* Default: true

cmds_black_list = <comma separated list>
```

```
* DEPRECATED. Use the 'excluded_commands' setting instead.

excluded_commands = <Commands List>
* A comma-separated list of commands that are not affected by projection
  elimination optimization.
* No default.
```

[search_optimization::required_field_values]

```
enabled = <boolean>
* Enables required field value optimization
* Default: true

fields = <comma-separated-string>
* Provide a comma-separated-list of field names to optimize.
* Currently the only valid field names are eventtype and tag.
* Optimization of event type and tag field values applies to transforming
  searches. This optimization ensures that only the event types and
  tags necessary to process a search are loaded by the search processor.
* Only change this setting if you need to troubleshoot an issue.
* Default: eventtype, tag
```

[search_optimization::search_flip_normalization]

```
enabled = <boolean>
* Enables predicate flip normalization.
* This type of normalization takes 'where' command statements
  in which the value is placed before the field name and reverses
  them so that the field name comes first.
* Predicate flip normalization only works for numeric values and
  string values where the value is surrounded by quotes.
* Predicate flip normalization also prepares searches to take
  advantage of predicate merge optimization.
* Disable search_flip_normalization if you determine that it is
  causing slow search performance.
* Default: true
```

[search_optimization::reverse_calculated_fields]

```
enabled = <boolean>
* Enables reversing of calculated fields optimization.
* Default: true
```

[search_optimization::search_sort_normalization]

```
enabled = <boolean>
* Enables predicate sort normalization.
* This type of normalization applies lexicographical sorting logic
  to 'search' command expressions and 'where' command statements,
  so they are consistently ordered in the same way.
* Disable search_sort_normalization if you determine that it is
  causing slow search performance.
* Default: true
```

[search_optimization::eval_merge]

```
enabled = <boolean>
* Enables a search language optimization that combines two consecutive
  "eval" statements into one and can potentially improve search performance.
* There should be no side-effects to enabling this setting and need not
  be changed unless you are troubleshooting an issue with search results.
* Default: true
```

[search_optimization::replace_table_with_fields]

```
enabled = <boolean>
* Enables a search language optimization that replaces the table
  command with the fields command
  in reporting or stream reporting searches
* There should be no side-effects to enabling this setting and need not
  be changed unless you are troubleshooting an issue with search results.
* Default: true
```

[search_optimization::replace_stats_cmds_with_tstats]

```
enabled = <boolean>
* If you are not using summary indexing, enable this setting to improve
  performance for searches that perform statistical operations only on indexed
  fields.
* Do not enable this optimizer if you are dependent on summary indexes. When it
  is enabled, searches that perform timechart operations on summary indexes may
  need to perform extra work to run a fallback search and may run slower.
  This is because the 'tstats' command does not respect the fields created by
  summary indexing commands. If you use summary indexing but still choose to
  enable this optimization globally, you can disable this optimization on
  a per-search basis by appending
  '| noop search_optimization.replace_stats_cmds_with_tstats=f' to the search
  string.
* Default: true

detect_search_time_field_collisions = <boolean>
* Enables checking field collisions between fields.conf which indicates
  whether a field is indexed and props.conf which may contain fields which
  override those fields at search time.
* This enables logic to perform an additional search expansion before the
  replace_stats_cmds_with_tstats optimizer can be applied so that we
  get correct results when this case occurs.
* Default: true
```

[search_optimization::replace_datamodel_stats_cmds_with_tstats]

```
enabled = <boolean>
* Enables a search language optimization that replaces stats commands with
  tstats commands in "| datamodel .. | stats" and "| from datamodel .. | stats"
  SPL strings.
* Default: true
```

[search_optimization::replace_chart_cmds_with_tstats]

```
* If you are not using summary indexing, enable this optimizer to improve
  performance for searches that perform timechart queries on statistical
  operations only on indexed fields.
* Do not enable this optimizer if you are dependent on summary indexes. When it
  is enabled, searches that perform timechart operations on summary indexes may
  need to perform extra work to run a fallback search and may run slower.
  This is because the 'tstats' command does not respect the fields created by
  summary indexing commands. If you use summary indexing but still choose to
  enable this optimization globally, you can disable this optimization on
  a per-search basis by appending
  '| noop search_optimization.replace_chart_cmds_with_tstats=f'
  to the search string.
* Default: true

detect_search_time_field_collisions = <Boolean>
* When set to 'true', the Splunk software checks for field collisions between
  'fields.conf', which indicates whether a field is indexed, and 'props.conf',
  which may contain fields that override indexed fields at search time.
* This setting enables logic which performs an additional search expansion
  before the replace_chart_cmds_with_tstats optimizer can be applied,
  to ensure that searches return correct results when these field collisions
  occur.
* Default: true
```

[directives]

```
required_tags = enabled|disabled
* Enables the use of the required tags directive, which allows the search
  processor to load only the required tags from the conf system.
* Disable this setting only to troubleshoot issues with search results.
* Default: enabled

required_eventtypes = enabled|disabled
* Enables the use of the required eventtypes directive, which allows the search
  processor to load only the required event types from the conf system.
* Disable this setting only to troubleshoot issues with search results.
* Default: enabled

read_summary = enabled|disabled
* Enables the use of the read summary directive, which allows the search
  processor to leverage existing data model acceleration summary data when it
  performs event searches.
* Disable this setting only to troubleshoot issues with search results.
* Default: enabled
```

[parallelreduce]

```
maxReducersPerPhase = <positive integer>
* The maximum number of valid indexers that can be used as intermediate
  reducers in the reducing phase of a parallel reduce operation. Only healthy
  search peers are valid indexers.
* If you specify a number greater than 200 or an invalid value, parallel
  reduction does not take place. All reduction processing moves to the search
  head.
* Default: 20
```

```
defaultReducersPerPhase = <positive integer>
* Specifies the default number of valid indexers that are used as intermediate
  reducers in the reducing phase of a parallel reduce search job, if the number
  of indexers is not set in the search string by the 'prjob' or 'redistribute'
  commands.
* If 'winningRate' calculates that the size of the potential reducer pool is
  lower than 'defaultReducersPerPhase', the Splunk software uses the number of
  indexers determined by 'winningRate'.
* The value of this setting cannot exceed 'maxReducersPerPhase'.
* Default: 4

maxRunningPrdSearches = <unsigned integer>
* DEPRECATED. Use the 'maxPrdSearchesPerCpu' setting instead.

maxPrdSearchesPerCpu = <unsigned integer>
* The maximum number of parallel reduce searches that can run, per CPU core,
  on an indexer.
* If 'maxPrdSearchesPerCpu=1' and the number of concurrent searches exceeds the
  number of CPU cores on the indexer, new search requests will fail.
* If 'maxPrdSearchesPerCpu=0', there is no limit. The indexer runs as many
  parallel reduce searches as the indexer hardware permits
* Default: 0

reducers = <string>
* Use this setting to configure one or more valid indexers as dedicated
  intermediate reducers for parallel reduce search operations. Only healthy
  search peers are valid indexers.
* For <string>, specify the indexer host and port using the following format -
  host:port. Separate each host:port pair with a comma to specify a list of
  intermediate reducers.
* If the 'reducers' list includes one or more valid indexers, all of those
  indexers (and only these indexers) are used as intermediate reducers when you
  run a parallel reduce search. If the number of valid indexers in the
  'reducers' list exceeds 'maxReducersPerPhase', the Splunk software randomly
  selects the set of indexers that are used as intermediate reducers.
* If all of the indexers in the 'reducers' list are invalid, the search runs
  without parallel reduction. All reduce operations for the search are
  processed on the search head.
* If 'reducers' is empty or not configured, all valid indexers are potential
  intermediate reducer candidates. The Splunk software randomly selects valid
  indexers as intermediate reducers with limits determined by the 'winningRate'
  and 'maxReducersPerPhase' settings.
* Default: ""

winningRate = <positive integer>
* The percentage of valid indexers that can be selected from the search peers
  as intermediate reducers for a parallel reduce search operation.
* This setting is only respected when the 'reducers' setting is empty or not
  configured.
* If 100 is specified, the search head attempts to use all of the indexers.
* If 1 is specified, the search head attempts to use 1% of the indexers.
* The minimum number of indexers used as intermediate reducers is 1.
* The maximum number of indexers used as intermediate reducers is the value of
  'maxReducersPerPhase'.
* Default: 50

rdinPairingTimeout = <positive integer>
* The amount of time (in seconds) to wait so that indexers and intermediate
  indexers may get paired
* Note: Only change this setting unless instructed to do so by Splunk Support.
* Default: 30
```

```
autoAppliedPercentage = <non-negative integer>
* The percentage of search queries to be selected to run as prjob, should be
  in range of [0, 100].
* If 100 is specified, all search queries will be wrapped as 'prjob'; if 0 is
  specified, no search query will be wrapped.
* Default: 0

autoAppliedToAdhocSearches = <boolean>
* When set to true, the Splunk software uses parallel reduce processing to
  improve the performance of qualifying ad-hoc searches.
* This setting is ignored when '0' is specified for 'autoAppliedPercentage'.
* Default: false

maxPreviewMemUsageMb = <positive integer>
* Sets the maximum amount of memory usage (in MB) that parallel reduce search
  can use in its preview cache.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: 100

enablePreview = <boolean>
* When set to 'true', parallel reduce search jobs generate preview data,
  meaning that partial search results are returned as the search job runs.
* When set to 'false', parallel reduce search jobs do not generate preview data.
  They display only the final results of a parallel reduce search job when the
  search job completes.
* Default: true

disabledCommandList = <comma-separated list>
* Specifies a list of commands that are not run for searches that undergo
  parallel reduce search processing.
  * This list is comma-separated, without spaces.
* For example, to disable the 'dedup' and 'sort' commands in parallel reduce
  searches, set 'disabledCommandList = dedup,sort'.
* Note: Do not change this setting unless instructed to do so by Splunk Support.
* Default: addinfo,lookup

previewReducerDutyCycle = <number>
* Sets the maximum time to spend generating previews on intermediate reducers,
  as a fraction of the total search time.
* Note: This setting affects only preview generation on intermediate reducers.
  This setting is not affected by the 'preview_duty_cycle' setting under the
  '[search]' stanza, which controls preview generation on the search head.
* Must be > 0.0 and < 1.0
* Default: 0.1
```

[rollup]

```
minSpanAllowed = <integer>
* Sets the minimum timespan for the scheduled searches that generate metric
  rollup summaries.
* Each rollup summary uses a scheduled search to provide its metric data point
  aggregations. The interval of the search matches the span defined for the
  rollup summary.
* However, when you run large numbers of scheduled searches with short
  intervals, you can encounter search concurrency problems, where some searches
  skip scheduled runs.
* To reduce the risk of search concurrency issues, this setting ensures that
  the rollup summaries created for your have longer spans.
* Do not set below 60 seconds.
* Default: 300
```

[mcollect]

```
always_use_single_value_output = <boolean>
* When set to true, mcollect outputs metric data points that only have one
  measure per data point.
* When set to false, mcollect outputs metric data points that can have
  several measures per data point.
* When your Splunk platform instance is fully upgraded to Splunk 8.0.0, change
  this setting to 'false'.
* Default:true
```

[segmenter]

```
use_segmenter_v2 = <boolean>
* When set to true, this setting causes certain tokenization operations to use
  SSE (Streaming SIMD Extensions) instructions. This improves overall search
  performance.
* This setting affects only those CPUs that support SSE4.2.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true
```

Required Field Optimization

[search_optimization::set_required_fields]

```
* The settings in this stanza affect how the search processors handle required
  field optimization.
* Required field optimization prevents specified but unused fields from being
  extracted or otherwise created during a search. This can improve search
  performance.

stats = <boolean>
* This setting determines whether the stats processor uses the required field
  optimization methods of Stats V2, or if it falls back to the older, less
  optimized version of required field optimization that was used prior to Stats
  v2.
* When set to 'true': the stats processor uses the Stats v2 version of the
    required field optimization. Do not set the value to "1" to indicate "true",
    because some systems might not parse this value correctly.
* When set to 'false': the stats processor falls back to the older version of
    the required field optimization.
* Do not change this setting unless instructed to do so by Splunk support.
* Default: false
```

[watchdog]

```
stack_files_ttl = <integer>
* The amount of time to keep a watchdog stack file.
* The interval can be specified as a string for minutes, seconds, hours, days.
* For example; 60s, 1m, 1h, 1d etc.
```

```
* These files are located in $SPLUNK_HOME/var/log/watchdog.
* If set to 0, the files will not be removed.
* Default: 7d

stack_files_removal_period = <integer>
* The time interval used to check for files that exceed the 'stack_files_ttl' setting.
* The interval can be specified as a string for minutes, seconds, hours, days.
* For example; 60s, 1m, 1h, 1d etc.
* Default: 1h
```

Ingest Actions

[ingest_actions]

```
rfs.provider.rawdata_limit_mb = <non-negative integer>
* Limits the amount of RAM, in megabytes (MB), that a specific storage provider
  type (such as AWS S3) can use for forwarding events to one or more destinations.
* This limit is applied only to the raw event data held in memory or in the
  process of being written to the storage endpoint.
* If the limit is reached, the RFS worker thread will not fetch further events
  from the pipeline, potentially causing upstream queues to fill up and eventually
  block the overall event pipeline.
* To avoid queue blocking, RFS worker thread may attempt to flush events more
  frequently than required, resulting in creating smaller files than expected.
  Therefore, a lower limit can result in smaller file sizes.
* Default: 1024

rfs.provider.max_workers = <non-negative integer>
* Max number of worker threads per storage provider type (such as AWS S3) used
  to serialize events into compressed JSON file for storing on one or more
  destinations.
* Default: 4

rfsS3DestinationOff = <boolean>
* Specifies whether Ingest Actions S3 destination configuration is turned off.
* If S3 destination configuration is turned off, users will not see "Destination"
  page in the UI.
* If S3 destination configuration is turned off, users will not be able to configure
  S3 destination through REST endpoint.
* S3 destination configuration is turned off by default in GCP instances.
* Default: false
```

SPL2

[spl2]

```
origin = [all|none|<search-origin>]
* Limits where the SPL2 search can originate from.
* Use a comma-separated list for the value. Currently, the only supported value is "ad-hoc".
* Default: all
```

limits.conf.example

```
#   Version 9.2.2
# CAUTION: Do not alter the settings in limits.conf unless you know what you are doing.
# Improperly configured limits may result in splunkd crashes and/or memory overuse.


[searchresults]
maxresultrows = 50000
# maximum number of times to try in the atomic write operation (1 = no retries)
tocsv_maxretry = 5
# retry period is 1/2 second (500 milliseconds)
tocsv_retryperiod_ms = 500

[subsearch]
# maximum number of results to return from a subsearch
maxout = 100
# maximum number of seconds to run a subsearch before finalizing
maxtime = 10
# time to cache a given subsearch's results
ttl = 300

[anomalousvalue]
maxresultrows = 50000
# maximum number of distinct values for a field
maxvalues = 100000
# maximum size in bytes of any single value (truncated to this size if larger)
maxvaluesize = 1000

[associate]
maxfields = 10000
maxvalues = 10000
maxvaluesize = 1000

# for the contingency, ctable, and counttable commands
[ctable]
maxvalues = 1000

[correlate]
maxfields = 1000

# for bin/bucket/discretize
[discretize]
maxbins = 50000
# if maxbins not specified or = 0, defaults to searchresults::maxresultrows

[inputcsv]
# maximum number of retries for creating a tmp directory (with random name in
# SPLUNK_HOME/var/run/splunk)
mkdir_max_retries = 100

[kmeans]
maxdatapoints = 100000000

[kv]
# when non-zero, the point at which kv should stop creating new columns
maxcols = 512

[rare]
maxresultrows = 50000
```

```
# maximum distinct value vectors to keep track of
maxvalues = 100000
maxvaluesize = 1000

[restapi]
# maximum result rows to be returned by /events or /results getters from REST
# API
maxresultrows = 50000

[search]
# how long searches should be stored on disk once completed
ttl = 86400

# the approximate maximum number of timeline buckets to maintain
status_buckets = 300

# the last accessible event in a call that takes a base and bounds
max_count = 10000

# the minimum length of a prefix before a * to ask the index about
min_prefix_len = 1

# the length of time to persist search cache entries (in seconds)
cache_ttl = 300

# By default, we will not retry searches in the event of indexer
# failures with indexer clustering enabled.
# Hence, the default value for search_retry here is false.
search_retry = false

# Timeout value for checking search marker files like hotbucketmarker or backfill
# marker.
check_search_marker_done_interval = 60

# Time interval of sleeping between subsequent search marker files checks.
check_search_marker_sleep_interval = 1

# The total number of concurrent searches is set to 10 manually.
total_search_concurrency_limit = 100

# If number of CPUs in your machine is 14, then the total system-wide limit of
# concurrent historical searches on this machine is 20, which is
# max_searches_per_cpu x number_of_cpus + base_max_searches = 1 x 14 + 6 = 20.
max_searches_per_cpu = 1
base_max_searches = 6

# Whether maximum number of concurrent searches are enforced cluster-wide
# for admission of adhoc searches
shc_adhoc_quota_enforcement = on

# Enable throttling on both CPU and memory
remote_search_requests_throttling_type = per_cpu, physical_ram

# If the peer node has 48 cores, the following setting allows a maximum of 720
# concurrent searches.
[search_throttling::per_cpu]
max_concurrent = 13

# If the peer has 64 GB of RAM, the following setting allows a maximum of 512
# concurrent searches.
[search_throttling::physical_ram]
min_memory_per_search = 134217728
```

```
[scheduler]

# Percent of total concurrent searches that will be used by scheduler is
# total concurrency x max_searches_perc = 20 x 60% = 12 scheduled searches
# User default value (needed only if different from system/default value) when
# no max_searches_perc.<n>.when (if any) below matches.
max_searches_perc = 60

# Increase the value between midnight-5AM.
max_searches_perc.0 = 75
max_searches_perc.0.when = * 0-5 * * *

# More specifically, increase it even more on weekends.
max_searches_perc.1 = 85
max_searches_perc.1.when = * 0-5 * * 0,6

# Maximum number of concurrent searches is enforced cluster-wide by the
# captain for scheduled searches. For a 3 node SHC total concurrent
# searches = 3 x 20 = 60. The total searches (adhoc + scheduled) = 60, then
# no more scheduled searches can start until some slots are free.
shc_syswide_quota_enforcement = true

[slc]
# maximum number of clusters to create
maxclusters = 10000

[findkeywords]
#events to use in findkeywords command (and patterns UI)
maxevents = 50000

[stats]
maxresultrows = 50000
maxvalues = 10000
maxvaluesize = 1000

[top]
maxresultrows = 50000
# maximum distinct value vectors to keep track of
maxvalues = 100000
maxvaluesize = 1000

[search_optimization]
enabled = true

[search_optimization::predicate_split]
enabled = true

[search_optimization::predicate_push]
enabled = true

[search_optimization::predicate_merge]
enabled = true
inputlookup_merge = true
merge_to_base_search = true

[search_optimization::projection_elimination]
enabled = true
excluded_commands = eval, rename

[search_optimization::search_flip_normalization]
enabled = true
```

```
[search_optimization::reverse_calculated_fields]
enabled = true

[search_optimization::search_sort_normalization]
enabled = true

[search_optimization::replace_table_with_fields]
enabled = true

[search_optimization::replace_stats_cmds_with_tstats]
enabled = true
detect_search_time_field_collisions = true

[search_optimization::replace_datamodel_stats_cmds_with_tstats]
enabled = true
```

literals.conf

The following are the spec and example files for `literals.conf`.

literals.conf.spec

```
#   Version 9.2.2
#
# This file and all forms of literals.conf are now deprecated.
# Instead, use the messages.conf file which is documented
# at "Customize Splunk Web messages" in the Splunk documentation.
```

literals.conf.example

```
# Version 9.2.2
#
# This file and all forms of literals.conf are now deprecated.
# Instead, use the messages.conf file which is documented
# at "Customize Splunk Web messages" in the Splunk documentation.
```

macros.conf

The following are the spec and example files for `macros.conf`.

macros.conf.spec

```
# Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use for
# for search language macros.
```

```
#
# There is a macros.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name macros.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see macros.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[<STANZA_NAME>]

```
* Each stanza represents a search macro that can be referenced in any search.
* The stanza name is the name of the macro if the macro takes no arguments.
  Otherwise, the stanza name is the macro name appended with "(<numargs>)",
  where <numargs> is the number of arguments that this macro takes.
* Macros can be overloaded, which means they can have the same name but a
  different number of arguments. If you have these stanzas - [foobar], [foobar(1)],
  [foobar(2)], and so forth - they are not the same macro.
* You can specify settings with a macro, which are described below.
  The settings are:
  * A set of macro arguments (args)
  * A definition string with argument substitutions
  * A validation string, with or without an error message
  * A setting that identifies if the defintion is an eval expression
  * A description for the macro
* Macros can be used in the search language by enclosing the macro name and any
  argument list in backtick marks. For example:`foobar(arg1,arg2)` or `footer`.
* The Splunk platform does not expand macros when they are inside quoted values, for
  example: "foo`bar`baz"

args = <string>,<string>,...
* A comma-separated list of argument names.
* Argument names can only contain alphanumeric characters, underscores ( _ ), and
  hyphens ( - ).
* If the stanza name indicates that this macro takes no arguments, this
  setting is ignored.
* This list cannot contain any repeated elements.

definition = <string>
* The string that the macro will expand to, with the argument substitutions
  made. The exception is when "iseval = true", see below.
* Arguments to be substituted must begin and end with a dollar sign ($). For example:
  "The last part of this string will be replaced by the value of argument foo $foo$".
* The Splunk platform replaces the $<arg>$ pattern globally in the string, even
  inside quotation marks.

validation = <string>
* A validation string that is an 'eval' expression.  This expression must
  evaluate to a Boolean or a string.
* Use this setting to verify that the macro's argument values are acceptable.
* If the validation expression is Boolean, validation succeeds when it returns
  "true". If it returns "false" or is NULL, validation fails and the Splunk platform
  returns the error message defined by the 'errormsg' setting.
* If the validation expression is not Boolean, the Splunk platform expects it to
```

```
  return a string or NULL. If it returns NULL, validation is considered a success.
  Otherwise, the string returned is the error message.

errormsg = <string>
* The error message displayed if the 'validation' setting is a Boolean expression and
  the expression does not evaluate to "true".

iseval = true|false
* If set to "true", the 'definition' setting is expected to be an eval expression that
  returns a string representing the expansion of this macro.
* Default: false.

description = <string>
* OPTIONAL. A simple description of what the macro does.
```

macros.conf.example

```
#   Version 9.2.2
#
# Example macros.conf
#

# macro foobar that takes no arguments can be invoked via `foobar`
[foobar]
# the defintion of a macro can invoke another macro.  nesting can be indefinite
# and cycles will be detected and result in an error
definition = `foobar(foo=defaultfoo)`


# macro foobar that takes one argument, invoked via `foobar(someval)`
[foobar(1)]
args = foo
# note this is definition will include the leading and trailing quotes, i.e.
# something `foobar(someval)`
# would expand to
# something "foo = someval"
definition = "foo = $foo$"

# macro that takes two arguments
# note that macro arguments can be named so this particular macro could be
# invoked equivalently as `foobar(1,2)` `foobar(foo=1,bar=2)` or
# `foobar(bar=2,foo=1)`
[foobar(2)]
args = foo, bar
definition = "foo = $foo$, bar = $bar$"

# macro that takes one argument that does validation
[foovalid(1)]
args = foo
definition = "foovalid = $foo$"
# the validation eval function takes any even number of arguments (>=2) where
# the first argument is a boolean expression, the 2nd a string, the third
# boolean, 4th a string, etc etc etc
validation = validate(foo>15,"foo must be greater than 15",foo<=100,"foo must be <= 100")

# macro showing simple boolean validation, where if foo > bar is not true,
# errormsg is displayed
[foovalid(2)]
args = foo, bar
```

```
definition = "foo = $foo$ and bar = $bar$"
validation = foo > bar
errormsg = foo must be greater than bar

# example of an eval-based definition.  For example in this case
# `fooeval(10,20)` would get replaced by 10 + 20
[fooeval(2)]
args = foo, bar
definition = if (bar > 0, "$foo$ + $bar$", "$foo$ - $bar$")
iseval = true
```

messages.conf

The following are the spec and example files for `messages.conf`.

messages.conf.spec

```
#   Version 9.2.2
#
# This file contains attribute/value pairs for configuring externalized strings
# in messages.conf.
#
# There is a messages.conf in $SPLUNK_HOME/etc/system/default/.  To set custom
# configurations, place a messages.conf in $SPLUNK_HOME/etc/system/local/. You
# must restart the instance to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# For the full list of all messages that can be overridden, check out
# $SPLUNK_HOME/etc/system/default/messages.conf
#
# The full name of a message resource is component_key + ':' + message_key.
# After a descriptive message key, append two underscores, and then use the
# letters after the % in printf style formatting, surrounded by underscores.
#
# For example, assume the following message resource is defined:
#
#   [COMPONENT:MSG_KEY__D_LU_S]
#   message = FunctionX returned %d, expected %lu.
#   action  = See %s for details.
#
# The message key expects 3 printf-style arguments: %d, %lu, %s. These arguments
# can be in either the message or action fields but must appear in the same order.
#
# In addition to the printf style arguments above, some custom UI patterns are
# allowed in the message and action fields. These patterns are rendered by
# the UI before displaying the text.
#
# For example, a message can link to a specific Splunk Web page using this pattern:
#
#   [COMPONENT:MSG_LINK__S]
#   message = License key '%s' is invalid.
#   action  = See [[/manager/system/licensing|Licensing]] for details.
#
# Another custom formatting option is for date/time arguments. If the argument
# should be rendered in local time and formatted to a specific language,
# provide the unix timestamp and prefix the printf style argument with "$t".
```

```
# This indicates that the argument is a timestamp (not a number) and
# should be formatted into a date/time string.
#
# The language and timezone used to render the timestamp is determined during
# render time given the current user viewing the message. It is not required to
# provide these details here.
#
# For example, assume the following message resource is defined:
#
#   [COMPONENT:TIME_BASED_MSG__LD]
#   message = Component exception @ $t%ld.
#   action  = See splunkd.log for details.
#
# The first argument is prefixed with "$t", and therefore will be treated as a
# unix timestamp. It will be formatted as a date/time string.
#
# For these and other examples, check out
# $SPLUNK_HOME/etc/system/README/messages.conf.example
#


################################################################################
# Component
################################################################################
```

[<component>]

```
name = <string>
* The human-readable name used to prefix all messages under this component.
* Required.
* No default.

################################################################################
# Message
################################################################################
```

[<component>:<key>]

```
message = <string>
* String describing what and why something happened.
* Required.

message_alternate = <string>
* An alternative static string for this message.
* Any arguments are ignored.
* Default: empty string

action = <string>
* A string that describes the suggested next step to take in reaction
  to the message.
* Default: empty string

severity = critical|error|warn|info|debug
* The severity of the message.
* Default: warn

capabilities = <comma-separated list>
* A comma-separated list of the capabilities required to view the message.
* Default: empty string
```

```
roles = <comma-separated list>
* A comma-separated list of the roles required to view the message.
* If a user belongs to any of these roles, the user will see the message.
* If a role scope is specified with this setting, it takes precedence over the
  "capabilities" setting, which is ignored for the message.
* This setting should be manually configured with any system- or user-created
  role.
* Default (Splunk Enterprise): not set

help = <string>
* The location string to link users to specific documentation.
* No default.

target = [auto|ui|log|ui,log|none]
* Sets the message display target.
  * "auto" means the message display target is automatically determined by
    context.
  * "ui" messages are displayed in Splunk Web and can be passed on from
    search peers to search heads in a distributed search environment.
  * "log" messages are displayed only in the log files for the instance under
    the BulletinBoard component, with log levels that respect their message
    severity. For example, messages with severity "info" are displayed as INFO
    log entries.
  * "ui,log" combines the functions of the "ui" and "log" options.
  * "none" completely hides the message. (Please consider using "log" and
    reducing severity instead. Using "none" might impact diagnosability.)
* Default: auto
```

messages.conf.example

```
#   Version 9.2.2
#
# This file contains an example messages.conf of attribute/value pairs for
# configuring externalized strings.
#
# There is a messages.conf in $SPLUNK_HOME/etc/system/default/.  To set custom
# configurations, place a messages.conf in $SPLUNK_HOME/etc/system/local/. You
# must restart the instance to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# For the full list of all literals that can be overridden, check out
# $SPLUNK_HOME/etc/system/default/messages.conf


[DISK_MON]
name = Disk Monitor

[DISK_MON:INSUFFICIENT_DISK_SPACE_ERROR__S_S_LLU]
message      = Cannot write data to index path '%s' because you are low on disk space on partition '%s'.
Indexing has been paused.
action       = Free disk space above %lluMB to resume indexing.
severity     = warn
capabilities = indexes_edit
help         = learnmore.indexer.setlimits
```

```
[LM_LICENSE]
name = License Manager

[LM_LICENSE:EXPIRED_STATUS__LD]
message      = Your license has expired as of $t%ld.
action       = $CONTACT_SPLUNK_SALES_TEXT$
capabilities = license_edit

[LM_LICENSE:EXPIRING_STATUS__LD]
message      = Your license will soon expire on $t%ld.
action       = $CONTACT_SPLUNK_SALES_TEXT$
capabilities = license_edit

[LM_LICENSE:INDEXING_LIMIT_EXCEEDED]
message      = Daily indexing volume limit exceeded today.
action       = See [[/manager/search/licenseusage|License Manager]] for details.
severity     = warn
capabilities = license_view_warnings
help         = learnmore.license.features

[LM_LICENSE:MASTER_CONNECTION_ERROR__S_LD_LD]
message      = Failed to contact license master: reason='%s', first failure time=%ld ($t%ld).
severity     = warn
capabilities = license_edit
help         = learnmore.license.features

[LM_LICENSE:SLAVE_WARNING__LD_S]
message      = License warning issued within past 24 hours: $t%ld.
action       = Please refer to the License Usage Report view on license master '%s' to find out more.
severity     = warn
capabilities = license_edit
help         = learnmore.license.features
```

metric_alerts.conf

The following are the spec and example files for `metric_alerts.conf`.

metric_alerts.conf.spec

```
#   Version 9.2.2
#
# This file contains possible setting/value pairs for metric alert entries in the
# metric_alerts.conf file. You can configure metric alerts by creating your own
# metric_alerts.conf file.
#
# There is a default metric_alerts.conf file in $SPLUNK_HOME/etc/system/default. To
# set custom configurations, place a metric_alerts.conf file in
# $SPLUNK_HOME/etc/system/local/. For examples, see the
# metric_alerts.conf.example file. You must restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top of
#    the file.
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, settings are combined. In the case of multiple
#    definitions of the same settings, the last definition in the file wins.
#  * If a setting is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.

#*******
# The possible settings for the metric_alerts.conf file are:
#*******
```

[<alert_name>]

```
* The <alert_name> is the name of the metric alert.
* Required.

description = <string>
* This string provides a description of the metric alert.
* Optional.
* No default.

groupby = <list of dimension fields>
* The list of dimension fields, delimited by comma, for the group-by clause of
  the alert search.
* This leads to multiple aggregation values, one per group, instead of one
  single value.
* Optional.
* No default.

filter = <string>
* This setting provides one or more Boolean expressions like
  '<dimension_field>=<value>' to filter the search result dataset to monitor
  for the alert condition.
* Link multiple Boolean expressions with the 'AND' operator.
* The filter does not support subsearches, macros, tags, event types, or time
  modifiers such as 'earliest' or 'latest'.
* This setting combines with the metric_indexes setting to provide the full alert
  search filter.
* Optional.
* No default.

metric_indexes = <metric index name>
* Specifies one or more metric indexes, delimited by comma.
* Combines with the filter setting to filter the search result dataset to monitor
  for the alert condition.
* Required.
* No default.

condition = <boolean eval expression>
* Specifies an alert condition for one or more metric_name and aggregation
  pairs. The Splunk software applies this evaluation to the results of the
  alert search on a regular interval. This alert search takes the form of
  an 'mstats' search.
* When the alert condition evaluates to 'true', the Splunk software might trigger
```

```
  the alert, depending on how 'trigger.threshold' and 'trigger.suppress' are
  evaluated.
* The condition must reference at least one metric aggregation in single
  quotes: '<mstats_aggregation_function>(<metric_name>)'
* The condition can also reference dimensions specified in the group-by fields.
* Dimension field names starting with numeric characters or with non-alphanumeric
  characters must be surrounded by single quotation marks.
* If the expression references a literal string, the literal string must be
  surrounded by double quotation marks.
* Required.
* No default.

trigger.prepare = <string>
* Specifies a postprocessing search that the Splunk software applies to the
  filtered results of the alert search, before it runs the designated alert
  actions.
* Use this postprocessing search to augment or filter the filtered results of
  the alert search.
  * Employ commands like 'eval' or 'inputlookup' to rename existing fields in
    the results or add new fields to the results.
  * Design filters that remove unnecessary events from the result dataset used
    by the alert action.
* Optional.
* No default.

trigger.suppress = <time-specifier>
* Specifies the suppression period to silence alert actions and notifications.
  * The suppression period goes into effect when an alert is triggered.
  * During this period, if the alert is triggered again, its actions do not happen
    and its notifications do not go out.
  * When the period elapses, a subsequent triggering of the alert causes alert
    actions and notifications to take place as usual, and the alert is
    suppressed again.
* Use [number]m to specify a timespan in minutes.
* Set to 0 to disable suppression.
* Default: 0

trigger.expires = <time-specifier>
* Sets the period of time that a triggered alert record displays on the
  Triggered Alerts page.
* Use [positive integer][time-unit], where time_unit can be 'm' for minutes,
  'h' for hours, and 'd' for days.
* Set to 0 to make triggered alerts expire immediately so they do not appear on
  the Triggered Alerts page at all.
* Default: 24h

trigger.max_tracked = <number>
* Specifies the maximum number of instances of this alert that can display in
  the triggered alerts dashboard.
* When this threshold is passed, the Splunk software removes the earliest
  instances from the dashboard to honor this maximum number.
* Set to 0 to remove the cap.
* Default: 20

trigger.evaluation_per_group = <boolean>
* Optional.
* Only applies if 'groupby' is set.
* When set to true, the Splunk software independently evaluates the alert
  'condition', 'trigger.threshold', and 'trigger.suppress' settings against
  each result, in correspondence with a unique group of dimension field values
  defined by the 'groupby' setting.
* Use 'trigger.evaluation_per_group' in conjunction with the
```

```
  'trigger.action_per_group' setting.
* Default: false

trigger.action_per_group = <boolean>
* Optional.
* Only applies if 'groupby' and 'trigger.evaluation_per_group' are set.
* When set to true, the Splunk software runs actions for each result, in
  correspondence with a unique group of dimension field values defined by the
  'groupby' setting, using the evaluations produced by the
  'trigger.evaluation_per_group' setting.
* When 'trigger.evaluation_per_group' is enabled and this setting is disabled,
  the Splunk software runs the alert action only once when one or more groups
  meet the alert condition.
* This setting cannot be enabled when 'trigger.evaluation_per_group'
  is disabled.
* Default: false

trigger.threshold = [always|once|always after <number>m|once after <number>m]
* Specify when to perform an alert action such as sending an email:
  * always - Whenever the alert 'condition' is true.
  * once - Only once, the first time the alert 'condition' makes a positive
    state change from false to true.
  * always after <number>m - Whenever the alert 'condition' is met continuously
    for <number> minutes.
  * once after <number>m - Only once, the first time the alert 'condition' is
    met continuously for <number> minutes.
* Examples:
  * A setting of 'always after 5m' means that the Splunk software performs the
    alert action every time the alert condition is met for 5 minutes in a row.
    So if the alert condition is true for 8 minutes, the Splunk software
    performs the action 3 times.
  * A setting of 'once after 5m' means that the Splunk software performs the
    alert action the first time the alert condition is met for 5 minutes in a
    row. If the alert condition is met continuously for 8 minutes the Splunk
    software performs the action only once. If after that, the condition
    switches to false and is then true continuously for another 12 minutes, the
    Splunk software would perform the action again.
* Default: always

label.<label-name> = <label-value>
* Arbitrary key-value pairs for labeling this alert.
* These settings will be opaque to the backend (not interpreted in any way).
* Can be used by applications calling `alerts/metric_alerts` endpoint.

splunk_ui.<label-name> = <label-value>
* For Splunk internal use only.
* Arbitrary key-value pairs for labeling this alert for the exclusive use of
  the Splunk software.

splunk_ui.track = <boolean>
* Optional.
* Indicates whether the alert is tracked on the Triggered Alerts page and the
  Splunk Analytics Workspace.
* Defaults: false

splunk_ui.severity = <integer>
* Optional.
* Sets the severity level displayed for the alert in Splunk Web.
* Valid values are: 1-debug, 2-info, 3-warn, 4-error, 5-severe, 6-fatal
* Default: 3

#*******
```

```
# generic action settings.
# For a comprehensive list of actions and their arguments, refer to the
# alert_actions.conf file.
#*******

action.<action_name> = <boolean>
* Indicates whether the action is enabled or disabled for a particular metric
  alert.
* The 'action_name' can be: email | logevent | rss | script | webhook
* For more about the defined alert actions see the alert_actions.conf file.
* Optional.
* No default.

action.<action_name>.<parameter> = <value>
* Overrides an action's parameter as defined in the alert_actions.conf file,
  with a new <value> for this metric alert only.
* No default.

action.email.include.smaDefinition = [1|0]
* Specify whether to include streaming alert setup information in the email content.
* Setup information includes indexes, filter, groupby, condition.
```

metric_alerts.conf.example

```
#   Version 9.2.2
#
# This file contains example metric alerts.
#
# To use one or more of these configurations, copy the configuration block into
# metric_alerts.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# The following searches are example searches.  To create your own search,
# modify the values by following the spec outlined in metric_alerts.conf.spec.


[alert1]
groupby = host, app
filter = region=east
condition = 'avg(mem.used)' > 50
action.email = 1
action.email.to = nonexist@abc.xyz

[alert2]
groupby = host, app
filter = region=east
condition = 'max(cpu.util)' > 80
action.email = 1
action.email.to = nonexist@abc.xyz
```

metric_rollups.conf

The following are the spec and example files for `metric_rollups.conf`.

metric_rollups.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attribute/value pairs for rollup policy entries in
# metric_rollups.conf.  You can configure rollup policies by creating your own
# metric_rollups.conf.
#
# There is a default metric_rollups.conf in $SPLUNK_HOME/etc/system/default. To
# set custom configurations, place a metric_rollups.conf in
# $SPLUNK_HOME/etc/system/local/.  For examples, see
# metric_rollups.conf.example. You must restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top of
#    the file.
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, attributes are combined. In the case of multiple
#    definitions of the same attribute, the last definition in the file wins.
#  * If an attribute is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.

#*******
# The possible attribute/value pairs for metric_rollups.conf are:
#*******
```

[index:<Metric Index Name>]

```
* Each metric_rollups.conf stanza defines the rollup summarization policy for a
  specific metric index.
* A rollup policy can include multiple rollup summaries, each with a
  different rollup period.
* Go to indexes.conf to find metric index configurations. Metric indexes have
  datatype=metric in their configurations.

defaultAggregation = <'#' separated list of aggregation functions>
* Required. The default aggregation function for the rollup policy. The Splunk
  software uses this aggregation function to generate the rollup summmary data
  points for all metrics in the source index with the exception of metrics that
  are identified by 'aggregation.<metric_name>'
  exclusion rules.
* For example, if a rollup summary with a period of 1 hour has
  'defaultAggregation = avg', each metric data point that it generates is the
  average of an hour of data points from the source metric.
```

```
* Note that the 'perc' and 'upperperc' options require an integer.
* Supported aggregation functions: [avg|count|max|median|min|perc<int>|sum]
* Default: avg

dimensionList = <comma-separated list of dimensions>
* Optional. This setting provides a comma-separated list of dimensions. The
  dimensions must be present within the index to which the rollup policy
  applies.
* This list corresponds to the `dimensionListType` setting, which determines
  whether this set of dimensions is included or excluded from the rollup
  metrics that are generated by the rollup summary.
* Use the Metrics Catalog REST API endpoints to see the metrics and dimensions
  for a particular index. For more information see the REST API Reference
  Manual.
* Default: not set

dimensionListType = [excluded|included]
* Optional. This setting determines whether the list of dimensions specified by
  the `dimensionList` setting is included or excluded from the rollup metrics
  that are generated by the rollup summaries in the rollup policy.
* Select 'included' to indicate that the rollup metrics produced by the rollup
  policy will filter out all dimensions except the ones in the list.
* Select 'excluded' to indicate that the rollup metrics produced by the rollup
  policy will include all available dimensions except the ones in the list.
* Default: excluded

metricList = <comma-separated list of metrics>
* Optional. This setting provides a comma-separated list of metrics.
* This list corresponds to the 'metricListType' setting.
* The listed metrics must be present within the source metric index.
  * Use the Metrics Catalog REST API endpoints in conjunction with the 'rest'
    command to see the metrics that exist within a particular source index. See
    the REST API Reference Manual and the Search Reference for more information.
* Default: not set

metricListType = <excluded/included>
* Optional. This setting determines whether the list of metrics specified by
  the 'metricList' setting is included or excluded when the search head rolls
  metrics up to the rollup summaries.
* Select "included" to have the search head roll up only the listed metrics.
* Select "excluded" to have the search head roll up all available metrics in
  the source metric index except the listed metrics.
* Default: excluded

aggregation.<metric_name> = <'#' separated list of aggregation functions>
* Optional. Sets an exclusion rule for a rollup policy. Use this setting to
  override the 'defaultAggregation' setting for a specific metric.
* Create exclusion rules for metrics that require different aggregation
  functions than the majority of the metrics in a rollup policy.
* A single rollup policy can have multiple exclusion rules.
* Supported aggregation functions: [avg|count|max|median|min|perc<int>|sum]
* Default: no values

rollup.<summary number>.span = <time range string>
* Required for each rollup summary in the rollup policy.
* The Splunk software defines the '<summary number>' when you create a summary
  policy through Splunk Web or the REST API endpoint.
* Defines the rollup period for a rollup summary.
* The '<time range string>' cannot be shorter than the 'minSpanAllowed' setting
  in limits.conf.
* This setting is required. Do not leave it blank.
* Default for <summary number>: 1
```

```
* Default for <time range string>: 1h

rollup.<summary number>.rollupIndex = <string Index name>
* Required for each rollup summary in the rollup policy.
* Defines the target index for the rollup metrics generated by a rollup summary.
* The Splunk software defines the '<summary number>' when you create a summary
  policy through Splunk Web or the REST API endpoint.
* The index name must exist in indexes.conf.
* This setting is required. Do not leave it blank.
* Default for <summary number>: 1
* Default for <string Index name>: The <Metric Index Name> in the stanza header
  for this rollup policy.
```

metric_rollups.conf.example

```
#   Version 9.2.2
#
# This file contains example saved searches and alerts.
#
# To use one or more of these configurations, copy the configuration block into
# metric_rollups.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# The following searches are example searches.  To create your own search,
# modify the values by following the spec outlined in metric_rollups.conf.spec.

[index:mySourceMetricIndex]
# defaultAggregation is applied to all the measures/metric names unless overided
defaultAggregation = avg
# Override metric_name_1 aggregation from avg to min
aggregation.metric_name_1 = min
# Override metric_name_2 aggregation from avg to count
aggregation.metric_name_2 = count
# Exclude dimension_1 and dimension_2 during rollup
dimensionList = dimension_1, dimension_2
dimensionListType = excluded
# All the above settings applies globally to all the summary definitions below
# Each summary here specifies the target index and span
# Two summaries definied, need to define each summary as rollup.<0, 1, 2..>...
rollup.0.rollupIndex = myTargetMetricIndex_0
rollup.0.span = 1h
rollup.1.rollupIndex = myTargetMetricIndex_1
rollup.1.span = 1d
# Exclude metric_1 and metric_2 during rollup
metricList = metric_1, metric_2
metricListType = excluded
```

multikv.conf

The following are the spec and example files for `multikv.conf`.

multikv.conf.spec

```
#   Version 9.2.2
#
# This file contains descriptions of the settings that you can use to
# create multikv rules.  Multikv is the process of extracting events
# from table-like events, such as the output of top, ps, ls, netstat, etc.
#
# To set custom configurations, create a new file with the name multikv.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see multikv.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# NOTE: Only configure multikv.conf if the default multikv behavior does
# not meet your needs.

# A table-like event includes a table consisting of four sections:
#
```

Section Name | Description

```
# pre          | optional: info/description (for example: the system summary output in top)
# header       | optional: if not defined, fields are named Column_N
# body         | required: the body of the table from which child events are constructed
# post         | optional: info/description
#-------------------------------------------------------------------------------

# NOTE: Each section must have a definition and a processing component. See
# below.

[<multikv_config_name>]
* Name of the stanza to use with the multikv search command, for example:
       '| multikv conf=<multikv_config_name> rmorig=f | ....'
* Follow this stanza name with any number of the following setting/value pairs.
```

Section Definition

```
# Define where each section begins and ends.

<Section Name>.start = <regex>
* A line matching this regex denotes the start of this section (inclusive).

OR

<Section Name>.start_offset = <int>
* Line offset from the start of an event or the end of the previous section
  (inclusive).
* Use this if you cannot define a regex for the start of the section.

<Section Name>.member = <regex>
* A line membership test.
```

```
* Member if lines match the regex.

<Section Name>.end = <regex>
* A line matching this regex denotes the end of this section (exclusive).

OR

<Section Name>.linecount = <int>
* Specify the number of lines in this section.
* Use this if you cannot specify a regex for the end of the section.
```

Section processing

```
# Set processing for each section.

<Section Name>.ignore = [_all_|_none_|_regex_ <regex-list>]
* Determines which member lines will be ignored and not processed further.

<Section Name>.replace = <quoted-str> = <quoted-str>, <quoted-str> = <quoted-str>,...
* List of the form: "toReplace" = "replaceWith".
* Can have any number of quoted string pairs.
* For example: "%" = "_", "#" = "_"

<Section Name>.tokens = [<chopper>|<tokenizer>|<aligner>|<token-list>]
* See below for definitions of each possible token: chopper, tokenizer, aligner,
  and token-list.

<chopper> = _chop_, <int-list>
* A token that transform each string into a list of tokens specified by
  <int-list>.
* <int-list> is a list of (offset, length) tuples, separated by commas. Do not
  contain tuples within parentheses.
  * Example: body.tokens = _chop_, 0, 9,  10, 4,  15, 4,  20, 7


<tokenizer> = _tokenize_ <max_tokens (int)> <delims> (<consume-delims>)?
* A token used to tokenize the string using the delimiter characters.
* This generates at most 'max_tokens' number of tokens.
* Set 'max_tokens' to:
  * -1 for complete tokenization.
  * 0 to inherit from the previous section, usually the header section.
  * A non-zero number for a specific token count.
* If tokenization is limited by the 'max_tokens', the rest of the string is
  added onto the last token.
* <delims> is a comma-separated list of delimiting characters.
* <consume-delims> - A Boolean that specifies whether to consume consecutive
  delimiters. Set to "false" or "0" if you want consecutive delimiters treated
  as empty values.
* Default: true

<aligner> = _align_, <header_string>, <side>, <max_width>
* A token that generates tokens by extracting text aligned to the specified header fields.
* header_string: A complete or partial header field value that the columns
  are aligned with.
* side: Either L or R (for left or right align, respectively).
* max_width: The maximum width of the extracted field.
  * Set 'max_width' to -1 for automatic width. This expands the field until any
    of the following delimiters are found: " ", "\t"

<token_list> = _token_list_ <comma-separated list>
```

```
* A token that defines a list of static tokens in a section.
* This setting is useful for tables with no header,
  for example: the output of 'ls -lah' which misses a header altogether.
```

multikv.conf.example

```
#   Version 9.2.2
#
# This file contains example multi key/value extraction configurations.
#
# To use one or more of these configurations, copy the configuration block into
# multikv.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# This example breaks up the output from top:

# Sample output:

# Processes: 56 total, 2 running, 54 sleeping... 221 threads 10:14:07
#.....
#
#   PID COMMAND  %CPU TIME     #TH #PRTS #MREGS RPRVT RSHRD RSIZE  VSIZE
# 29960 mdimport 0.0%  0:00.29  3    60    50  1.10M  2.55M 3.54M  38.7M
# 29905 pickup   0.0%  0:00.01  1    16    17   164K   832K  764K  26.7M
#....

[top_mkv]
# pre table starts at "Process..." and ends at line containing "PID"
pre.start = "Process"
pre.end = "PID"
pre.ignore = _all_

# specify table header location and processing
header.start = "PID"
header.linecount = 1
header.replace = "%" = "_", "#" = "_"
header.tokens = _tokenize_, -1," "

# table body ends at the next "Process" line (ie start of another top) tokenize
# and inherit the number of tokens from previous section (header)
body.end = "Process"
body.tokens  = _tokenize_,  0, " "



## This example handles the output of 'ls -lah' command:
#
# total 2150528
# drwxr-xr-x 88 john john 2K   Jan 30 07:56 .
# drwxr-xr-x 15 john john 510B Jan 30 07:49 ..
# -rw------- 1  john john 2K   Jan 28 11:25 .hiden_file
# drwxr-xr-x 20 john john 680B Jan 30 07:49 my_dir
# -r--r--r-- 1  john john 3K   Jan 11 09:00 my_file.txt
```

```
[ls-lah-cpp]
pre.start     = "total"
pre.linecount = 1

# the header is missing, so list the column names
header.tokens = _token_list_, mode, links, user, group, size, date, name

# The ends when we have a line starting with a space
body.end     = "^\s*$"
# This filters so that only lines that contain with .cpp are used
body.member  = "\.cpp"
# concatenates the date into a single unbreakable item
body.replace = "(\w{3})\s+(\d{1,2})\s+(\d{2}:\d{2})" ="\1_\2_\3"

# ignore dirs
body.ignore = _regex_ "^drwx.*",
body.tokens  = _tokenize_, 0, " "
```

outputs.conf

The following are the spec and example files for `outputs.conf`.

outputs.conf.spec

```
#   Version 9.2.2
#
# Forwarders require outputs.conf. Splunk instances that do not forward
# do not use it. Outputs.conf determines how the forwarder sends data to
# receiving Splunk instances, either indexers or other forwarders.
#
# To configure forwarding, create an outputs.conf file in
# $SPLUNK_HOME/etc/system/local/. For examples of its use, see
# outputs.conf.example.
#
# You must restart the Splunk software to enable configurations.
#
# To learn more about configuration files (including precedence) see the topic
# "About Configuration Files" in the Splunk Enterprise Admin manual.
#
# To learn more about forwarding, see the topic "About forwarding and
# receiving data" in the Splunk Enterprise Forwarding manual.
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top
#     of the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file wins.
#   * If an setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
#   * Do not use the 'sslPassword', 'socksPassword', or 'token' settings
```

```
#     to set passwords in this stanza as they may remain readable to
#     attackers, specify these settings in the [tcpout] stanza instead.
```

TCP Output stanzas

```
# There are three levels of TCP Output stanzas:
# * Global: [tcpout]
# * Target group: [tcpout:<target_group>]
# * Single server: [tcpout-server://<ip address>:<port>]
#
# Settings at more specific levels override settings at higher levels. For
# example, an setting set for a single server overrides the value of that
# setting, if any, set at that server's target group stanza. See the
# online documentation on configuring forwarders for details.
#
# This spec file first describes the three levels of stanzas (and any
# settings unique to a particular level). It then describes the optional
# settings, which you can set at any of the three levels.
# Default: true
# If set to 'true', prevents the logs from being forwarded to the indexing tiers.

[httpout]

httpEventCollectorToken = <string>
* The value of the HEC token.
* HEC uses this token to authenticate inbound connections.
* No default.

uri = <string>
* The URI and management port of the Http Event Collector(HEC) end point.
* For example, https://SplunkHEC01.example.com:8088
* No default.

batchSize = <integer>
* The size, in bytes, of the HTTP OUT send buffer.
* HTTP OUT batch pipeline data before sending out.
* If the current buffer size is greater than 'batchSize', HEC sends the data
  out immediately.
* Default: 65536

batchTimeout = <integer>
* How often, in seconds, to send out pipeline data.
* HTTP OUT batch pipeline data before sending out.
* If the wait time is greater than 'batchTimeout', HEC sends the data
  out immediately.
* Default: 30

#----TCP Output Global Configuration -----
# You can overwrite the global configurations specified here in the
# [tcpout] stanza in stanzas for specific target groups, as described later.
# You can only set the 'defaultGroup' and 'indexAndForward' settings
# here, at the global level.
#
# Starting with version 4.2, the [tcpout] stanza is no longer required.

[tcpout]

defaultGroup = <comma-separated list>
* A comma-separated list of one or more target group names, specified later
```

```
  in [tcpout:<target_group>] stanzas.
* The forwarder sends all data to the specified groups.
* If you don't want to forward data automatically, don't configure this setting.
* Can be overridden by the '_TCP_ROUTING' setting in the inputs.conf file,
  which in turn can be overridden by a props.conf or transforms.conf modifier.
* Starting with version 4.2, this setting is no longer required.

indexAndForward = <boolean>
* Set to "true" to index all data locally, in addition to forwarding it.
* This is known as an "index-and-forward" configuration.
* This setting is only available for heavy forwarders.
* This setting is only available at the top level [tcpout] stanza. It
  cannot be overridden in a target group.
* Default: false

enableOldS2SProtocol = <boolean>
* Whether or not the forwarder enables use of versions 3 and lower of the Splunk-to-Splunk protocol,
  otherwise known as the "old" S2S protocol, to connect with other Splunk platform instances.
* A value of "true" means the forwarder can use the old protocol, depending on other settings
  you configure.
* A value of "false" means the forwarder uses only version 4 of the S2S protocol,
  and does not use any of the old protocol versions.
* When you disable the use of old S2S protocols, forwarders always use the new protocol. Indexers
  that run a version of Splunk Enterprise below 6.0 only support the old protocol, and forwarders
  can't connect to those indexers over S2S.
* If you give 'negotiateProtocolLevel' a value of 0, or 'negotiateNewProtocol' a value of
  "false" in inputs.conf or outputs.conf to use the old S2S protocol, the forwarder will instead override
these
  settings to use the lowest protocol version that all instances support.
* This setting is only available for configuration at the top level [tcpout] stanza. You
  can't override it in a target group with settings that force usage of the older protocol.
* Default: false

#----Target Group Configuration -----

# If you specify multiple servers in a target group, the forwarder
# performs auto load-balancing, sending data alternately to each available
# server in the group. For example, assuming you have three servers
# (server1, server2, server3) and autoLBFrequency=30, the forwarder sends
# all data to server1 for 30 seconds, then it sends all data to server2 for
# the next 30 seconds, then all data to server3 for the next 30 seconds,
# finally cycling back to server1.
#
# You can have as many target groups as you want.
# If you specify more than one target group, the forwarder sends all data
# to each target group. This is known as "cloning" the data.
#
# NOTE: A target group stanza name cannot contain spaces or colons.
# Splunk software ignores target groups whose stanza names contain
# spaces or colons.

[tcpout:<target_group>]

server = <comma-separated list>
* A comma-separated list of one or more systems to send data to over a
  TCP socket.
* You can specify each element as either an IP address or a hostname
  and a port number. For example: 192.168.1.10:9997, mysplunkserver.com:9997
* Required if the 'indexerDiscovery' setting is not set.
* Typically used to specify receiving Splunk systems, although you can use
  it to send data to non-Splunk systems (see the 'sendCookedData' setting).
* For each system you list, the following information is required:
```

```
  * The IP address or server name where one or more systems are listening.
  * The port on which the syslog server is listening.

blockWarnThreshold = <integer>
* The output pipeline send failure count threshold after which a
  failure message appears as a banner in Splunk Web.
* Optional.
* To disable Splunk Web warnings on blocked output queue conditions, set this
  to a large value (for example, 2000000).
* Default: 100

indexerDiscovery = <string>
* The name of the manager node to use for indexer discovery.
* Instructs the forwarder to fetch the list of indexers from the manager node
  specified in the corresponding [indexer_discovery:<name>] stanza.
* No default.

token = <string>
* The access token for receiving data.
* If you configured an access token for receiving data from a forwarder,
  Splunk software populates that token here.
* If you configured a receiver with an access token and that token is not
  specified here, the receiver rejects all data sent to it.
* This setting is optional.
* No default.

#----Single server configuration-----

# You can define specific configurations for individual indexers on a
# server-by-server basis. However, each server must also be part of a
# target group.

[tcpout-server://<ip address>:<port>]
* Optional. There is no requirement to have a [tcpout-server] stanzas.
```

TCPOUT SETTINGS

```
# These settings are optional and can appear in any of the three stanza levels.

[tcpout<any of above>]

#----General Settings----

disabled = <boolean>
* Whether or not to disable forwarding to the receiver or output group, as defined by the forwarding stanza.
* Set to true to disable forwarding to this receiver or output group.
* Default: false

sendCookedData = <boolean>
* Whether or not to send processed or unprocessed data to the receiving server.
* A value of "true" means Splunk software processes the events before sending them
  to the server, thus "cooking" them.
* A value of "false" means events are raw and untouched prior to sending.
* Set to "false" if you are sending events to a third-party system.
* Default: true

heartbeatFrequency = <integer>
* How often, in seconds, to send a heartbeat packet to the receiving server.
```

```
* This setting is a mechanism for the forwarder to know that the receiver
  (indexer) is alive. If the indexer does not send a return packet to the
  forwarder, the forwarder declares the receiver unreachable and does not
  forward data to it.
* The forwarder only sends heartbeats if the 'sendCookedData' setting
  is set to "true".
* Default: 30

blockOnCloning = <boolean>
* Whether or not the TcpOutputProcessor should wait until at least one
  of the cloned output groups receives events before attempting to send
  more events.
* If set to "true", the TcpOutputProcessor blocks until at least one of the
  cloned groups receives events. It does not drop events when all the
  cloned groups are down.
* If set to "false", the TcpOutputProcessor drops events when all the
  cloned groups are down and all queues for the cloned groups are full.
  When at least one of the cloned groups is up and queues are not full,
  the events are not dropped.
* Default: true

blockWarnThreshold = <integer>
* The output pipeline send failure count threshold, after which a
  failure message appears as a banner in Splunk Web.
* To disable Splunk Web warnings on blocked output queue conditions, set this
  to a large value (for example, 2000000).
* This setting is optional.
* Default: 100

compressed = <boolean>
* Whether or not forwarders and receivers communicate with one another in
  compressed format.
* A value of "true" means the receiver communicates with the forwarder in
  compressed format for forwarding that does not use TLS/SSL.
* A value of "true" means the receiver communicates with the forwarder in
  compressed format for TLS/SSL forwarding if either
  'useClientSSLCompression' has a value of "false" or the TLS/SSL
  connection does not use 'zlib' compression.
* If set to "true", you do not need to set the 'compressed' setting to
  "true" in the inputs.conf file on the receiver for compression of data
  to occur.
* If you use this setting, the 'tcpout_connections' group in the metrics.log
  file shows throughput values after compression has occurred.
* Default: false

negotiateProtocolLevel = <unsigned integer>
* When setting up a connection to an indexer, Splunk software tries to
  negotiate the use of the Splunk forwarder protocol with the
  specified feature level based on the value of this setting.
* If set to a lower value than the default, this setting denies the
  use of newer forwarder protocol features when it negotiates a connection.
  This might impact indexer efficiency.
* Default (if 'negotiateNewProtocol' is "true"): 1
* Default (if 'negotiateNewProtocol' is not "true"): 0

negotiateNewProtocol = <boolean>
* The default value of the 'negotiateProtocolLevel' setting.
* DEPRECATED. Set 'negotiateProtocolLevel' instead.
* Default: true

channelReapInterval = <integer>
* How often, in milliseconds, that channel codes are reaped, or made
```

```
  available for re-use.
* This value sets the minimum time between reapings. In practice,
  consecutive reapings might be separated by greater than the number of
  milliseconds specified here.
* Default: 60000 (1 minute)

channelTTL = <integer>
* How long, in milliseconds, a channel can remain "inactive" before
  it is reaped, or before its code is made available for reuse by a
  different channel.
* Default: 300000 (5 minutes)

channelReapLowater = <integer>
* This value essentially determines how many active-but-old channels Splunk
  software keeps "pinned" in memory on both sides of a
  Splunk-to-Splunk connection.
* If the number of active channels is greater than 'channelReapLowater',
  Splunk software reaps old channels to make their channel codes available
  for re-use.
* If the number of active channels is less than 'channelReapLowater',
  Splunk software does not reap channels, no matter how old they are.
* A non-zero value helps ensure that Splunk software does not waste network
  resources by "thrashing" channels in the case of a forwarder sending
  a trickle of data.
* Default: 10

socksServer = <string>
* The IP address or server name of the Socket Secure version 5 (SOCKS5) server.
* Required. Specify this value as either an IP address or hostname and port
  number, for example: 192.168.1.10:8080 or mysplunkserver.com:8080.
* This setting specifies the port on which the SOCKS5 server is listening.
* After you configure and restart the forwarder, it connects to the SOCKS5
  proxy host, and optionally authenticates to the server on demand if
  you provide credentials.
* NOTE: Only SOCKS5 servers are supported.
* No default.

socksUsername = <string>
* The SOCKS5 username to use when authenticating against the SOCKS5 server.
* Optional.

socksPassword = <string>
* The SOCKS5 password to use when authenticating against the SOCKS5 server.
* Optional.

socksResolveDNS = <boolean>
* Whether or not a forwarder should rely on the SOCKS5 proxy server Domain
  Name Server (DNS) to resolve hostnames of indexers in the output group to
  which the forwarder sends data.
* A value of "true" means the forwarder sends the hostnames of the indexers to the
  SOCKS5 server, and lets the SOCKS5 server do the name resolution. It
  does not attempt to resolve the hostnames on its own.
* A value of "false" means the forwarder attempts to resolve the hostnames of the
  indexers through DNS on its own.
* Optional.
* Default: false

#----Queue Settings----

maxQueueSize = [<integer>|<integer>[KB|MB|GB]|auto]
* The maximum size of the forwarder output queue.
* The size can be limited based on the number of entries, or on the total
```

```
  memory used by the items in the queue.
* If specified as a lone integer (for example, "maxQueueSize=100"),
  the 'maxQueueSize' setting indicates the maximum count of queued items.
* If specified as an integer followed by KB, MB, or GB
  (for example, maxQueueSize=100MB), the 'maxQueueSize' setting indicates
  the maximum random access memory (RAM) size of all the items in the queue.
* If set to "auto", this setting configures a value for the output queue
  depending on the value of the 'useACK' setting:
  * If 'useACK' is set to "false", the output queue uses 500KB.
  * If 'useACK' is set to "true", the output queue uses 7MB.
* If you enable indexer acknowledgment by configuring the 'useACK'
  setting to "true", the forwarder creates a wait queue where it temporarily
  stores data blocks while it waits for indexers to acknowledge the receipt
  of data it previously sent.
  * The forwarder sets the wait queue size to triple the value of what
    you set for 'maxQueueSize.'
  * For example, if you set "maxQueueSize=1024KB" and "useACK=true",
    then the output queue is 1024KB and the wait queue is 3072KB.
  * Although the wait queue and the output queue sizes are both controlled
    by this setting, they are separate.
  * The wait queue only exists if 'useACK' is set to "true".
* Limiting the queue sizes by quantity is historical. However,
  if you configure queues based on quantity, keep the following in mind:
  * Queued items can be events or blocks of data.
    * Non-parsing forwarders, such as universal forwarders, send
      blocks, which can be up to 64KB.
    * Parsing forwarders, such as heavy forwarders, send events, which
      are the size of the events. Some events are as small as
      a few hundred bytes. In unusual cases (data dependent), you might
      arrange to produce events that are multiple megabytes.
* Default: auto
  * if 'useACK' is set to "true" and this setting is set to "auto", then
    the output queue is 7MB and the wait queue is 21MB.

dropEventsOnQueueFull = <integer>[ms|s|m]
* The amount of time to wait before the output queue throws out all
  new events until it has space.
* If set to 0ms(milliseconds), 0s(seconds), or 0m(minutes),
  the queue immediately throws out all new events until it has space.
* If set to a positive number, the queue waits the specified number of
  milliseconds, seconds, or minutes before throwing out all new events.
  If "ms", "s", or "m" is not specified, the default unit is seconds.
* If set to -1 or 0, the output queue is blocked because it is full, but events
  are not dropped.
* If any target group queue is blocked, no more data reaches any other
  target group.
* CAUTION: Do not set to a positive integer if you are monitoring files
  because the files will not be fully ingested if the queue remains blocked
  for the specified amount of time.
* Default: -1

dropClonedEventsOnQueueFull = <integer>[ms|s|m]
* The amount of time to wait before dropping events from the group.
* If set to 0ms(milliseconds), 0s(seconds), or 0m(minutes), the queue
  immediately throws out all new events until it has space.
* If set to a positive number, the queue does not block completely, but
  waits up to the specified number of milliseconds, seconds, or minutes to
  queue events to a group.
  * If it cannot queue to a group for more than the specified amount of time,
    the queue begins dropping events from the group and makes sure that at
    least one group in the cloning configuration can receive events.
  * The queue blocks if it cannot deliver events to any of the cloned groups.
```

```
* If set to -1, the TcpOutputProcessor ensures that each group
  receives all of the events. If one of the groups is down, the
  TcpOutputProcessor blocks everything.
* Default: 5 seconds

#######
# Backoff Settings When Unable To Send Events to Indexer
# The settings in this section determine forwarding behavior when there are
# repeated failures in sending events to an indexer ("sending failures").
#######

maxFailuresPerInterval = <integer>
* The maximum number of failures allowed per interval before a forwarder
  applies backoff (stops sending events to the indexer for a specified
  number of seconds). The interval is defined in the 'secsInFailureInterval'
  setting.
* Default: 2

secsInFailureInterval = <integer>
* The number of seconds contained in a failure interval.
* If the number of write failures to the indexer exceeds
  'maxFailuresPerInterval' in the specified 'secsInFailureInterval' seconds,
  the forwarder applies backoff.
* The backoff time period range is 1-10 * 'autoLBFrequency'.
* Default: 1

backoffOnFailure = <positive integer>
* The number of seconds a forwarder backs off, or stops sending events,
  before attempting to make another connection with the indexer.
* Default: 30

maxConnectionsPerIndexer = <integer>
* The maximum number of allowed connections per indexer.
* In the presence of failures, the maximum number of connection attempts
  per indexer at any point in time.
* Default: 2

connectionsPerTarget = [<integer>|auto]
* The maximum number of allowed outbound connections for each target IP address
  as resolved by DNS on the machine.
* A value of "auto" or < 1 means splunkd configures a value for connections for each
  target IP address. Depending on the number of IP addresses that DNS resolves,
  splunkd sets 'connectionsPerTarget' as follows:
  * If the number of resolved target IP addresses is greater than or equal to 10,
    'connectionsPerTarget' gets a value of 1.
  * If the number of resolved target IP addresses is greater than 5
    and less than 10, 'connectionsPerTarget' gets a value of 2.
  * If the number of resolved target IP addresses is greater than 3
    or less than equal to 5, 'connectionsPerTarget' gets a value of 3.
  * If the number of resolved target IP addresses is less than or equal to 3,
    'connectionsPerTarget' gets a value of 4.
* Default: auto

connectionTimeout = <integer>
* The time to wait, in seconds, for a forwarder to establish a connection
  with an indexer.
* The connection times out if an attempt to establish a connection
  with an indexer does not complete in 'connectionTimeout' seconds.
* Default: 20

readTimeout = <integer>
* The time to wait, in seconds, for a forwarder to read from a socket it has
```

```
  created with an indexer.
* The connection times out if a read from a socket does not complete in
  'readTimeout' seconds.
* This timeout is used to read acknowledgment when indexer acknowledgment is
  enabled (when you set 'useACK' to "true").
* Default: 300 seconds (5 minutes)

writeTimeout = <integer>
* The time to wait, in seconds, for a forwarder to complete a write to a
  socket it has created with an indexer.
* The connection times out if a write to a socket does not finish in
  'writeTimeout' seconds.
* Default: 300 seconds (5 minutes)

connectionTTL = <integer>
* The time, in seconds, for a forwarder to keep a socket connection
  open with an existing indexer despite switching to a new indexer.
* This setting reduces the time required for indexer switching.
* Useful during frequent indexer switching potentially caused
  by using the 'autoLBVolume' setting.
* Default: 0 seconds

tcpSendBufSz = <integer>
* The size of the TCP send buffer, in bytes.
* Only use this setting if you are a TCP/IP expert.
* Useful to improve throughput with small events, like Windows events.
* Default: the system default

ackTimeoutOnShutdown = <integer>
* The time to wait, in seconds, for the forwarder to receive indexer
  acknowledgments during a forwarder shutdown.
* The connection times out if the forwarder does not receive indexer
  acknowledgements (ACKs) in 'ackTimeoutOnShutdown' seconds during
  forwarder shutdown.
* Default: 30 seconds

polling_interval = <integer>
* The initial time to wait upon splunk start, in seconds, for the forwarder to fetch
  the list of indexers from the indexer discovery server specified in
  the corresponding [indexer_discovery:<name>] stanza. Subsequently polling interval
  is set by indexer discovery server response.
* Default: 5 seconds

dnsResolutionInterval = <integer>
* The base time interval, in seconds, at which indexer Domain Name Server
  (DNS) names are resolved to IP addresses.
* This is used to compute runtime dnsResolutionInterval as follows:
  Runtime interval =
   'dnsResolutionInterval' + (number of indexers in server settings - 1) * 30.
* The DNS resolution interval is extended by 30 seconds for each additional
  indexer in the server setting.
* Default: 300 seconds (5 minutes)

forceTimebasedAutoLB = <boolean>
* Forces existing data streams to switch to a newly elected indexer every
  auto load balancing cycle.
* On universal forwarders, use the 'EVENT_BREAKER_ENABLE' and
  'EVENT_BREAKER' settings in props.conf rather than 'forceTimebasedAutoLB'
  for improved load balancing, line breaking, and distribution of events.
* Default: false

#----Index Filter Settings.
```

```
# These settings are only applicable under the global [tcpout] stanza.
# This filter does not work if it is created under any other stanza.

forwardedindex.<n>.whitelist = <regular expression>
forwardedindex.<n>.blacklist = <regular expression>
* These filters determine which events get forwarded to the index,
  based on the indexes the events are targeted to.
* An ordered list of allow lists and deny lists, which together
  decide if events are forwarded to an index.
* The order is determined by <n>. <n> must start at 0 and continue with
  positive integers, in sequence. There cannot be any gaps in the sequence.
  * For example:
    forwardedindex.0.whitelist, forwardedindex.1.blacklist,
    forwardedindex.2.whitelist, ...
* The filters can start from either whitelist or blacklist. They are tested
  from forwardedindex.0 to forwardedindex.<max>.
* If both 'forwardedindex.<n>.whitelist' and 'forwardedindex.<n>.blacklist' are
  present for the same value of n, then 'forwardedindex.<n>.whitelist' is
  honored. 'forwardedindex.<n>.blacklist' is ignored in this case.
* In general, you do not need to change these filters from their default
  settings in $SPLUNK_HOME/system/default/outputs.conf.
* Filtered out events are not indexed if you do not enable local indexing.

forwardedindex.filter.disable = <boolean>
* Whether or not index filtering is active.
* A value of "true" means index filtering is disabled. Events for all indexes
  are then forwarded.
* Default: false

#----Automatic Load-Balancing
# Automatic load balancing is the only way to forward data.
# Round-robin method of load balancing is no longer supported.

autoLBFrequency = <integer>
* The amount of time, in seconds, that a forwarder sends data to an indexer
  before redirecting outputs to another indexer in the pool.
* Use this setting when you are using automatic load balancing of outputs
  from universal forwarders (UFs).
* Every 'autoLBFrequency' seconds, a new indexer is selected randomly from the
  list of indexers provided in the server setting of the target group
  stanza.
* Default: 30

autoLBFrequencyIntervalOnGroupFailure = <integer>
* When the entire target group is not reachable,
  'autoLBFrequencyIntervalOnGroupFailure' is the amount of time, in seconds,
  that a forwarder waits before attempting to connect to a target host in the
  group.
* While 'autoLBFrequencyIntervalOnGroupFailure' is in effect, 'autoLBFrequency'
  is ignored. Once first connection is established to a group, 'autoLBFrequency'
  comes into effect again.
* This setting is applied only when
  'autoLBFrequencyIntervalOnGroupFailure' is less than 'autoLBFrequency'.
* Every 'autoLBFrequencyIntervalOnGroupFailure' seconds, a new indexer is
  selected randomly from the list of indexers provided in the server setting
  of the target group stanza.
* -1 means this setting is not active.
* Default: -1

autoLBVolume = <integer>
* The volume of data, in bytes, to send to an indexer before a new indexer
  is randomly selected from the list of indexers provided in the server
```

```
  setting of the target group stanza.
* This setting is closely related to the 'autoLBFrequency' setting.
  The forwarder first uses 'autoLBVolume' to determine if it needs to
  switch to another indexer. If the 'autoLBVolume' is not reached,
  but the 'autoLBFrequency' is, the forwarder switches to another
  indexer as the forwarding target.
* A non-zero value means that volume-based forwarding is active.
* 0 means the volume-based forwarding is not active.
* Default: 0

maxSendQSize = <integer>
* The size of the tcpout client send buffer, in bytes.
  If tcpout client(indexer/receiver connection) send buffer is full,
  a new indexer is randomly selected from the list of indexers provided
  in the server setting of the target group stanza.
* This setting allows forwarder to switch to new indexer/receiver if current
  indexer/receiver is slow.
* A non-zero value means that max send buffer size is set.
* 0 means no limit on max send buffer size.
* Default: 0

autoBatch = <boolean>
* When set to 'true', the forwarder automatically sends chunks/events in batches
  to target receiving instance connection. The forwarder creates batches only
  if there are two or more chunks/events available in output connection queue.
* When set to 'false', the forwarder sends one chunk/event to target receiving
  instance connection. This is old legacy behavior.
* Default: true

#----Secure Sockets Layer (SSL) Settings----

# To set up SSL on the forwarder, set the following setting/value pairs.
# If you want to use SSL for authentication, add a stanza for each receiver
# that must be certified.

useSSL = <true|false|legacy>
* Whether or not the forwarder uses SSL to connect to the receiver, or relies
  on the 'clientCert' setting to be active for SSL connections.
* You do not need to set 'clientCert' if 'requireClientCert' is set to
  "false" on the receiver.
* A value of "true" means the forwarder uses SSL to connect to the receiver.
* A value of "false" means the forwarder does not use SSL to connect to the
  receiver.
* The special value "legacy" means the forwarder uses the 'clientCert' property to
  determine whether or not to use SSL to connect.
* Default: legacy

sslPassword = <password>
* The password associated with the Certificate Authority certificate (CAcert).
* The default Splunk CAcert uses the password "password".
* No default.

clientCert = <path>
* The full path to the client SSL certificate in Privacy Enhanced Mail (PEM)
  format.
* If you have not set 'useSSL', then this connection uses SSL if and only if
  you specify this setting with a valid client SSL certificate file.
* No default.

sslCertPath = <path>
* DEPRECATED.
* Use the 'clientCert' setting instead.
```

```
* The full path to the client SSL certificate.

cipherSuite = <string>
* The specified cipher string for the input processors.
* This setting ensures that the server does not accept connections using weak
  encryption protocols.
* The default can vary. See the 'cipherSuite' setting in
  $SPLUNK_HOME/etc/system/default/outputs.conf for the current default.

sslCipher = <string>
* The specified cipher string for the input processors.
* DEPRECATED.
* Use the 'cipherSuite' setting instead.

ecdhCurves = <comma-separated list>
* A list of Elliptic Curve-Diffie-Hellmann curves to use for ECDH
  key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of an SSL Client Hello.
* The server supports only the curves specified in the list.
* Splunk software only supports named curves that have been specified
  by their SHORT names.
* The list of valid named curves by their short and long names can be obtained
  by running this CLI command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Example setting: "ecdhCurves = prime256v1,secp384r1,secp521r1"
* The default can vary. See the 'ecdhCurves' setting in
  $SPLUNK_HOME/etc/system/default/outputs.conf for the current default.

sslRootCAPath = <path>
* The full path to the root Certificate Authority (CA) certificate store.
* DEPRECATED.
* Use the 'server.conf/[sslConfig]/sslRootCAPath' setting instead.
* Used only if 'sslRootCAPath' in server.conf is not set.
* The <path> must refer to a PEM format file containing one or more root
  CA certificates concatenated together.
* No default.

sslVerifyServerCert = <boolean>
* Serves as an additional step for authenticating your indexers.
* A value of "true" ensures that the server you are connecting to has a valid
  SSL certificate.
  * NOTE: Certificates with the same Common Name as the CA's certificate
    will fail this check.
* Both the common name and the alternate name of the server are then checked
  for a match.
* Default: false

tlsHostname = <string>
* A Transport Layer Security (TLS) extension that allows sending an identifier
  with SSL Client Hello.
* Default: empty string

sslVerifyServerName = <boolean>
* Whether or not splunkd, as a client, performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
```

```
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This terminates
  the connection between the client and the server. Splunkd logs this failure at
  the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

sslCommonNameToCheck = <comma-separated list>
* Checks the Common Name of the server's certificate against one or more of the
  names you specify for this setting.
* Separate multiple common names with commas.
* The Common Name identifies the host name associated with the certificate.
  For example, example www.example.com or example.com
* If there is no match, assume that Splunk software is not authenticated
  against this server.
* You must set the 'sslVerifyServerCert' setting to "true" for this setting
  to work.
* This setting is optional.
* Default: empty string (no common name checking).

sslAltNameToCheck = <comma-separated list>
* Checks the alternate name of the server's certificate against one or more of
  the names you specify for this setting.
* Separate multiple subject alternate names with commas.
* If there is no match, assume that Splunk software is not authenticated
  against this server.
* You must set the 'sslVerifyServerCert' setting to "true" for this setting to work.
* This setting is optional.
* Default: no alternate name checking

useClientSSLCompression = <boolean>
* Whether or not compression on TLS/SSL connections is enabled.
* Server-side compression in splunkd is on by default. Configuring this
  setting on the client side enables compression between both server and
  client.
* If server-side compression is off, this client-side setting has no effect.
* A value of "true" means compression on TLS/SSL is enabled.
* If you use this setting, the 'tcpout_connections' group in the metrics.log
  file shows throughput values before compression occurs.
* Default: true

sslQuietShutdown = <boolean>
* Enables quiet shutdown mode in SSL.
* Default: false

sslVersions = <comma-separated list>
* A comma-separated list of SSL versions to support.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2"
* The special version "*" selects all supported versions. The version "tls"
  selects all versions tls1.0 or newer
* If you prefix a version with "-", it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list, but
```

```
  does nothing.
* When configured in FIPS mode, "ssl3" is always disabled regardless
  of this configuration.
* The default can vary. See the 'sslVersions' setting in
  $SPLUNK_HOME/etc/system/default/outputs.conf for the current default.

#----Indexer Acknowledgment ----
# Indexer acknowledgment ensures that forwarded data is reliably delivered
# to the receiver.
#
# If the receiver is an indexer, it indicates that the indexer has received
# the data, indexed it, and written it to the file system. If the receiver
# is an intermediate forwarder, it indicates that the intermediate forwarder
# has successfully forwarded the data to the terminating indexer and has
# received acknowledgment from that indexer.
#
# Indexer acknowledgment is a complex feature that requires
# careful planning. Before using it, read the online topic describing it in
# the Splunk Enterprise Distributed Deployment manual.

useACK = <boolean>
* Whether or not to use indexer acknowledgment.
* Indexer acknowledgment is an optional capability on forwarders that helps
  prevent loss of data when sending data to an indexer.
* A value of "true" means the forwarder retains a copy of each sent event
  until the receiving system sends an acknowledgment.
  * The receiver sends an acknowledgment when it has fully handled the event
    (typically when it has written it to disk in indexing).
  * If the forwarder does not receive an acknowledgment, it resends the data
    to an alternative receiver.
  * NOTE: The maximum memory used for the outbound data queues increases
    significantly by default (500KB -> 28MB) when the 'useACK' setting is
    enabled. This is intended for correctness and performance.
* A value of "false" means the forwarder considers the data fully processed
  when it finishes writing it to the network socket.
* You can configure this setting at the [tcpout] or [tcpout:<target_group>]
  stanza levels. You cannot set it for individual servers at the
  [tcpout-server: ...] stanza level.
* Default: false
```

Syslog output----

```
# The syslog output processor is not available for universal or light
# forwarders.

# The following configuration is used to send output using syslog.

[syslog]

defaultGroup = <target_group>, <target_group>, ...

dropEventsOnQueueFull = <integer>[ms|s|m]
* See 'dropEventsOnQueueFull' in the "[tcpout]" stanza for
  information on this setting.

dropClonedEventsOnQueueFull = <integer>[ms|s|m]
* See 'dropClonedEventsOnQueueFull' in the "[tcpout]" stanza for
  information on this setting.

#######
```

```
# For the following settings, see the [syslog:<target_group>] stanza.

type = [tcp|udp]
priority = <<integer>> | NO_PRI
maxEventSize = <integer>

[syslog:<target_group>]

#----REQUIRED SETTINGS----
# The following settings are required for a syslog output group.

server = [<ip>|<servername>]:<port>
* The IP address or server name and port where the syslog server is running.
* Required.
* This setting specifies the port on which the syslog server listens.
* Default: 514

#----OPTIONAL SETTINGS----

# The following are optional settings for syslog output:

type = [tcp|udp]
* The network protocol to use.
* Default: udp

priority = <<integer>>|NO_PRI
* The priority value included at the beginning of each syslog message.
* The priority value ranges from 0 to 191 and is made up of a Facility
  value and a Level value.
* Enclose the priority value in "<>" delimeters. For example, specify a
  priority of 34 as follows: <34>
* The integer must be one to three digits in length.
* The value you enter appears in the syslog header.
* The value mimics the number passed by a syslog interface call. See the
  *nix man page for syslog for more information.
* Calculate the priority value as follows: Facility * 8 + Severity
  For example, if Facility is 4 (security/authorization messages)
  and Severity is 2 (critical conditions), the priority will be
  (4 * 8) + 2 = 34. Set the setting to <34>.
* If you do not want to add a priority value, set the priority to "<NO_PRI>".
* The table of facility and severity (and their values) is located in
  RFC3164. For example, http://www.ietf.org/rfc/rfc3164.txt section 4.1.1
* The table is reproduced briefly below. Some values are outdated.
  Facility:
     0 kernel messages
     1 user-level messages
     2 mail system
     3 system daemons
     4 security/authorization messages
     5 messages generated internally by syslogd
     6 line printer subsystem
     7 network news subsystem
     8 UUCP subsystem
     9 clock daemon
    10 security/authorization messages
    11 FTP daemon
    12 NTP subsystem
    13 log audit
    14 log alert
    15 clock daemon
    16 local use 0  (local0)
    17 local use 1  (local1)
```

```
    18 local use 2  (local2)
    19 local use 3  (local3)
    20 local use 4  (local4)
    21 local use 5  (local5)
    22 local use 6  (local6)
    23 local use 7  (local7)
  Severity:
    0  Emergency: system is unusable
    1  Alert: action must be taken immediately
    2  Critical: critical conditions
    3  Error: error conditions
    4  Warning: warning conditions
    5  Notice: normal but significant condition
    6  Informational: informational messages
    7  Debug: debug-level messages
* Default: <13> (Facility of "user" and Severity of "Notice")

syslogSourceType = <string>
* Specifies an additional rule for handling data, in addition to that
  provided by the 'syslog' source type.
* This string is used as a substring match against the sourcetype key. For
  example, if the string is set to "syslog", then all sourcetypes
  containing the string 'syslog' receive this special treatment.
* To match a sourcetype explicitly, use the pattern
  "sourcetype::sourcetype_name".
    * Example: syslogSourceType = sourcetype::apache_common
* Data that is "syslog" or matches this setting is assumed to already be in
  syslog format.
* Data that does not match the rules has a header, optionally a timestamp
  (if defined in 'timestampformat'), and a hostname added to the front of
  the event. This is how Splunk software causes arbitrary log data to match syslog expectations.
* No default.

timestampformat = <format>
* If specified, Splunk software prepends formatted timestamps to events
  forwarded to syslog.
* As above, this logic is only applied when the data is not syslog, or the
  type specified in the 'syslogSourceType' setting, because it is assumed
  to already be in syslog format.
* If the data is not in syslog-compliant format and you do not specify a
 'timestampformat', the output will not be RFC3164-compliant.
* The format is a strftime (string format time)-style timestamp formatting
  string. This is the same implementation used in the 'eval' search command,
  Splunk logging, and other places in splunkd.
  * For example: %b %e %H:%M:%S for RFC3164-compliant output
    * %b - Abbreviated month name (Jan, Feb, ...)
    * %e - Day of month
    * %H - Hour
    * %M - Minute
    * %s - Second
* For a more exhaustive list of the formatting specifiers, refer to the
  online documentation.
* Do not put the string in quotes.
* No default. No timestamp is added to the front of events.

maxEventSize = <integer>
* The maximum size of an event, in bytes, that Splunk software will transmit.
* All events exceeding this size are truncated.
* Optional.
* Default: 1024

#---- Routing Data to Syslog Server -----
```

```
# To route data to syslog servers:
# 1) Decide which events to route to which servers.
# 2) Edit the props.conf, transforms.conf, and outputs.conf files on the
#    forwarders.

# Edit $SPLUNK_HOME/etc/system/local/props.conf and set a TRANSFORMS-routing
# setting as shown below.
#
# [<spec>]
# TRANSFORMS-routing=<unique_stanza_name>

* <spec> can be:
  * <sourcetype>, the source type of an event
  * host::<host>, where <host> is the host for an event
  * source::<source>, where <source> is the source for an event

* Use the <unique_stanza_name> when creating your entry in transforms.conf.

# Edit $SPLUNK_HOME/etc/system/local/transforms.conf and set rules to match
# your props.conf stanza:
#
#  [<unique_stanza_name>]
#  REGEX = <your_regex>
#  DEST_KEY = _SYSLOG_ROUTING
#  FORMAT = <unique_group_name>

* Set <unique_stanza_name> to match the name you created in props.conf.
* Enter the regex rules in 'REGEX' to determine which events get
  conditionally routed.
* Set 'DEST_KEY' to "_SYSLOG_ROUTING" to send events via syslog.
* Set 'FORMAT' to match the syslog group name you create in outputs.conf.
```

IndexAndForward Processor-----

```
# The IndexAndForward processor determines the default behavior for indexing
# data on a Splunk instance. It has the "index" property, which determines
# whether indexing occurs.
#
# When the Splunk platform instance is not configured as a forwarder,
# 'index' is set to "true". That is, the Splunk platform instance indexes
# data by default.
#
# When the Splunk platform instance is configured as a forwarder, the
# processor sets 'index' to "false". That is, the Splunk platform instance
# does not index data by default.
#
# The IndexAndForward processor has no effect on the universal forwarder,
# which can never index data.
#
# If the [tcpout] stanza configures the indexAndForward setting, the value
# of that setting overrides the default value of 'index'. However, if you
# set 'index' in the [indexAndForward] stanza described below, it
# supersedes any value set in [tcpout].

[indexAndForward]

index = <boolean>
* Whether or not indexing is enabled on a Splunk platform instance.
* A value of "true" means the Splunk platform instance indexes data.
```

```
* A value of "false" means the Splunk platform instance does not index data.
* The default can vary. It depends on whether the Splunk
  platform instance is configured as a forwarder, and whether it is
  modified by any value configured for the 'indexAndForward'
  setting in the [tcpout] stanza.

selectiveIndexing = <boolean>
* Whether or not to index specific events that have the
  '_INDEX_AND_FORWARD_ROUTING' setting configured.
* A value of "true" means you can choose to index only specific events that have
  the '_INDEX_AND_FORWARD_ROUTING' setting configured.
* Configure the '_INDEX_AND_FORWARD_ROUTING' setting in inputs.conf as:
  [<input_stanza>]
  _INDEX_AND_FORWARD_ROUTING = local
* Default: false

[indexer_discovery:<name>]

pass4SymmKey = <string>
* The security key used to communicate between the cluster manager
  and the forwarders.
* This value must be the same for all forwarders and the manager node.
* You must explicitly set this value for each forwarder.
* If you specify a password here, you must also specify the same password
  on the manager node identified by the 'manager_uri' setting.

send_timeout = <decimal>
* The low-level timeout, in seconds, for sending messages to the manager node.
* Fractional seconds are allowed (for example, 60.95 seconds).
* Default: 30

rcv_timeout = <decimal>
* The low-level timeout, in seconds, for receiving messages from the manager node.
* Fractional seconds are allowed (for example, 60.95 seconds).
* Default: 30

cxn_timeout = <decimal>
* The low-level timeout, in seconds, for connecting to the manager node.
* Fractional seconds are allowed (for example, 60.95 seconds).
* Default: 30

manager_uri = <string>
* The URI and management port of the cluster manager used in indexer discovery.
* For example, https://SplunkManager01.example.com:8089

master_uri = <string>
* DEPRECATED. Use the 'manager_uri' setting instead.
```

Remote Queue Output

```
[remote_queue:<name>]

* This section explains possible settings for configuring a remote queue.
* Each remote_queue stanza represents an individually configured remote
  queue output.
* NOTE: Only ONE remote queue stanza is supported as an
  output queue.

remote_queue.* = <string>
```

```
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* This section explains possible settings for configuring a remote queue.
* With remote queues, the splunk indexer might require additional configuration,
  specific to the type of remote queue. You can pass configuration information
  to the splunk indexer by specifying the settings through the following schema:
  remote_queue.<scheme>.<config-variable> = <value>.
  For example:
  remote_queue.sqs.access_key = ACCESS_KEY
* This setting is optional.
* No default.

remote_queue.type = sqs|kinesis|sqs_smartbus|sqs_smartbus_cp
* Currently not supported. This setting is related to a feature that is
  still under development.
* Required.
* Specifies the remote queue type, which can be "SQS", "Kinesis", "SQS Smartbus", or "SQS Smartbus CP".
* If the type is "sqs_smartbus_cp", the [cloud_processing_queue] stanza must be present.

compressed = <boolean>
* See the description for TCPOUT SETTINGS in outputs.conf.spec.

negotiateProtocolLevel = <unsigned integer>
* See the description for TCPOUT SETTINGS in outputs.conf.spec.

channelReapInterval = <integer>
* See the description for TCPOUT SETTINGS in outputs.conf.spec.

channelTTL = <integer>
* See the description for TCPOUT SETTINGS in outputs.conf.spec.

channelReapLowater = <integer>
* See the description for TCPOUT SETTINGS in outputs.conf.spec.

concurrentChannelLimit = <unsigned integer>
* See the description for [splunktcp] in inputs.conf.spec.
```

Simple Queue Service (SQS) specific settings

```
remote_queue.sqs.access_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The access key to use when authenticating with the remote queue
  system that supports the SQS API.
* If not specified, the forwarder looks for the environment variables
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the forwarder
  attempts to use the secret key from the IAM (Identity and Access
  Management) role.
* Optional.
* Default: not set

remote_queue.sqs.secret_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the secret key to use when authenticating with the remote queue
  system supporting the SQS API.
* If not specified, the forwarder looks for the environment variables
```

```
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the forwarder
  attempts to use the secret key from the IAM (Identity and Access
  Management) role.
* Optional.
* Default: not set

remote_queue.sqs.auth_region = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The authentication region to use when signing the requests while interacting
  with the remote queue system supporting the Simple Queue Service (SQS) API.
* If not specified and the forwarder is running on EC2, the auth_region is
  constructed automatically based on the EC2 region of the instance where the
  the forwarder is running.
* Optional.
* Default: not set

remote_queue.sqs.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote queue system supporting the Simple Queue Service (SQS) API.
* Use the scheme, either http or https, to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://sqs.<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via the 'remote_queue.sqs.auth_region' setting
  or a value constructed automatically based on the EC2 region of the
  running instance.
* Example: https://sqs.us-west-2.amazonaws.com/
* Optional.

remote_queue.sqs.message_group_id = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the Message Group ID for Amazon Web Services Simple Queue Service
  (SQS) First-In, First-Out (FIFO) queues.
* Setting a Message Group ID controls how messages within an AWS SQS queue are
  processed.
* For information on SQS FIFO queues and how messages in those queues are
  processed, see "Recommendations for FIFO queues" in the AWS SQS Developer
  Guide.
* If you configure this setting, Splunk software assumes that the SQS queue is
  a FIFO queue, and that messages in the queue should be processed first-in,
  first-out.
* Otherwise, Splunk software assumes that the SQS queue is a standard queue.
* Can be between 1-128 alphanumeric or punctuation characters.
* NOTE: FIFO queues must have Content-Based De-duplication enabled.
* Optional.
* Default: not set

remote_queue.sqs.retry_policy = max_count|none
* Sets the retry policy to use for remote queue operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a queue operation is
    retried upon intermittent failure. Set max_count with the
    'max_count.max_retries_per_part' setting.
  + "none": Do not retry file operations upon failure.
* Optional.
```

```
* Default: max_count

remote_queue.sqs.max_count.max_retries_per_part = <unsigned integer>
* When the 'remote_queue.sqs.retry_policy' setting is "max_count", sets the
  maximum number of times a queue operation will be retried upon intermittent
  failure.
* Optional.
* Default: 9

remote_queue.sqs.timeout.connect = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* Sets the connection timeout, in milliseconds, to use when interacting with
  the SQS for this queue.
* Default: 5000

remote_queue.sqs.timeout.read = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Sets the read timeout, in milliseconds, to use when interacting with the
  SQS for this queue.
* Optional.
* Default: 60000

remote_queue.sqs.timeout.write = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Sets the write timeout, in milliseconds, to use when interacting with
  the SQS for this queue.
* Optional.
* Default: 60000

remote_queue.sqs.large_message_store.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote storage system supporting the S3 API.
* Use the scheme, either http or https, to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://s3-<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via 'remote_queue.sqs.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://s3-us-west-2.amazonaws.com/
* Optional.
* Default: not set

remote_queue.sqs.large_message_store.path = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The remote storage location where messages larger than the underlying
  queue's maximum message size will reside.
* The format for this value is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string for
    identifying a location inside the storage system.
* The following external systems are supported:
  * Object stores that support AWS's S3 protocol. These stores use the scheme
    "s3". For example, "path=s3://mybucket/some/path".
* If not specified, the queue drops messages exceeding the underlying queue's
  maximum message size.
```

```
* Optional.
* Default: not set

remote_queue.sqs.send_interval = <number><unit>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The interval that the remote queue output processor waits for data to
  arrive before sending a partial batch to the remote queue.
* Examples: 30s, 1m
* Optional.
* Default: 30s

remote_queue.sqs.max_queue_message_size = <integer>[KB|MB|GB]
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum message size to which events are batched for upload to
  the remote queue.
* Specify this value as an integer followed by KB, MB, or GB (for example,
  10MB is 10 megabytes)
* Queue messages are sent to the remote queue when the next event processed
  would otherwise result in a message exceeding the maximum message size.
* The maximum value for this setting is 5GB.
* Optional.
* Default: 10MB

remote_queue.sqs.enable_data_integrity_checks = <boolean>
* If "true", Splunk software sets the data checksum in the metadata field of
  the HTTP header during upload operation to S3.
* The checksum is used to verify the integrity of the data on uploads.
* Default: false

remote_queue.sqs.enable_signed_payloads  = <boolean>
* If "true", Splunk software signs the payload during upload operation to S3.
* This setting is valid only for remote.s3.signature_version = v4
* Default: true
```

Kinesis specific settings

```
remote_queue.kinesis.access_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the access key to use when authenticating with the remote queue
  system supporting the Kinesis API.
* If not specified, the forwarder looks for the environment variables
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the forwarder
  attempts to use the secret key from the IAM role.
* Optional.
* Default: not set

remote_queue.kinesis.secret_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the secret key to use when authenticating with the remote queue
  system supporting the Kinesis API.
* If not specified, the forwarder looks for the environment variables
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the forwarder
  attempts to use the secret key from the IAM role.
```

```
* Optional.
* Default: not set

remote_queue.kinesis.auth_region = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The authentication region to use when signing the requests when interacting
  with the remote queue system supporting the Kinesis API.
* If not specified and the forwarder is running on EC2, the auth_region is
  constructed automatically based on the EC2 region of the instance where the
  the forwarder is running.
* Optional.
* Default: not set

remote_queue.kinesis.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote queue system supporting the Kinesis API.
* Use the scheme, either http or https, to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://kinesis.<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via the 'remote_queue.kinesis.auth_region' setting
  or a value constructed automatically based on the EC2 region of the running instance.
* Optional.
* Example: https://kinesis.us-west-2.amazonaws.com/

remote_queue.kinesis.enable_data_integrity_checks = <boolean>
* If "true", Splunk software sets the data checksum in the metadata field
  of the HTTP header during upload operation to S3.
* The checksum is used to verify the integrity of the data on uploads.
* Default: false

remote_queue.kinesis.enable_signed_payloads  = <boolean>
* If "true", Splunk software signs the payload during upload operation to S3.
* This setting is valid only for remote.s3.signature_version = v4
* Default: true

remote_queue.kinesis.retry_policy = max_count|none
* Sets the retry policy to use for remote queue operations.
* Optional.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a queue operation is
    retried upon intermittent failure. Specify the max_count with the
    'max_count.max_retries_per_part' setting.
  + "none": Do not retry file operations upon failure.
* Default: max_count

remote_queue.kinesis.max_count.max_retries_per_part = <unsigned integer>
* When the 'remote_queue.kinesis.retry_policy' setting is max_count,
  sets the maximum number of times a queue operation is retried
  upon intermittent failure.
* Optional.
* Default: 9

remote_queue.kinesis.timeout.connect = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
```

```
* Sets the connection timeout, in milliseconds, to use when interacting with
  Kinesis for this queue.
* Default: 5000

remote_queue.kinesis.timeout.read = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* Sets the read timeout, in milliseconds, to use when interacting with Kinesis
  for this queue.
* Default: 60000

remote_queue.kinesis.timeout.write = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Sets the write timeout, in milliseconds, to use when interacting with
  Kinesis for this queue.
* Optional.
* Default: 60000

remote_queue.kinesis.large_message_store.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote storage system supporting the S3 API.
* Use the scheme, either http or https, to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://s3-<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via 'remote_queue.kinesis.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://s3-us-west-2.amazonaws.com/
* Optional.
* Default: not set

remote_queue.kinesis.large_message_store.path = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The remote storage location where messages larger than the underlying
  queue's maximum message size will reside.
* The format for this setting is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string for
    identifying a location inside the storage system.
* The following external systems are supported:
  * Object stores that support AWS's S3 protocol. These stores use the
    scheme "s3".
    For example, "path=s3://mybucket/some/path".
* If not specified, the queue drops messages exceeding the underlying queue's
  maximum message size.
* Optional.
* Default: not set

remote_queue.kinesis.send_interval = <number><unit>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The interval that the remote queue output processor waits for data to
  arrive before sending a partial batch to the remote queue.
* For example, 30s, 1m
* Optional.
* Default: 30s
```

```
remote_queue.kinesis.max_queue_message_size = <integer>[KB|MB|GB]
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum message size to which events are batched for upload to the remote
  queue.
* Specify this value as an integer followed by KB or MB (for example, 500KB
  is 500 kilobytes).
* Queue messages are sent to the remote queue when the next event processed
  would otherwise result in the message exceeding the maximum message size.
* The maximum value for this setting is 5GB.
* Optional.
* Default: 10MB

remote_queue.kinesis.tenantId = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The ID of the tenant that owns the messages being
  written to the remote queue.
* If not specified, the messages do not belong to any tenant.
* Optional.
* Default: not set
```

Simple Queue Service Smartbus (SQS Smartbus) or Simple Queue Service Smartbus CP (SQS Smartbus CP) specific settings

```
# The settings for SQS Smartbus (sqs_smartbus) and SQS Smartbus CP (sqs_smartbus_cp)
# are identical in the remote queue output. The following section uses "sqs_smartbus"
  as an example.

remote_queue.sqs_smartbus.encoding_format = protobuf|s2s
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the encoding format used to write data to the
  remote queue.
* Default: protobuf

remote_queue.sqs_smartbus.access_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The access key to use when authenticating with the remote queue
  system that supports the SQS API.
* If not specified, the splunk instance looks for the environment variables
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the splunk instance
  attempts to use the secret key from the IAM (Identity and Access
  Management) role.
* Optional.
* Default: not set

remote_queue.sqs_smartbus.secret_key = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the secret key to use when authenticating with the remote queue
  system supporting the SQS API.
* If not specified, the splunk instance looks for the environment variables
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order). If the environment
  variables are not set and the forwarder is running on EC2, the splunk instance
  attempts to use the secret key from the IAM (Identity and Access
```

```
  Management) role.
* Optional.
* Default: not set

remote_queue.sqs_smartbus.auth_region = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* The authentication region to use when signing the requests while interacting
  with the remote queue system supporting the Simple Queue Service (SQS) API.
* If not specified and the splunk instance is running on EC2, the auth_region is
  constructed automatically based on the EC2 region of the instance where the
  the splunk instance is running.
* Default: not set

remote_queue.sqs_smartbus.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote queue system supporting the Simple Queue Service (SQS) API.
* Use the scheme, either http or https, to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://sqs.<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via the 'remote_queue.sqs.auth_region' setting
  or a value constructed automatically based on the EC2 region of the
  running instance.
* Example: https://sqs.us-west-2.amazonaws.com/
* Optional.

remote_queue.sqs_smartbus.message_group_id = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the Message Group ID for Amazon Web Services Simple Queue Service
  (SQS) First-In, First-Out (FIFO) queues.
* Setting a Message Group ID controls how messages within an AWS SQS queue are
  processed.
* For information on SQS FIFO queues and how messages in those queues are
  processed, see "Recommendations for FIFO queues" in the AWS SQS Developer
  Guide.
* If you configure this setting, Splunk software assumes that the SQS queue is
  a FIFO queue, and that messages in the queue should be processed first-in,
  first-out.
* Otherwise, Splunk software assumes that the SQS queue is a standard queue.
* Can be between 1-128 alphanumeric or punctuation characters.
* NOTE: FIFO queues must have Content-Based De-duplication enabled.
* Optional.
* Default: not set

remote_queue.sqs_smartbus.retry_policy = max_count|none
* Sets the retry policy to use for remote queue operations.
* Optional.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a queue operation is
    retried upon intermittent failure. Set max_count with the
    'max_count.max_retries_per_part' setting.
  + "none": Do not retry file operations upon failure.
* Default: max_count

remote_queue.sqs_smartbus.max_count.max_retries_per_part = <unsigned integer>
```

```
* When the 'remote_queue.sqs_smartbus.retry_policy' setting is "max_count", sets the
  maximum number of times a queue operation will be retried upon intermittent
  failure.
* Optional.
* Default: 3

remote_queue.sqs_smartbus.timeout.connect = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Sets the connection timeout, in milliseconds, to use when interacting with
  the SQS for this queue.
* Optional.
* Default: 5000

remote_queue.sqs_smartbus.timeout.read = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Sets the read timeout, in milliseconds, to use when interacting with the
  SQS for this queue.
* Optional.
* Default: 60000

remote_queue.sqs_smartbus.timeout.write = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Sets the write timeout, in milliseconds, to use when interacting with
  the SQS for this queue.
* Optional.
* Default: 60000

remote_queue.sqs_smartbus.large_message_store.endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The URL of the remote storage system supporting the S3 API.
* Use the scheme, either http or https, to enable or disable SSL connectivity
  with the endpoint.
* If not specified, the endpoint is constructed automatically based on the
  auth_region as follows: https://s3-<auth_region>.amazonaws.com
* If specified, the endpoint must match the effective auth_region, which is
  either a value specified via 'remote_queue.sqs_smartbus.auth_region' or a value
  constructed automatically based on the EC2 region of the running instance.
* Example: https://s3-us-west-2.amazonaws.com/
* Optional.
* Default: not set

remote_queue.sqs_smartbus.large_message_store.path = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The remote storage location where messages larger than the underlying
  queue's maximum message size will reside.
* The format for this value is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string for
    identifying a location inside the storage system.
* The following external systems are supported:
  * Object stores that support AWS's S3 protocol. These stores use the scheme
    "s3". For example, "path=s3://mybucket/some/path".
* If not specified, the queue drops messages exceeding the underlying queue's
  maximum message size.
* Optional.
* Default: not set
```

```
remote_queue.sqs_smartbus.large_message_store.sslVerifyServerCert = <boolean>
* If set to true, the Splunk platform verifies the certificate presented by the S3
  server and checks that the common name and alternative name match the ones
  specified in 'remote_queue.sqs_smartbus.large_message_store.sslCommonNameToCheck' and
  'remote_queue.sqs_smartbus.large_message_store.sslAltNameToCheck'.
* Default: false

remote_queue.sqs_smartbus.large_message_store.sslVersions = <comma-separated list>
* Comma-separated list of SSL versions to connect to
  'remote.sqs_smartbus.large_message_store.endpoint'.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list
  but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Default: tls1.2

remote_queue.sqs_smartbus.large_message_store.sslCommonNameToCheck =
  <comma-separated list>
* If this value is set, and
  'remote_queue.sqs_smartbus.large_message_store.sslVerifyServerCert' is set to true,
  the Splunk platform instance checks the common name of the certificate presented by
  the remote server (specified in
  'remote_queue.sqs_smartbus.large_message_store.endpoint') against this list
  of common names.
* Default: not set

remote_queue.sqs_smartbus.large_message_store.sslAltNameToCheck = <comma-separated list>
* If this value is set, and
  'remote_queue.sqs_smartbus.large_message_store.sslVerifyServerCert' is set to true,
  the Splunk platform instance checks the alternate name(s) of the certificate
  presented by the remote server (specified in
  'remote_queue.sqs_smartbus.large_message_store.endpoint') against this list of
  subject alternate names.
* Default: not set

remote_queue.sqs_smartbus.large_message_store.sslRootCAPath = <path>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together. S3 certificate
  will be validated against the CAs present in this file.
* Default: The value of [sslConfig]/'caCertFile' in server.conf

remote_queue.sqs_smartbus.large_message_store.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Default: TLSv1+HIGH:TLSv1.2+HIGH:@STRENGTH

remote_queue.sqs_smartbus.large_message_store.ecdhCurves = <comma-separated list>
* ECDH curves to use for ECDH key negotiation.
* Specify the curves in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only supports named curves specified
  by their short names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: not set
```

```
remote_queue.sqs_smartbus.large_message_store.dhFile = <string>
* PEM format Diffie-Hellman parameter file name.
* DH group size must be no less than 2048bits.
* This file is required in order to enable any Diffie-Hellman ciphers.
* Optional.
* Default: not set

remote_queue.sqs_smartbus.send_interval = <number><unit>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* The interval that the remote queue output processor waits for data to
  arrive before sending a partial batch to the remote queue.
* Examples: 100ms, 5s
* Default: 4s

remote_queue.sqs_smartbus.consume_interval = <number><unit>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* The interval that the remote output worker consumes from data queue.
* Examples: 50ms, 1s
* Default: 100ms

remote_queue.sqs_smartbus.max_queue_message_size = <integer>[KB|MB|GB]
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum message size for batched events for upload to the remote queue.
* Queue messages contain a series of one or more events. When an event causes the message
  size to exceed this setting, the message is sent to the remote queue.
* Specify this value as an integer followed by KB, MB, or GB (for example,
  10MB is 10 megabytes)
* Default: 10MB

remote_queue.sqs_smartbus.enable_data_integrity_checks = <boolean>
* If "true", Splunk software sets the data checksum in the metadata field of
  the HTTP header during upload operation to S3.
* The checksum is used to verify the integrity of the data on uploads.
* Default: false

remote_queue.sqs_smartbus.enable_signed_payloads  = <boolean>
* If "true", Splunk software signs the payload during upload operation to S3.
* This setting is valid only for remote.s3.signature_version = v4
* Default: true

remote_queue.sqs_smartbus.drop_data = <boolean>
* Currently not supported. This setting is related to a feature that is still
  under development.
* Determines whether Splunk software drops the data from all Splunk managed internal
  indexes and indexes listed in 'remote_queue.sqs_smartbus.drop_data_index_list'
* A value of "true" means that Splunk software drops the data.
* Default: false

remote_queue.sqs_smartbus.drop_data_index_list = <comma-separated list>
* Currently not supported. This setting is related to a feature that is still
  under development.
* A comma-separated list of indexes for which you want to drop the data.
  For example: index_1, index_2, index_3.
* If 'remote_queue.sqs_smartbus.drop_data' is set to "true" then Splunk software
  drops the data from 'drop_data_index_list'.
* Default: not set
```

```
remote_queue.sqs_smartbus.executor_max_workers_count = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum number of worker threads available per pipeline set to execute SQS output
  worker tasks.
* A value of 0 is equivalent to 1.
* The maximum value for this setting is 20.
* Default: 10

remote_queue.sqs_smartbus.executor_max_jobs_count = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum number of jobs that each worker thread per pipeline set can queue.
* A value of 0 is equivalent to 1.
* The maximum value for this setting is 100.
* Default: 50

remote_queue.sqs_smartbus.large_message_store.encryption_scheme = sse-s3 | sse-c | none
* Currently not supported. This setting is related to a feature that is
  still under development.
* The encryption scheme used by remote storage.
* Default: none.

remote_queue.sqs_smartbus.large_message_store.kms_endpoint = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The endpoint to connect to for generating KMS keys.
* This setting is required if 'large_message_store.encryption_scheme' is
  set to sse-c.
* Examples: https://kms.us-east-2.amazonaws.com
* No default.

remote_queue.sqs_smartbus.large_message_store.key_id = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The ID for the primary key that KMS uses to generate a data key pair.
  The primary key is stored in AWS.
* This setting is required if 'large_message_store.encryption_scheme' is
  set to sse-c.
* Examples: alias/sqsssekeytrial, 23456789-abcd-1234-11aa-c50f99011223
* No default.

remote_queue.sqs_smartbus.large_message_store.key_refresh_interval = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The time interval to refresh primary key.
* Default: 24h

remote_queue.sqs_smartbus.enable_inline_data = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies whether to bypass S3 and use SQS directly to send events.
* A value of "true" means that if the data packet is small enough to fit in
  SQS (256KB max), the Splunk Cloud Platform will use SQS to send event data.
* This setting only applies when remote_queue.sqs_smartbus.encoding_format=protobuf
* Default: false

remote_queue.sqs_smartbus.check_replication_enabled = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies whether to enable cross-region replication status checks of
```

```
  uploaded ingest blobs on remote storage.
* Default: false

remote_queue.sqs_smartbus.check_replication_interval = <number><unit>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* The interval that the remote queue output processor waits before checking
  the replication status of an uploaded ingest blob on remote storage.
* Examples: 100ms, 5s
* Default: 60s

remote_queue.sqs_smartbus.check_replication_executor_max_workers_count = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum number of worker threads available per pipeline set to execute SQS output
  replication related tasks such as replication status checks.
* A value of 0 is equivalent to 5.
* Default: 5

remote_queue.sqs_smartbus.check_replication_executor_max_jobs_count = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The maximum number of jobs that each replication executor worker thread per pipeline set
  can queue.
* A value of 0 is equivalent to 1000.
* Default: 1000

remote_queue.sqs_smartbus.enable_shared_receipts = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* If "true", receipts will be shared among ingest blobs.
* Default: false
```

Remote File System (RFS) Output

```
[rfs]
* Global settings that individual rfs output destinations can inherit.

partitionBy = legacy | (year|month|day) [, sourcetype]
* Specifies schema to partition and store events into seperate files on the
  rfsoutput destination(s). It affects the file storage location specified by
  the "path" for any given destination in the manner described below.
  * legacy - no partitioning and the events are batched together on the order of
    arrival. This appends path segments that encode the latest timestamp among
    all events in the batch similar to the strftime format "%Y/%m/%d".
  * year|month|day - span of event timestamp to use as the primary partition key.
    This encodes primary field as multiple path segments and appends to the path
    in the decreasing order of significance. For example, "day" produces
    "year=%Y/month=%m/day=%d", "month" produces "year=%Y/month=%m" and "year"
    produces "year=%Y".
  * sourcetype - optional secondary partition key applied over primary partition key.
    This appends a single path segment encoded in the form "sourcetype=<sourcetype>".
* Examples:
    Illustrated below is the set of possible settings and how they affect the
    file storage path, for events generated on August 15, 2022.
    * partitionBy = day, sourcetype
       Results file path into "<path>/year=2022/month=08/day=15/sourcetype=<srctype>/"
    * partitionBy = day
       Results file path into "<path>/year=2022/month=08/day=15/"
```

```
    * partitionBy = month
       Results file path into "<path>/year=2022/month=08/"
    * partitionBy = year
       Results file path into "<path>/year=2022/"
    * partitionBy = legacy
       Results file path into "<path>/2022/08/15/"
* Default: legacy

dropEventsOnUploadError = <boolean>
* Whether or not the ingest actions feature drops events if it encounters an
  error when uploading events to output destination.
* A value of "true" means that, if there is an error writing to the destination, the
  error will be logged, and the events in that batch dropped. Ingest will not be
  blocked, but data might be lost.
* A value of "false" means, if there is an error writing to the destination, the
  error will be logged, and events will NOT be dropped. splunkd will continually
  attempt to write the batch. Because events are not dropped, this might cause
  queues to become blocked, and data ingestion to stop.
* This setting is optional.
* Default: false

batchSizeThresholdKB = <integer>
* The size, in kilobytes, of the uncompressed events in the RfsOutputProcessor send buffer.
  RfsOutputProcessor batches events before writing them into files on the destination.
  If the current buffer size is greater than 'batchSizeThresholdKB' kilobytes, then
  the data will be written to the destination immediately.
* This threshold may not be honored if the total memory usage for raw events exceeds
  limits.conf/[ingest_actions]/rfs.provider.rawdata_limit_mb for a storage provider.
* If you increase this setting, you may also want to increase the value of
  server.conf/[queue:rfsQueue]/maxSize.
* Default: 131072 (128 MiB)
* Max threshold value: 5242880 (5 GiB)

batchTimeout = <integer>
* RfsOutputProcessor batches events before flushing to the destination.
* If a batch has not hit any other criteria for being flushed, and
  the batch is at least this many seconds old, flush the batch.
* This threshold may not be honored if the total memory usage for raw events exceeds
  limits.conf/[ingest_actions]/rfs.provider.rawdata_limit_mb for a storage provider.
* Default = 30

compression = none|gzip|lz4|zstd
* Sets the algorithm to use for compressing files before writing to the destination.
* The RfsOutputProcessor writes files with the appropriate extension for the compression
  algorithm, for example, .zst for zstd, .gz for gzip and .lz4 for lz4.
* Default: zstd

compressionLevel = <integer>
* Sets compression level for the specified compression algorithm,
  when RfsOutputProcessor writes files. Must be between 0 and 10.
* Default: 3

format = json|ndjson|raw
* Specifies output format when RfsOutputProcessor writes events into files
  on the destination.
* json: The file will include a JSON array. Each event will be element of
  the JSON array.
* ndjson: The file will include multiple JSON objects separated by a newline
  character. Each event is corresponding to one JSON object.
* raw: The file includes multiple raw events separated by a newline character.
* Default: json
```

```
format.json.index_time_fields = <boolean>
* Specifies whether to include index-time fields when RfsOutputProcessor
  writes events to the destination in HEC JSON format.
* Default: true

format.ndjson.index_time_fields = <boolean>
* Specifies whether to include index-time fields when RfsOutputProcessor
  writes events to the destination in new line delimited JSON format.
* Default: true

[rfs:<name>]

* This section explains the configuration settings for the ingest actions feature
  to send data to an external storage interface, such as AWS S3 or a file-system mount.
* Each [rfs:<name>] stanza represents an individually configured location.
* The "name" is a unique identifier for the storage destination, and is shown
  as a routing destination when using the ingest actions UI.
  For example: [rfs:syslog_filtered_events], [rfs:threat_detection_logs] etc.

path = <string>
* Required.
* This setting points to the storage location where files would be stored.
* The format for specifying storage location is:
  <scheme>://<storage-location-specifier>
    * The "scheme" identifies the storage interface used.
    * Currently "s3" and "file" are the only supported schemes.
    * The "storage-location-specifier" is a system-specific string for
      identifying a location inside the storage system.
    * For AWS S3, this is specified as "path=s3://mybucket/some/path"
    * For filesystem interface, this is specified as "file://my/local/path"

description = <string>
* Optional.
* A general description to explain the configuration settings for the ingest actions
  feature to send data to a destination.
* No default.

remote.* = <string>
* Optional.
* This section explains possible settings for configuring a remote output.
* With remote outputs, the splunk indexer might require additional configuration,
  specific to the type of remote storage. You can pass configuration information
  to the splunk indexer by specifying the settings through the following schema:
  remote_queue.<scheme>.<config-variable> = <value>.
  For example:
  remote.s3.access_key = ACCESS_KEY
  Refer to "Volume settings" in indexes.conf for all settings.
* This setting is optional.
* No default.

remote.s3.access_key = <string>
* Specifies the access key to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, the indexer will look for these environment variables:
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the access key from the IAM role.
* Optional.
* No default.

remote.s3.secret_key = <string>
* Specifies the secret key to use when authenticating with the remote storage
```

```
  system supporting the S3 API.
* If not specified, the indexer will look for these environment variables:
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the secret key from the IAM role.
* Optional.
* No default.

remote.s3.signature_version = v2|v4
* The signature version to use when authenticating with the remote storage
  system supporting the S3 API.
* For 'sse-kms' and 'sse-c' server-side encryption schemes, and for 'cse'
  client-side encryption scheme, you must use signature_version=v4.
* For signature_version=v2 you must set url_version=v1.
* Optional.
* Default: v4

remote.s3.url_version = v1|v2
* Specifies which url version to use, both for parsing the endpoint/path, and
* for communicating with the remote storage. This value only needs to be
* specified when running on non-AWS S3-compatible storage that has been configured
* to use v2 urls.
* In v1 the bucket is the first element of the path.
* Example: mydomain.com/bucketname/rest/of/path
* In v2 the bucket is the outermost subdomain in the endpoint.
* Exmaple: bucketname.mydomain.com/rest/of/path
* Default: v1

remote.s3.supports_versioning = <boolean>
* Specifies whether the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage.
* This setting determines how splunkd removes data from remote storage.
  If set to true, splunkd will delete all versions of objects at
  time of data removal. Otherwise, if set to false, splunkd will use a simple DELETE
  (See https://docs.aws.amazon.com/AmazonS3/latest/dev/DeletingObjectVersions.html).
* Optional.
* Default: true

remote.s3.endpoint = <URL>
* The URL of the remote storage system supporting the S3 API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified and the indexer is running on EC2, the endpoint will be
  constructed automatically based on the EC2 region of the instance where the
  indexer is running, as follows: https://<bucketname>.s3-<region>.amazonaws.com
* Example: https://<bucketname>.s3-us-west-2.amazonaws.com
* Optional.

remote.s3.encryption = sse-s3 | sse-kms | none
* The encryption scheme to use for output to remote storage for data stored (data at rest).
* sse-s3: Search for "Protecting Data Using Server-Side Encryption with AWS S3-Managed
          Encryption Keys" on the Amazon Web Services documentation site.
* sse-kms: Search for "Protecting Data Using Server-Side Encryption with CMKs Stored in AWS
           Key Management Service (SSE-KMS)" on the Amazon Web Services documentation site.
* Note: sse-c is not supported for RfsOutputProcessor
* Optional.
* No default.

remote.s3.auth_region = <string>
* The authentication region to use for signing requests when interacting
  with the remote storage system supporting the S3 API.
```

```
* Used with v4 signatures only.
* If unset and the endpoint (either automatically constructed or explicitly
  set with remote.s3.endpoint setting) uses an AWS URL (for example,
  https://<bucketname>.s3-us-west-1.amazonaws.com), the instance attempts to extract
  the value from the endpoint URL (for example, "us-west-1").  See the
  description for the remote.s3.endpoint setting.
* If unset and an authentication region cannot be determined, the request
  will be signed with an empty region value. This can lead to rejected
  requests when using non-AWS S3-compatible storage.
* Optional.
* No default.

remote.s3.retry_policy = max_count
* Sets the retry policy to use for remote file operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a file operation will be
    retried upon intermittent failure both for individual parts of a multipart
    download or upload and for files as a whole.
* Optional.
* Default: max_count

remote.s3.sslVerifyServerCert = <boolean>
* If this is set to true, Splunk verifies certificate presented by S3
  server and checks that the common name/alternate name matches the ones
  specified in 'remote.s3.sslCommonNameToCheck' and
  'remote.s3.sslAltNameToCheck'.
* Optional
* Default: false

remote.s3.sslVersions = <versions_list>
* Comma-separated list of SSL versions to connect to 'remote.s3.endpoint'.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list
  but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Optional.
* Default: tls1.2

remote.s3.sslCommonNameToCheck = <commonName1>, <commonName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to true,
  splunkd checks the common name of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this list
  of common names.
* Default: not set

remote.s3.sslAltNameToCheck = <alternateName1>, <alternateName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to true,
  splunkd checks the alternate name(s) of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this list of
  subject alternate names.
* No default.

remote.s3.sslRootCAPath = <path>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together. S3 certificate
  will be validated against the CAs present in this file.
```

```
* Optional.
* Default: The value of '[sslConfig]/caCertFile' in server.conf

remote.s3.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Optional.
* Default: TLSv1+HIGH:TLSv1.2+HIGH:@STRENGTH

remote.s3.ecdhCurves = <comma-separated list>
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only supports named curves specified
  by their SHORT names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Optional.
* No default.

remote.s3.kms.auth_region = <string>
* Required if 'remote.s3.auth_region' is unset and Splunk can not
  automatically extract this information.
* Similar to 'remote.s3.auth_region'.
* If not specified, KMS access uses 'remote.s3.auth_region'.
* No default.

remote.s3.kms.key_id = <string>
* Required if remote.s3.encryption = sse-kms
* Specifies the identifier for Customer Master Key (CMK) on KMS. It can be the
  unique key ID or the Amazon Resource Name (ARN) of the CMK or the alias
  name or ARN of an alias that refers to the CMK.
* Examples:
  Unique key ID: 1234abcd-12ab-34cd-56ef-1234567890ab
  CMK ARN: arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
  Alias name: alias/ExampleAlias
  Alias ARN: arn:aws:kms:us-east-2:111122223333:alias/ExampleAlias
* No default.

remote.s3.kms.<ssl_settings> = <...>
* Optional.
* See the descriptions of the SSL settings for remote.s3.<ssl_settings>
  above. e.g. remote.s3.sslVerifyServerCert.
* Valid ssl_settings are sslVerifyServerCert, sslVersions, sslRootCAPath,
  sslAltNameToCheck, sslCommonNameToCheck, cipherSuite, ecdhCurves, and dhFile.
* All of these settings are optional.
* All of these settings have the same defaults as
  'remote.s3.<ssl_settings>'.

remote.s3.metadata_max_attempts = <integer>
* Imposes a maximum number of times an operation will be
  retried upon failing to retrieve credentials from EC2 metadata service endpoint.
* This value must be between 1 and 10.
* Default: 10

remote.sts.assume_role.role_arn = <string>
* This feature is supported on Splunk Cloud only.
* The Amazon Resource Name (ARN) of the role to assume.
* Normally, splunkd will use whatever credentials are available (i.e. access_key/secret_key, instance
```

```
  IAM roles, etc) to directly access AWS services, such as S3 and KMS. If this is set, instead of
  using those credentials directly, splunkd will contact the STS AssumeRole API to get credentials
  associated with the role here, and use that "assumed" role to access other services.
* Make sure only to specify this when need temporary security credentials
  to access AWS resources that you might not normally have access to.
* Example:
  arn:aws:iam::111122223333:role/SplunkIngestActions
* Only applicable when the rfs destination is an aws s3 destination (path starts with 's3://').
* No default

remote.sts.assume_role.external_id = <string>
* This feature is supported on Splunk Cloud only.
* A unique identifier that might be required when you assume a role in another account.
* If the account to which the role belongs requires an external ID to assume,
  then must provide that value here.
* No default

remote.sts.assume_role.duration_secs = <integer>
* The duration, in seconds, of the role session.
* The value specified can range from 900 seconds (15 minutes) up to
  the maximum session duration set for the role.
* If you specify a value higher than this setting or the administrator setting (whichever is lower),
  the operation fails. For example, if you specify a session duration of 12 hours,
  but your administrator set the maximum session duration to 6 hours, your operation fails.
* Default: 3600

authMethod = <string>
* The authentication method used to access the remote destination.
* Optional.
* Do not configure this setting in outputs.conf. The system populates
* this setting when you choose an Authentication Method in the New or Edit
* Destination setup window in Splunk Web.
* Choosing "Access key and Secret key" in Splunk Web sets this
* setting to "basic".
* Choosing "IAM role" in Splunk Web sets this setting to "iam".
* No default.

partitionBy = legacy | (year|month|day) [, sourcetype]
* Specifies schema to partition and store events forwarded to this destination.
  Any setting will override the global partitionBy settings of [rfs] stanza.
  Refer to the detailed description of this property under global [rfs] stanza
  and how it affects the file storage path.
* Default: Inherited partitionBy setting from the global [rfs] stanza.

dropEventsOnUploadError = <boolean>
* Whether or not the ingest actions feature drops events if it encounters an
  error when uploading events to remote storage.
* A value of "true" means that, if there is an error writing to a remote file system, the
  error will be logged, and the events in that batch dropped. Ingest will not be
  blocked, but data might be lost.
* A value of "false" means, if there is an error writing to a remote file system, the
  error will be logged, and events will NOT be dropped. splunkd will continually
  attempt to write the batch. Because events are not dropped, this might cause
  queues to become blocked, and data ingestion to stop.
* This setting is optional.
* Default: Inherited dropEventsOnUploadError setting from the global [rfs] stanza.

batchSizeThresholdKB = <integer>
* The size, in kilobytes, of the uncompressed events in the RfsOutputProcessor send buffer.
* RfsOutputProcessor batches events before flushing them to destination.
* If the current buffer size is greater than 'batchSizeThresholdKB' kilobytes, then
* the data will be written to the destination immediately.
```

```
* If you increase this setting, you may also want to increase the value of
  server.conf/[queue:rfsQueue]/maxSize.
* Default: Inherited batchSizeThresholdKB setting from the global [rfs] stanza.

batchTimeout = <integer>
* RfsOutputProcessor batches events before flushing to the destination.
* If a batch has not hit any other criteria for being flushed, and
  the batch is at least this many seconds old, flush the batch.
* Default: Inherited batchTimeout setting from the global [rfs] stanza.

compression = none|gzip|lz4|zstd
* Sets the algorithm to use for compressing files before writing to the destination.
* The RfsOutputProcessor writes files with the appropriate extension for the compression
  algorithm, for example, .zst for zstd, .gz for gzip and .lz4 for lz4.
* Default: Inherited compression setting from the global [rfs] stanza.

compressionLevel = <integer>
* Sets compression level for the specified compression algorithm,
  when RfsOutputProcessor writes files. Must be between 0 and 10.
* Default: Inherited compressionLevel setting from the global [rfs] stanza.

format = json|ndjson|raw
* Specifies output format when RfsOutputProcessor writes events into files
  on the destination.
* json: The file will include a JSON array. Each event will be element of
  the JSON array.
* ndjson: The file will include multiple JSON objects separated by a newline
  character. Each event is corresponding to one JSON object.
* raw: The file includes multiple raw events separated by a newline character.
* Default: Inherited format setting from the global [rfs] stanza.

format.json.index_time_fields = <boolean>
* Specifies whether to include index-time fields when RfsOutputProcessor
  writes events to the destination in HEC JSON format.
* Default: Inherited format.json.index_time_fields setting from the global
  [rfs] stanza.

format.ndjson.index_time_fields = <boolean>
* Specifies whether to include index-time fields when RfsOutputProcessor
  writes events to the destination in new line delimited JSON format.
* Default: Inherited format.ndjson.index_time_fields setting from the global
  [rfs] stanza.
```

Cloud Processing Queue Output

```
[cloud_processing_queue]

* This section explains possible settings for configuring a cloud processing queue.
* Each cloud_processing_queue stanza represents an individually configured cloud
  processing queue output.
* NOTE: Only 1 cloud processing queue stanza is supported as an
  output queue.

cloud_processing_queue.* = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* This section explains possible settings for configuring a cloud processing queue.
* With cloud processing queues, the indexer might require additional configuration,
```

```
  which is specific to the type of cloud processing queue.
  You can pass configuration information to the indexer by specifying the settings
  through the following schema: cloud_processing_queue.<scheme>.<config-variable> = <value>.
  For example:
  cloud_processing_queue.cp_queue.encoding_format = s2s
* No default.

cloud_processing_queue.type = cp_queue
* Currently not supported. This setting is related to a feature that is
  still under development.
* Required.
* Specifies the cloud processing queue type, for example, CP Queue.
```

Cloud Processing Queue (CP Queue) specific settings

```
cloud_processing_queue.cp_queue.encoding_format = s2s
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies the encoding format used to write data to the
  cloud processing queue.
* Default: s2s

cloud_processing_queue.cp_queue.retry_policy = max_count|none
* Sets the retry policy to use for cloud processing queue operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a queue operation is
    retried upon intermittent failure. Set "max_count" with the
    'max_count.max_retries_per_part' setting.
  + "none": Do not retry file operations upon failure.
* Optional.
* Default: max_count

cloud_processing_queue.cp_queue.max_count.max_retries_per_part = <unsigned integer>
* When the 'cloud_processing_queue.cp_queue.retry_policy' setting is "max_count",
  sets the maximum number of times a queue operation will be retried upon intermittent
  failure.
* Optional.
* Default: 3

cloud_processing_queue.cp_queue.large_message_store.sslVerifyServerCert = <boolean>
* A value of "true" means the Splunk platform verifies the certificate presented by the S3
  server and checks that the common name and alternate name match the ones
  specified in 'cloud_processing_queue.cp_queue.large_message_store.sslCommonNameToCheck' and
  'cloud_processing_queue.cp_queue.large_message_store.sslAltNameToCheck'.
* Default: false

cloud_processing_queue.cp_queue.large_message_store.sslVersions = <comma-separated list>
* A list of TLS versions to use to connect to the large message store.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list
  but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Default: tls1.2
```

```
cloud_processing_queue.cp_queue.large_message_store.sslRootCAPath = <string>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together.
  The S3 certificate will be validated against the CAs present in this file.
* Default: The value of [sslConfig]/'caCertFile' in server.conf

cloud_processing_queue.cp_queue.large_message_store.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* You must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Default: TLSv1+HIGH:TLSv1.2+HIGH:@STRENGTH

cloud_processing_queue.cp_queue.large_message_store.ecdhCurves = <comma-separated list>
* ECDH curves to use for ECDH key negotiation.
* Specify the curves in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only supports named curves specified
  by their short names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: not set

cloud_processing_queue.cp_queue.large_message_store.encryption_scheme = sse-s3 | none
* Currently not supported. This setting is related to a feature that is
  still under development.
* The encryption scheme used by remote storage.
* Default: none.

cloud_processing_queue.cp_queue.large_message_store.key_refresh_interval = <string>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The time interval to refresh primary key.
* Default: 24h
```

outputs.conf.example

```
#   Version 9.2.2
#
# This file contains an example outputs.conf.  Use this file to configure
# forwarding in a distributed set up.
#
# To use one or more of these configurations, copy the configuration block into
# outputs.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# Specify a target group for an IP:PORT which consists of a single receiver.
# This is the simplest possible configuration; it sends data to the host at
# 10.1.1.197 on port 9997.

[tcpout:group1]
server=10.1.1.197:9997
```

```
# Specify a target group for a hostname which consists of a single receiver.

[tcpout:group2]
server=myhost.Splunk.com:9997


# Specify a target group made up of two receivers.  In this case, the data will
# be distributed using AutoLB between these two receivers.  You can specify as
# many receivers as you wish here. You can combine host name and IP if you
# wish.
# NOTE: Do not use this configuration with SplunkLightForwarder.

[tcpout:group3]
server=myhost.Splunk.com:9997,10.1.1.197:6666


# You can override any of the global configuration values on a per-target group
# basis.  All target groups that do not override a global config will inherit
# the global config.

# Send every event to a receiver at foo.Splunk.com:9997 with a maximum queue
# size of 100,500 events.

[tcpout:group4]
server=foo.Splunk.com:9997
heartbeatFrequency=45
maxQueueSize=100500

# Send data to a receiving system that controls access by tokens.
# NOTE: token value is encrypted. Encryption is done by REST endpoint while saving.
[tcpout:group4]
server=foo.Splunk.com:9997
token=$1$/fRSBT+2APNAyCB7tlcgOyLnAtqAQFC8NI4TGA2wX4JHfN5d9g==

# Clone events to groups indexer1 and indexer2. Also, index all this data
# locally as well.

[tcpout]
indexAndForward=true

[tcpout:indexer1]
server=Y.Y.Y.Y:9997

[tcpout:indexer2]
server=X.X.X.X:6666


# Clone events between two data balanced groups.

[tcpout:indexer1]
server=A.A.A.A:1111, B.B.B.B:2222

[tcpout:indexer2]
server=C.C.C.C:3333, D.D.D.D:4444

# Syslout output configuration
# This example sends only events generated by the splunk daemon to a remote
# syslog host in syslog-compliant format:

[syslog:syslog-out1]
```

```
disabled = false
server = X.X.X.X:9099
type = tcp
priority = <34>
timestampformat = %b %e %H:%M:%S


# Auto Load Balancing
# This example balances output between two indexers listening on
# port 4433: 192.0.2.100:4433 and 192.0.2.101:4433.
# To achieve this you'd create a DNS entry for 'splunkLB' pointing
# to the two IP addresses of your indexers:
#
#   $ORIGIN example.com.
#   splunkLB A 192.0.2.100
#   splunkLB A 192.0.2.101

[tcpout]
defaultGroup = lb

[tcpout:lb]
server = splunkLB.example.com:4433

# Alternatively, you can use autoLB directly without DNS:

[tcpout]
defaultGroup = lb

[tcpout:lb]
server = 192.0.2.100:4433, 192.0.2.101:4433


# Compression
#
# This example sends compressed events to the remote indexer.
# If set to "true", you do not need to set the 'compressed' setting to
  "true" in the inputs.conf file on the receiver for compression
  of data to occur.
# This setting applies to non-SSL forwarding only. For SSL forwarding with
  compression, Splunk software uses the 'useClientSSLCompression' setting.

[tcpout]
server = splunkServer.example.com:4433
compressed = true


# SSL
#
# This example sends events to an indexer via SSL using splunk's
# self signed cert:

[tcpout]
server = splunkServer.example.com:4433
sslPassword = password
clientCert = $SPLUNK_HOME/etc/auth/server.pem

#
# The following example shows how to route events to syslog server
# This is similar to tcpout routing, but DEST_KEY is set to _SYSLOG_ROUTING
#

# 1. Edit $SPLUNK_HOME/etc/system/local/props.conf and set a TRANSFORMS-routing
```

```
#    attribute:
[default]
TRANSFORMS-routing=errorRouting

[syslog]
TRANSFORMS-routing=syslogRouting

# 2. Edit $SPLUNK_HOME/etc/system/local/transforms.conf and set errorRouting
#    and syslogRouting rules:
[errorRouting]
REGEX=error
DEST_KEY=_SYSLOG_ROUTING
FORMAT=errorGroup

[syslogRouting]
REGEX=.
DEST_KEY=_SYSLOG_ROUTING
FORMAT=syslogGroup

# 3. Edit $SPLUNK_HOME/etc/system/local/outputs.conf and set which syslog
#    outputs go to with servers or groups:
[syslog]
defaultGroup=everythingElseGroup

[syslog:syslogGroup]
server = 10.1.1.197:9997

[syslog:errorGroup]
server=10.1.1.200:9999

[syslog:everythingElseGroup]
server=10.1.1.250:6666

#
# Perform selective indexing and forwarding
#
# Using a heavy forwarder, you can index and store data locally, and
# forward the data out to a receiving indexer. In the example, by
# setting the defaultGroup to a non-existent group named "noforward",
# the forwarder only forwards data that has been routed using explicit
# target groups defined in the inputs.conf


# 1. In outputs.conf:
[tcpout]
defaultGroup = noforward

[indexAndForward]
index=true
selectiveIndexing=true

[tcpout:indexers]
server = 10.1.1.197:9997, 10.1.1.200:9997

# 2. In inputs.conf, add _INDEX_AND_FORWARD_ROUTING to the input
 stanza for any data that you want to index locally, or
_TCP_ROUTING=<target_group> for data to be forwarded.

[monitor:///var/log/messages/]
_INDEX_AND_FORWARD_ROUTING=local

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

```
_TCP_ROUTING=indexers

# Output to S3 for Ingest Actions

# For example, sending to an AWS bucket "buttercup-bucket", with a prefix
# in front of all paths "some-prefix", along with encryption using AWS
# SSE-S3 to the us-west-2 region:

[rfs:s3]
path = s3://buttercup-bucket/some-prefix
remote.s3.encryption = sse-s3
remote.s3.endpoint = https://s3.us-west-2.amazonaws.com
remote.s3.signature_version = v4
remote.s3.supports_versioning = false
remote.s3.access_key = <access key here>
remote.s3.secret_key = <secret key here>
```

passwords.conf

The following are the spec and example files for `passwords.conf`.

passwords.conf.spec

```
#   Version 9.2.2
#
# This file maintains the credential information for a given app in Splunk Enterprise.
#
# There is no global, default passwords.conf. Instead, anytime a user creates
# a new user or edit a user onwards hitting the storage endpoint
# will create this passwords.conf file which gets replicated
# in a search head clustering enviornment.
# Note that passwords.conf is only created from 6.3.0 release.
#
# You must restart Splunk Enterprise to reload manual changes to passwords.conf.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[credential:<realm>:<username>:]

```
password = <password>
* The password that corresponds to the given username for the given realm.
* NOTE: The realm is optional.
* The password can be in clear text, however when saved from splunkd the
  password will always be encrypted.
```

passwords.conf.example

```
#   Version 9.2.2
#
# The following are example passwords.conf configurations. Configure properties for
# your custom application.
#
# There is NO DEFAULT passwords.conf. The file only gets created once you add/edit
```

```
# a credential information via the storage endpoint as follows.
#
# The POST request to add user1 credentials to the storage/password endpoint
# curl -k -u admin:changeme https://localhost:8089/servicesNS/nobody/search/storage/passwords -d name=user1
-d password=changeme2
#
# The GET request to list all the credentials stored at the storage/passwords endpoint
# curl -k -u admin:changeme https://localhost:8089/services/storage/passwords
#
# To use one or more of these configurations, copy the configuration block into
# passwords.conf in $SPLUNK_HOME/etc/<apps>/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#

[credential::testuser:]
password = changeme
```

procmon-filters.conf

The following are the spec and example files for `procmon-filters.conf`.

procmon-filters.conf.spec

```
#   Version 9.2.2
#
# *** DEPRECATED ***
#
#
# This file contains potential attribute/value pairs to use when configuring
# Windows registry monitoring. The procmon-filters.conf file contains the
# regular expressions you create to refine and filter the processes you want
# Splunk to monitor. You must restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

#### find out if this file is still being used.
```

[<stanza name>]

```
* The name of the filter being defined.

proc = <string>
* A regular expression that specifies process image that you want
  the Splunk platform to monitor.
* No default.

type = <string>
* A regular expression that specifies the type(s) of process events
  that you want the Splunk platform to monitor.
* No default

hive = <string>
```

```
* Not used in this context, but should always have value ".*"
```

procmon-filters.conf.example

```
#   Version 9.2.2
#
# This file contains example registry monitor filters. To create your own
# filter, use the information in procmon-filters.conf.spec.
#
# To use one or more of these configurations, copy the configuration block into
# procmon-filters.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[default]
hive = .*

[not-splunk-optimize]
proc = (?<!splunk-optimize.exe)$
type = create|exit|image
```

props.conf

The following are the spec and example files for `props.conf`.

props.conf.spec

```
#   Version 9.2.2
#
# This file contains possible setting/value pairs for configuring Splunk
# software's processing properties through props.conf.
#
# Props.conf is commonly used for:
#
# * Configuring line breaking for multi-line events.
# * Setting up character set encoding.
# * Allowing processing of binary files.
# * Configuring timestamp recognition.
# * Configuring event segmentation.
# * Overriding automated host and source type matching. You can use
#   props.conf to:
#     * Configure advanced (regular expression-based) host and source
        type overrides.
#     * Override source type matching for data from a particular source.
#     * Set up rule-based source type recognition.
#     * Rename source types.
# * Anonymizing certain types of sensitive incoming data, such as credit
#   card or social security numbers, using sed scripts.
# * Routing specific events to a particular index, when you have multiple
#   indexes.
# * Creating new index-time field extractions, including header-based field
#   extractions.
#   NOTE: Do not add to the set of fields that are extracted
```

```
#         at index time unless it is absolutely necessary because there are
#         negative performance implications.
# * Defining new search-time field extractions. You can define basic
#   search-time field extractions entirely through props.conf, but a
#   transforms.conf component is required if you need to create search-time
#   field extractions that involve one or more of the following:
#       * Reuse of the same field-extracting regular expression across
#         multiple sources, source types, or hosts.
#       * Application of more than one regular expression (regex) to the
#         same source, source type, or host.
#       * Delimiter-based field extractions (they involve field-value pairs
#         that are separated by commas, colons, semicolons, bars, or
#         something similar).
#       * Extraction of multiple values for the same field (multivalued
#         field extraction).
#       * Extraction of fields with names that begin with numbers or
#         underscores.
# * Setting up lookup tables that look up fields from external sources.
# * Creating field aliases.
#
# NOTE: Several of the above actions involve a corresponding transforms.conf
# configuration.
#
# You can find more information on these topics by searching the Splunk
# documentation (http://docs.splunk.com/Documentation/Splunk).
#
# There is a props.conf in $SPLUNK_HOME/etc/system/default/.  To set custom
# configurations, place a props.conf in $SPLUNK_HOME/etc/system/local/. For
# help, see props.conf.example.
#
# You can enable configurations changes made to props.conf by typing the
# following search string in Splunk Web:
#
# | extract reload=T
#
# To learn more about configuration files (including precedence) see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# For more information about using props.conf in conjunction with
# distributed Splunk deployments, see the Distributed Deployment Manual.
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top
#     of the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file wins.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

[<spec>]
* This stanza enables properties for a given <spec>.
* A props.conf file can contain multiple stanzas for any number of
  different <spec>.
* Follow this stanza name with any number of the following setting/value
```

```
  pairs, as appropriate for what you want to do.
* If you do not set a setting for a given <spec>, the default is used.

<spec> can be:
1. <sourcetype>, the source type of an event.
2. host::<host>, where <host> is the host, or host-matching pattern, for an
                 event.
3. source::<source>, where <source> is the source, or source-matching
                     pattern, for an event.
4. rule::<rulename>, where <rulename> is a unique name of a source type
                     classification rule.
5. delayedrule::<rulename>, where <rulename> is a unique name of a delayed
                            source type classification rule.
                            These are only considered as a last resort
                            before generating a new source type based on the
                            source seen.

**[<spec>] stanza precedence:**

For settings that are specified in multiple categories of matching [<spec>]
stanzas, [host::<host>] settings override [<sourcetype>] settings.
Additionally, [source::<source>] settings override both [host::<host>]
and [<sourcetype>] settings.

**Considerations for Windows file paths:**

When you specify Windows-based file paths as part of a [source::<source>]
stanza, you must escape any backslashes contained within the specified file
path.

Example: [source::c:\\path_to\\file.txt]

**[<spec>] stanza patterns:**

When setting a [<spec>] stanza, you can use the following regex-type syntax:
... recurses through directories until the match is met
    or equivalently, matches any number of characters.
*   matches anything but the path separator 0 or more times.
    The path separator is '/' on unix, or '\' on Windows.
    Intended to match a partial or complete directory or filename.
|   is equivalent to 'or'
( ) are used to limit scope of |.
\\ = matches a literal backslash '\'.

Example: [source::....(?<!tar.)(gz|bz2)]

 This matches any file ending with '.gz' or '.bz2', provided this is not
 preceded by 'tar.', so tar.bz2 and tar.gz would not be matched.

**[source::<source>] and [host::<host>] stanza match language:**

Match expressions must match the entire name, not just a substring. Match
expressions are based on a full implementation of Perl-compatible regular
expressions (PCRE) with the translation of "...", "*", and "." Thus, "."
matches a period, "*" matches non-directory separators, and "..." matches
any number of any characters.

For more information search the Splunk documentation for "specify input
paths with wildcards".

**[<spec>] stanza pattern collisions:**
```

Suppose the source of a given input matches multiple [source::<source>] patterns. If the [<spec>] stanzas for these patterns each supply distinct settings, Splunk software applies all of these settings.

However, suppose two [<spec>] stanzas supply the same setting. In this case, Splunk software chooses the value to apply based on the ASCII order of the patterns in question.

For example, take this source:

```
source::az
```

and the following colliding patterns:

```
[source::...a...]
sourcetype = a

[source::...z...]
sourcetype = z
```

In this case, the settings provided by the pattern [source::...a...] take precedence over those provided by [source::...z...], and sourcetype ends up with "a" as its value.

To override this default ASCII ordering, use the priority key:

```
[source::...a...]
sourcetype = a
priority = 5

[source::...z...]
sourcetype = z
priority = 10
```

Assigning a higher priority to the second stanza causes sourcetype to have the value "z".

Case-sensitivity for [<spec>] stanza matching:

By default, [source::<source>] and [<sourcetype>] stanzas match in a case-sensitive manner, while [host::<host>] stanzas match in a case-insensitive manner. This is a convenient default, given that DNS names are case-insensitive.

To force a [host::<host>] stanza to match in a case-sensitive manner use the "(?-i)" option in its pattern.

For example:

```
[host::foo]
FIELDALIAS-a = a AS one

[host::(?-i)bar]
FIELDALIAS-b = b AS two
```

The first stanza actually applies to events with host values of "FOO" or "Foo" . The second stanza, on the other hand, does not apply to events with host values of "BAR" or "Bar".

Building the final [<spec>] stanza:

The final [<spec>] stanza is built by layering together (1) literal-matching

```
stanzas (stanzas which match the string literally) and (2) any
regex-matching stanzas, according to the value of the priority field.

If not specified, the default value of the priority key is:
* 0 for pattern-matching stanzas.
* 100 for literal-matching stanzas.

NOTE: Setting the priority key to a value greater than 100 causes the
pattern-matched [<spec>] stanzas to override the values of the
literal-matching [<spec>] stanzas.

The priority key can also be used to resolve collisions
between [<sourcetype>] patterns and [host::<host>] patterns. However, be aware
that the priority key does *not* affect precedence across <spec> types. For
example, [<spec>] stanzas with [source::<source>] patterns take priority over
stanzas with [host::<host>] and [<sourcetype>] patterns, regardless of their
respective priority key values.


#*******************************************************************************
# The possible setting/value pairs for props.conf, and their
# default values, are:
#*******************************************************************************

priority = <number>
* Overrides the default ASCII ordering of matching stanza names

# International characters and character encoding.

CHARSET = <string>
* When set, Splunk software assumes the input from the given [<spec>] is in
  the specified encoding.
* Can only be used as the basis of [<sourcetype>] or [source::<spec>],
  not [host::<spec>].
* A list of valid encodings can be retrieved using the command "iconv -l" on
  most *nix systems.
* If an invalid encoding is specified, a warning is logged during initial
  configuration and further input from that [<spec>] is discarded.
* If the source encoding is valid, but some characters from the [<spec>] are
  not valid in the specified encoding, then the characters are escaped as
  hex (for example, "\xF3").
* When set to "AUTO", Splunk software attempts to automatically determine the
  character encoding and convert text from that encoding to UTF-8.
* For a complete list of the character sets Splunk software automatically
  detects, see the online documentation.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default (on Windows machines): AUTO
* Default (otherwise): UTF-8
```

Line breaking

```
# Use the following settings to define the length of a line.

TRUNCATE = <non-negative integer>
* The default maximum line length, in bytes.
* Although this is in bytes, line length is rounded down when this would
```

```
  otherwise land mid-character for multi-byte characters.
* Set to 0 if you never want truncation (very long lines are, however, often
  a sign of garbage data).
* Default: 10000

LINE_BREAKER = <regular expression>
* Specifies a regex that determines how the raw text stream is broken into
  initial events, before line merging takes place. (See the SHOULD_LINEMERGE
  setting, below.)
* The regex must contain a capturing group -- a pair of parentheses which
  defines an identified subcomponent of the match.
* Wherever the regex matches, Splunk software 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.
* The contents of the first capturing group are discarded, and are not
  present in any event. You are telling Splunk software that this text comes
  between lines.
* NOTE: You get a significant boost to processing speed when you use
  LINE_BREAKER to delimit multi-line events (as opposed to using
  SHOULD_LINEMERGE to reassemble individual lines into multi-line events).
  * When using LINE_BREAKER to delimit events, SHOULD_LINEMERGE should be set
    to false, to ensure no further combination of delimited events occurs.
  * Using LINE_BREAKER to delimit events is discussed in more detail in the
    documentation. Search the documentation for "configure event line breaking"
    for details.
* Default: ([\r\n]+) (Data is broken into an event for each line,
  delimited by any number of carriage return or newline characters.)


** Special considerations for LINE_BREAKER with branched expressions  **

When using LINE_BREAKER with completely independent patterns separated by
pipes, some special issues come into play.
    EG. LINE_BREAKER = pattern1|pattern2|pattern3

NOTE: This is not about all forms of alternation. For instance, there is
nothing particularly special about
    example: LINE_BREAKER = ([\r\n])+(one|two|three)
where the top level remains a single expression.

CAUTION: Relying on these rules is NOT encouraged.  Simpler is better, in
both regular expressions and the complexity of the behavior they rely on.
If possible, reconstruct your regex to have a leftmost capturing group
that always matches.

It might be useful to use non-capturing groups if you need to express a group
before the text to discard.
    Example: LINE_BREAKER = (?:one|two)([\r\n]+)
    * This matches the text one, or two, followed by any amount of
      newlines or carriage returns.  The one-or-two group is non-capturing
      via the ?: prefix and is skipped by LINE_BREAKER.

* A branched expression can match without the first capturing group
  matching, so the line breaker behavior becomes more complex.
  Rules:
  1: If the first capturing group is part of a match, it is considered the
     linebreak, as normal.
  2: If the first capturing group is not part of a match, the leftmost
     capturing group which is part of a match is considered the linebreak.
  3: If no capturing group is part of the match, the linebreaker assumes
     that the linebreak is a zero-length break immediately preceding the match.
```

```
Example 1:  LINE_BREAKER = end(\n)begin|end2(\n)begin2|begin3

  * A line ending with 'end' followed a line beginning with 'begin' would
    match the first branch, and the first capturing group would have a match
    according to rule 1.  That particular newline would become a break
    between lines.
  * A line ending with 'end2' followed by a line beginning with 'begin2'
    would match the second branch and the second capturing group would have
    a match.  That second capturing group would become the linebreak
    according to rule 2, and the associated newline would become a break
    between lines.
  * The text 'begin3' anywhere in the file at all would match the third
    branch, and there would be no capturing group with a match.  A linebreak
    would be assumed immediately prior to the text 'begin3' so a linebreak
    would be inserted prior to this text in accordance with rule 3.  This
    means that a linebreak occurs before the text 'begin3' at any
    point in the text, whether a linebreak character exists or not.

Example 2: Example 1 would probably be better written as follows.  This is
           not equivalent for all possible files, but for most real files
           would be equivalent.

           LINE_BREAKER = end2?(\n)begin(2|3)?

LINE_BREAKER_LOOKBEHIND = <integer>
* The number of bytes before the end of the raw data chunk
  to which Splunk software should apply the 'LINE_BREAKER' regex.
* When there is leftover data from a previous raw chunk,
  LINE_BREAKER_LOOKBEHIND indicates the number of bytes before the end of
  the raw chunk (with the next chunk concatenated) where Splunk software
  applies the LINE_BREAKER regex.
* You might want to increase this value from its default if you are
  dealing with especially large or multi-line events.
* Default: 100

# Use the following settings to specify how multi-line events are handled.

SHOULD_LINEMERGE = <boolean>
* Whether or not to combine several lines of data into a single
  multiline event, based on the configuration settings listed in
  this subsection.
* When you set this to "true", Splunk software combines several lines of data
  into a single multi-line event, based on values you configure
  in the following settings.
* When you set this to "false", Splunk software does not combine lines of
  data into multiline events.
* Default: true

# When SHOULD_LINEMERGE is set to true, use the following settings to
# define how Splunk software builds multi-line events.

BREAK_ONLY_BEFORE_DATE = <boolean>
* Whether or not to create a new event if a new line with a date is encountered
  in the data stream.
* When you set this to "true", Splunk software creates a new event only if it
  encounters a new line with a date.
  * NOTE: When using DATETIME_CONFIG = CURRENT or NONE, this setting is not
    meaningful, as timestamps are not identified.
* Default: true

BREAK_ONLY_BEFORE = <regular expression>
* When set, Splunk software creates a new event only if it encounters a new
```

```
  line that matches the regular expression.
* Default: empty string

MUST_BREAK_AFTER = <regular expression>
* When set, Splunk software creates a new event for the next input line only
  if the regular expression matches the current line.
* It is possible for the software to break before the current line if
  another rule matches.
* Default: empty string

MUST_NOT_BREAK_AFTER = <regular expression>
* When set, and the current line matches the regular expression, Splunk software
  does not break on any subsequent lines until the MUST_BREAK_AFTER expression
  matches.
* Default: empty string

MUST_NOT_BREAK_BEFORE = <regular expression>
* When set, and the current line matches the regular expression, Splunk
  software does not break the last event before the current line.
* Default: empty string

MAX_EVENTS = <integer>
* The maximum number of input lines to add to any event.
* Splunk software breaks after it reads the specified number of lines.
* Default: 256

ROUTE_EVENTS_OLDER_THAN = <non-negative integer>[s|m|h|d]
* If set, AggregatorProcessor routes events older than 'ROUTE_EVENTS_OLDER_THAN'
  to nullQueue after timestamp extraction.
* Default: no default

# Use the following settings to handle better load balancing from UF.
# NOTE: The EVENT_BREAKER properties are applicable for Splunk Universal
# Forwarder instances only.

EVENT_BREAKER_ENABLE = <boolean>
* Whether or not a universal forwarder (UF) uses the 'ChunkedLBProcessor'
  data processor to improve distribution of events to receiving
  indexers for a given source type.
* When set to true, a UF splits incoming data with a
  light-weight chunked line breaking processor ('ChunkedLBProcessor')
  so that data is distributed fairly evenly amongst multiple indexers.
* When set to false, a UF uses standard load-balancing methods to
  send events to indexers.
* Use this setting on a UF to indicate that data
  should be split on event boundaries across indexers, especially
  for large files.
* This setting is only valid on universal forwarder instances.
* Default: false

# Use the following to define event boundaries for multi-line events
# For single-line events, the default settings should suffice

EVENT_BREAKER = <regular expression>
* A regular expression that specifies the event boundary for a
  universal forwarder to use to determine when it can send events
  to an indexer.
* The regular expression must contain a capturing group
  (a pair of parentheses that defines an identified sub-component
  of the match.)
* When the UF finds a match, it considers the first capturing group
  to be the end of the previous event, and the end of the capturing group
```

```
  to be the beginning of the next event.
* At this point, the forwarder can then change the receiving indexer
  based on these event boundaries.
* This setting is only active if you set 'EVENT_BREAKER_ENABLE' to
  "true", only works on universal forwarders, and
  works best with multiline events.
* Default: "([\r\n]+)"

LB_CHUNK_BREAKER = <regular expression>
* A regular expression that specifies the event boundary for a
  universal forwarder to use to determine when it can send events
  to an indexer.
* The regular expression must contain a capturing group
  (a pair of parentheses that defines an identified sub-component
  of the match.)
* When the UF finds a match, it considers the first capturing group
  to be the end of the previous event, and the end of the capturing group
  to be the beginning of the next event.
* Splunk software discards the contents of the first capturing group.
  This content will not be present in any event, as Splunk software
  considers this text to come between lines.
* At this point, the forwarder can then change the receiving indexer
  based on these event boundaries.
* This is only used if [httpout] is configured in outputs.conf
* Default: ([\r\n]+)

LB_CHUNK_BREAKER_TRUNCATE = <non-negative integer>
* The maximum length of data chunk sent by LB_CHUNK_BREAKER, in bytes.
* Although this is in bytes, length is rounded down when this would
  otherwise land mid-character for multi-byte characters.
* Default: 2000000
```

Timestamp extraction configuration

```
DATETIME_CONFIG = [<filename relative to $SPLUNK_HOME> | CURRENT | NONE]
* Specifies which file configures the timestamp extractor, which identifies
  timestamps from the event text.
* This setting may also be set to "NONE" to prevent the timestamp
  extractor from running or "CURRENT" to assign the current system time to
  each event.
  * "CURRENT" sets the time of the event to the time that the event was
    merged from lines, or worded differently, the time it passed through the
    aggregator processor.
  * "NONE" leaves the event time set to whatever time was selected by
    the input layer
    * For data sent by Splunk forwarders over the Splunk-to-Splunk protocol,
      the input layer is the time that was selected on the forwarder by
      its input behavior (as below).
    * For file-based inputs (monitor, batch) the time chosen is the
      modification timestamp on the file being read.
    * For other inputs, the time chosen is the current system time when
      the event is read from the pipe/socket/etc.
  * Both "CURRENT" and "NONE" explicitly disable the per-text timestamp
    identification, so the default event boundary detection
    (BREAK_ONLY_BEFORE_DATE = true) is likely to not work as desired.  When
    using these settings, use 'SHOULD_LINEMERGE' and/or the 'BREAK_ONLY_*' ,
    'MUST_BREAK_*' settings to control event merging.
* For more information on 'DATETIME_CONFIG' and datetime.xml, see "Configure
  advanced timestamp recognition with datetime.xml" in the Splunk Documentation.
```

```
* Default: /etc/datetime.xml (for example, $SPLUNK_HOME/etc/datetime.xml).

TIME_PREFIX = <regular expression>
* If set, Splunk software scans the event text for a match for this regex
  in event text before attempting to extract a timestamp.
* The timestamping algorithm only looks for a timestamp in the text
  following the end of the first regex match.
* For example, if 'TIME_PREFIX' is set to "abc123", only text following the
  first occurrence of the text abc123 is used for timestamp extraction.
* If the 'TIME_PREFIX' cannot be found in the event text, timestamp extraction
  does not occur.
* Default: empty string

MAX_TIMESTAMP_LOOKAHEAD = <integer>
* The number of characters into an event Splunk software should look
  for a timestamp.
* This constraint to timestamp extraction is applied from the point of the
  'TIME_PREFIX'-set location.
* For example, if 'TIME_PREFIX' positions a location 11 characters into the
  event, and MAX_TIMESTAMP_LOOKAHEAD is set to 10, timestamp extraction is
  constrained to characters 11 through 20.
* 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
  'LINE_BREAKER' is redefined for event splitting).
* Default: 128

TIME_FORMAT = <strptime-style format>
* Specifies a "strptime" format string to extract the date.
* "strptime" is an industry standard for designating time formats.
* For more information on strptime, see "Configure timestamp recognition" in
  the online documentation.
* TIME_FORMAT starts reading after the TIME_PREFIX. If both are specified,
  the TIME_PREFIX regex must match up to and including the character before
  the TIME_FORMAT date.
* For good results, the <strptime-style format> should describe the day of
  the year and the time of day.
* Default: empty string

DETERMINE_TIMESTAMP_DATE_WITH_SYSTEM_TIME = <boolean>
* Whether or not the Splunk platform uses the current system time to
  determine the date of an event timestamp that has no date.
* If set to "true", the platform uses the system time to determine the
  date for an event that has a timestamp without a date.
  * If the future event has a timestamp that is less than three hours
    later than the current system time, then the platform presumes
    that the timestamp date for that event is the current date.
  * Otherwise, it presumes that the timestamp date is in the future, and
    uses the previous day's date instead.
* If set to "false", the platform uses the last successfully-parsed
  timestamp to determine the timestamp date for the event.
* Default: false

TZ = <timezone identifier>
* The algorithm for determining the time zone for a particular event is as
  follows:
  * If the event has a timezone in its raw text (for example, UTC, -08:00),
  use that.
  * If TZ is set to a valid timezone string, use that.
  * If the event was forwarded, and the forwarder-indexer connection uses
  the version 6.0 and higher forwarding protocol, use the timezone provided
  by the forwarder.
```

```
  * Otherwise, use the timezone of the system that is running splunkd.
* Default: empty string

TZ_ALIAS = <key=value>[,<key=value>]...
* Provides Splunk software 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 many expansions.
* There is no requirement to use 'TZ_ALIAS' if the traditional Splunk software
  default mappings for these values have been as expected. For example, EST
  maps to the Eastern US by default.
* Has no effect on the 'TZ' value. This only affects timezone strings from event
  text, either from any configured 'TIME_FORMAT', or from pattern-based guess
  fallback.
* The setting is a list of key=value pairs, separated by commas.
  * 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.
  * The value is another TZ specifier which expresses the desired offset.
  * Example: TZ_ALIAS = EST=GMT+10:00 (See props.conf.example for more/full
    examples)
* Default: not set

MAX_DAYS_AGO = <integer>
* The maximum number of days in the past, from the current date as
  provided by the input layer (For example forwarder current time, or modtime
  for files), that an extracted date can be valid.
* Splunk software still indexes events with dates older than 'MAX_DAYS_AGO'
  with the timestamp of the last acceptable event.
* If no such acceptable event exists, new events with timestamps older
  than 'MAX_DAYS_AGO' uses the current timestamp.
* For example, if MAX_DAYS_AGO = 10, Splunk software applies the timestamp
  of the last acceptable event to events with extracted timestamps older
  than 10 days in the past. If no acceptable event exists, Splunk software
  applies the current timestamp.
* If your data is older than 2000 days, increase this setting.
* Highest legal value: 10951 (30 years).
* Default: 2000 (5.48 years).

MAX_DAYS_HENCE = <integer>
* The maximum number of days in the future, from the current date as
  provided by the input layer(For e.g. forwarder current time, or
  modtime for files), that an extracted date can be valid.
* Splunk software still indexes events with dates more than 'MAX_DAYS_HENCE'
  in the future with the timestamp of the last acceptable event.
* If no such acceptable event exists, new events
  with timestamps after 'MAX_DAYS_HENCE' use the current timestamp.
* For example, if MAX_DAYS_HENCE = 3, Splunk software applies the timestamp of
  the last acceptable event to events with extracted timestamps more than 3
  days in the future. If no acceptable event exists, Splunk software applies
  the current timestamp.
* The default value includes dates from one day in the future.
* If your servers have the wrong date set or are in a timezone that is one
  day ahead, increase this value to at least 3.
* NOTE: False positives are less likely with a smaller window. Change with
  caution.
* Highest legal value: 10950 (30 years).
* Default: 2

MAX_DIFF_SECS_AGO = <integer>
* This setting prevents Splunk software from rejecting events with timestamps
```

```
  that are out of order.
* Do not use this setting to filter events. Splunk software uses
  complicated heuristics for time parsing.
* Splunk software warns you if an event timestamp is more than
  'MAX_DIFF_SECS_AGO' seconds BEFORE the previous timestamp and does not
  have the same time format as the majority of timestamps from the source.
* After Splunk software throws the warning, it only rejects an event if it
  cannot apply a timestamp to the event. (For example, if Splunk software
  cannot recognize the time of the event.)
* If your timestamps are wildly out of order, consider increasing
  this value.
* NOTE: If the events contain time but not date (date determined another way,
  such as from a filename) this check only considers the hour. (No one
  second granularity for this purpose.)
* Highest legal value: 2147483646 (68.1 years).
* Defaults: 3600 (one hour).

MAX_DIFF_SECS_HENCE = <integer>
* This setting prevents Splunk software from rejecting events with timestamps
  that are out of order.
* Do not use this setting to filter events. Splunk software uses
  complicated heuristics for time parsing.
* Splunk software warns you if an event timestamp is more than
  'MAX_DIFF_SECS_HENCE' seconds AFTER the previous timestamp and does not
  have the same time format as the majority of timestamps from the source.
* After Splunk software throws the warning, it only rejects an event if it
  cannot apply a timestamp to the event. (For example, if Splunk software
  cannot recognize the time of the event.)
* If your timestamps are wildly out of order, or you have logs that
  are written less than once a week, consider increasing this value.
* Highest legal value: 2147483646 (68.1 years).
* Default: 604800 (one week).

ADD_EXTRA_TIME_FIELDS = [none | subseconds | all | <boolean>]
* Whether or not Splunk software automatically generates and indexes the
  following keys with events:
  * date_hour, date_mday, date_minute, date_month, date_second, date_wday,
    date_year, date_zone, timestartpos, timeendpos, timestamp.
* These fields are never required, and may be turned off as desired.
* If set to "none" (or false), all indextime data about the timestamp is
  stripped out. This removes the above fields but also removes information
  about the sub-second timestamp granularity. When events are searched,
  only the second-granularity timestamp is returned as part of the
  "_time" field.
* If set to "subseconds", the above fields are stripped out but the data about
  subsecond timestamp granularity is left intact.
* If set to "all" (or true), all of the indextime fields from the time
  parser are included.
* Default: true (Enabled for most data sources.)
```

Structured Data Header Extraction and configuration

```
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.

# These special string delimiters, which are single ASCII characters,
# can be used in the settings that follow, which state
# "You can use the delimiters for structured data header extraction with
```

```
# this setting."
#
# You can only use a single delimiter for any setting.
# It is not possible to configure multiple delimiters or characters per
# setting.
#
# Example of using the delimiters:
#
# FIELD_DELIMITER=space
# * Tells Splunk software to use the space character to separate fields
# in the specified source.
# space           - Space separator (separates on a single space)
# tab / \t        - Tab separator
# fs              - ASCII file separator
# gs              - ASCII group separator
# rs              - ASCII record separator
# us              - ASCII unit separator
#\xHH             - HH is two heaxadecimal digits to use as a separator
                     Example : \x14 - select 0x14 as delimiter
# none            - (Valid for FIELD_QUOTE and HEADER_FIELD_QUOTE only)
                     null termination character separator
# whitespace / ws - (Valid for FIELD_DELIMITER and
                     HEADER_FIELD_DELIMITER only)
                     treats any number of spaces and tabs as a
                     single delimiter

INDEXED_EXTRACTIONS = <CSV|TSV|PSV|W3C|JSON|HEC>
* The type of file that Splunk software should expect for a given source
  type, and the extraction and/or parsing method that should be used on the
  file.
* The following values are valid for 'INDEXED_EXTRACTIONS':
  CSV  - Comma separated value format
  TSV  - Tab-separated value format
  PSV  - pipe ("|")-separated value format
  W3C  - World Wide Web Consortium (W3C) Extended Log File Format
  JSON - JavaScript Object Notation format
  HEC  - Interpret file as a stream of JSON events in the same format as the
         HTTP Event Collector (HEC) input.
* These settings change the defaults for other settings in this subsection
  to appropriate values, specifically for these formats.
* The HEC format lets events overide many details on a per-event basis, such
  as the destination index. Use this value to read data which you know to be
  well-formatted and safe to index with little or no processing, such as
  data generated by locally written tools.
* When 'INDEXED_EXTRACTIONS = JSON' for a particular source type, do not also
  set 'KV_MODE = json' for that source type. This causes the Splunk software to
  extract the JSON fields twice: once at index time, and again at search time.
* Default: not set

METRICS_PROTOCOL = <STATSD|COLLECTD_HTTP>
* Which protocol the incoming metric data is using:
  STATSD:        Supports the statsd protocol, in the following format:
                 <metric name>:<value>|<metric type>
                 Use the 'STATSD-DIM-TRANSFORMS' setting to manually extract
                 dimensions for the above format. Splunk software auto-extracts
                 dimensions when the data has "#" as dimension delimiter
                 as shown below:
                 <metric name>:<value>|<metric type>|#<dim1>:<val1>,
                 <dim2>:<val2>...
  COLLECTD_HTTP: This is data from the write_http collectd plugin being parsed
                 as streaming JSON docs with the _value living in "values" array
                 and the dimension names in "dsnames" and the metric type
```

```
                (for example, counter vs gauge) is derived from "dstypes".
* Default (for event (non-metric) data): not set

STATSD-DIM-TRANSFORMS = <statsd_dim_stanza_name1>,<statsd_dim_stanza_name2>..
* Valid only when 'METRICS_PROTOCOL' is set to "statsd".
* A comma separated list of transforms stanza names which are used to extract
  dimensions from statsd metric data.
* Optional for sourcetypes which have only one transforms stanza for extracting
  dimensions, and the stanza name is the same as that of sourcetype name.
* Stanza names must start with prefix "statsd-dims:"
  For example, in props.conf:
        STATSD-DIM-TRANSFORMS = statsd-dims:extract_ip
  In transforms.conf, stanza should be prefixed also as so:
        [statsd-dims:extract_ip]
* Default: not set

STATSD_EMIT_SINGLE_MEASUREMENT_FORMAT = <boolean>
* Valid only when 'METRICS_PROTOCOL' is set to 'statsd'.
* This setting controls the metric data point format emitted by the statsd
  processor.
* When set to true, the statsd processor produces metric data points in
  single-measurement format. This format allows only one metric measurement per
  data point, with one key-value pair for the metric name
  (metric_name=<metric_name>) and another key-value pair for the measurement
  value (_value=<numerical_value>).
* When set to false, the statsd processor produces metric data points in
  multiple-measurement format. This format allows multiple metric measurements
  per data point, where each metric measurement follows this syntax:
  metric_name:<metric_name>=<numerical_value>
* We recommend you set this to 'true' for statsd data, because the statsd data
  format is single-measurement per data point. This practice enables you to use
  downstream transforms to edit the metric_name if necessary. Multiple-value
  metric data points are harder to process with downstream transforms.
* Default: true

METRIC-SCHEMA-TRANSFORMS = <metric-schema:stanza_name>[,<metric-schema:stanza_name>]...
* A comma-separated list of metric-schema stanza names from transforms.conf
  that the Splunk platform uses to create multiple metrics from index-time
  field extractions of a single log event.
* NOTE: This setting is valid only for index-time field extractions.
  You can set up the TRANSFORMS field extraction configuration to create
  index-time field extractions. The Splunk platform always applies
  METRIC-SCHEMA-TRANSFORMS after index-time field extraction takes place.
* Optional.
* Default: empty

PREAMBLE_REGEX = <regex>
* A regular expression that lets Splunk software ignore "preamble lines",
  or lines that occur before lines that represent structured data.
* When set, Splunk software ignores these preamble lines,
  based on the pattern you specify.
* Default: not set

FIELD_HEADER_REGEX = <regex>
* A regular expression that specifies a pattern for prefixed headers.
* The actual header starts after the pattern. It is not included in
  the header field.
* This setting supports the use of the special characters described above.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

HEADER_FIELD_LINE_NUMBER = <integer>
```

```
* The line number of the line within the specified file or source that
  contains the header fields.
* If set to 0, Splunk software attempts to
  locate the header fields within the file automatically.
* Default: 0

FIELD_DELIMITER = <character>
* Which character delimits or separates fields in the
  specified file or source.
* You can use the delimiters for structured data header extraction with
  this setting.
* This setting supports the use of the special characters described above.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

HEADER_FIELD_DELIMITER = <character>
* Which character delimits or separates header fields in
  the specified file or source.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

HEADER_FIELD_ACCEPTABLE_SPECIAL_CHARACTERS = <string>
* This setting specifies the special characters that are allowed in header
  fields.
* When this setting is not set, the processor replaces all characters in header
  field names that are neither alphanumeric or a space (" ") with underscores.
  * For example, if you import a CSV file, and one of the header field names is
    "field.name", the processor replaces "field.name" with "field_name", and
    imports the field this way.
* If you configure this setting, the processor does not perform a character
  replacement in header field names if the special character it encounters
  matches one that you specify in the setting value.
  * For example, if you configure this setting to ".", the processor does not
    replace the "." characters in header field names with underscores.
* This setting only supports characters with ASCII codes below 128.
* CAUTION: Certain special characters can cause the Splunk instance to
  malfunction.
  * For example, the field name "fieldname=a" is currently sanitized to
    "fieldname_a" and the search query "fieldname_a=val" works fine. If the
    setting is set to "=" and the field name "fieldname=a" is allowed, it could
    result in an invalid-syntax search query "fieldname=a=val".
* Default: empty string

FIELD_QUOTE = <character>
* The character to use for quotes in the specified file
  or source.
* You can use the delimiters for structured data header extraction with
  this setting.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

HEADER_FIELD_QUOTE = <character>
* The character to use for quotes in the header of the
  specified file or source.
* You can use the delimiters for structured data header extraction with
  this setting.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

TIMESTAMP_FIELDS = [ <string>,..., <string>]
* Some CSV and structured files have their timestamp encompass multiple
  fields in the event separated by delimiters.
```

```
* This setting tells Splunk software to specify all such fields which
  constitute the timestamp in a comma-separated fashion.
* If not specified, Splunk software tries to automatically extract the
  timestamp of the event.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

FIELD_NAMES = [ <string>,..., <string>]
* Some CSV and structured files might have missing headers.
* This setting tells Splunk software to specify the header field names directly.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

MISSING_VALUE_REGEX = <regex>
* The placeholder to use in events where no value is present.
* The default can vary if 'INDEXED_EXTRACTIONS' is set.
* Default (if 'INDEXED_EXTRACTIONS' is not set): not set

JSON_TRIM_BRACES_IN_ARRAY_NAMES = <boolean>
* Whether or not the JSON parser for 'INDEXED_EXTRACTIONS' strips curly
  braces from names of fields that are defined as arrays in JSON events.
* When the JSON parser extracts fields from JSON events, by default, it
  extracts array field names with the curly braces that indicate they
  are arrays ("{}") intact.
* For example, given the following partial JSON event:
    {"datetime":"08-20-2015 10:32:25.267 -0700","log_level":"INFO",...,
      data:{...,"fs_type":"ext4","mount_point":["/disk48","/disk22"],...}}

  Because the "mount_point" field in this event is an array of two
  values ("/disk48" and "/disk22"), the JSON parser sees the field as an
  array, and extracts it as such, including the braces that identify
  it as an array. The resulting field name is "data.mount_point{}").
* Set 'JSON_TRIM_BRACES_IN_ARRAY_NAMES' to "true" if you want the JSON
  parser to strip these curly braces from array field names. (In this
  example, the resulting field is instead "data.mount_point").
* CAUTION: Setting this to "true" makes array field names that are extracted
  at index time through the JSON parser inconsistent with search-time
  extraction of array field names through the 'spath' search command.
* Default: false
```

Field extraction configuration

```
NOTE: If this is your first time configuring field extractions in
      props.conf, review the following information first. Additional
      information is also available in the Getting Data In Manual
      in the Splunk Documentation.

There are three different "field extraction types" that you can use to
configure field extractions: TRANSFORMS, REPORT, and EXTRACT. They differ in
two significant ways: 1) whether they create indexed fields (fields
extracted at index time) or extracted fields (fields extracted at search
time), and 2), whether they include a reference to an additional component
called a "field transform," which you define separately in transforms.conf.

**Field extraction configuration: index time versus search time**

Use the TRANSFORMS field extraction type to create index-time field
extractions. Use the REPORT or EXTRACT field extraction types to create
search-time field extractions.
```

NOTE: Index-time field extractions have performance implications.
      Create additions to the default set of indexed fields ONLY
      in specific circumstances. Whenever possible, extract
      fields only at search time.

There are times when you may find that you need to change or add to your set
of indexed fields. For example, you may have situations where certain
search-time field extractions are noticeably impacting search performance.
This can happen when the value of a search-time extracted field exists
outside of the field more often than not. For example, if you commonly
search a large event set with the expression company_id=1 but the value 1
occurs in many events that do *not* have company_id=1, you may want to add
company_id to the list of fields extracted by Splunk software at index time.
This is because at search time, Splunk software checks each
instance of the value 1 to see if it matches company_id, and that kind of
thing slows down performance when you have Splunk searching a large set of
data.

Conversely, if you commonly search a large event set with expressions like
company_id!=1 or NOT company_id=1, and the field company_id nearly *always*
takes on the value 1, you may want to add company_id to the list of fields
extracted by Splunk software at index time.

For more information about index-time field extraction, search the
documentation for "index-time extraction." For more information about
search-time field extraction, search the documentation for
"search-time extraction."

Field extraction configuration: field transforms vs. "inline" (props.conf only) configs

The TRANSFORMS and REPORT field extraction types reference an additional
component called a field transform, which you define separately in
transforms.conf. Field transforms contain a field-extracting regular
expression and other settings that govern the way that the transform
extracts fields. Field transforms are always created in conjunction with
field extraction stanzas in props.conf; they do not stand alone.

The EXTRACT field extraction type is considered to be "inline," which means
that it does not reference a field transform. It contains the regular
expression that Splunk software uses to extract fields at search time. You
can use EXTRACT to define a field extraction entirely within props.conf, no
transforms.conf component is required.

Search-time field extractions: Why use REPORT if EXTRACT will do?

This is a good question. And much of the time, EXTRACT is all you need for
search-time field extraction. But when you build search-time field
extractions, there are specific cases that require the use of REPORT and the
field transform that it references. Use REPORT if you want to:

* Reuse the same field-extracting regular expression across multiple
  sources, source types, or hosts. If you find yourself using the same regex
  to extract fields across several different sources, source types, and
  hosts, set it up as a transform, and then reference it in REPORT
  extractions in those stanzas. If you need to update the regex you only
  have to do it in one place. Handy!
* Apply more than one field-extracting regular expression to the same
  source, source type, or host. This can be necessary in cases where the
  field or fields that you want to extract from a particular source, source
  type, or host appear in two or more very different event patterns.
* Set up delimiter-based field extractions. Useful if your event data

```
  presents field-value pairs (or just field values) separated by delimiters
  such as commas, spaces, bars, and so on.
* Configure extractions for multivalued fields. You can have Splunk software
  append additional values to a field as it finds them in the event data.
* Extract fields with names beginning with numbers or underscores.
  Ordinarily, the key cleaning functionality removes leading numeric
  characters and underscores from field names. If you need to keep them,
  configure your field transform to turn key cleaning off.
* Manage formatting of extracted fields, in cases where you are extracting
  multiple fields, or are extracting both the field name and field value.

**Precedence rules for TRANSFORMS, REPORT, and EXTRACT field extraction types**

* For each field extraction, Splunk software takes the configuration from the
  highest precedence configuration stanza (see precedence rules at the
  beginning of this file).
* If a particular field extraction is specified for a source and a source
  type, the field extraction for source wins out.
* Similarly, if a particular field extraction is specified in ../local/ for
  a <spec>, it overrides that field extraction in ../default/.


TRANSFORMS-<class> = <transform_stanza_name>, <transform_stanza_name2>,...
* Used for creating indexed fields (index-time field extractions).
* <class> is a unique literal string that identifies the namespace of the
  field you're extracting.
  **Note:** <class> values do not have to follow field name syntax
  restrictions. You can use characters other than a-z, A-Z, and 0-9, and
  spaces are allowed. <class> values are not subject to key cleaning.
* <transform_stanza_name> is the name of your stanza from transforms.conf.
* Use a comma-separated list to apply multiple transform stanzas to a single
  TRANSFORMS extraction. Splunk software applies them in the list order. For
  example, this sequence ensures that the [yellow] transform stanza gets
  applied first, then [blue], and then [red]:
        [source::color_logs]
        TRANSFORMS-colorchange = yellow, blue, red
* See the RULESET-<class> setting for additional index-time transformation options.

RULESET-<class> = <string>
* This setting is used to perform index-time transformations, such as filtering,
  routing, and masking.
* A <class> is a unique string that identifies the name of a ruleset.
* Supply one or more transform stanza names as values for this setting. A
  transform stanza name is the name of a stanza in the transforms.conf
  file where you define your data transformations.
* Use a comma-separated list to apply multiple transform stanzas to a single
  RULESET extraction. Each transform stanza is applied in the order defined in the list.
* Use the REST endpoint: /services/data/ingest/rulesets to configure this setting.
* This setting is nearly identical to the TRANSFORMS-<class> setting, with the
  following exceptions:
  * If a RULESET is configured for a particular data stream on both indexers and
    heavy forwarders, the processor processes the RULESET on both Splunk platform
    instances. This is different from TRANSFORMS, which the processor ignores on an
    indexer if it has already been processed on the heavy forwarder.
  * If a data source matches both a TRANSFORMS and a RULESET, the processor applies
    the TRANSFORMS setting before the RULESET setting.
* Default: empty string

RULESET_DESC-<class> = <string>
* Description of 'RULESET-' to help users understand what a specific
  'RULESET-' index-time field extraction does.
  For example:
```

```
  RULESET_DESC-drop_debug_logs = Describes the 'RULESET-drop_debug_logs' field transform.

REPORT-<class> = <transform_stanza_name>, <transform_stanza_name2>,...
* Used for creating extracted fields (search-time field extractions) that
  reference one or more transforms.conf stanzas.
* <class> is a unique literal string that identifies the namespace of the
  field you're extracting.
  NOTE: <class> values do not have to follow field name syntax
  restrictions. You can use characters other than a-z, A-Z, and 0-9, and
  spaces are allowed. <class> values are not subject to key cleaning.
* <transform_stanza_name> is the name of your stanza from transforms.conf.
* Use a comma-separated list to apply multiple transform stanzas to a single
  REPORT extraction.
  Splunk software applies them in the list order. For example, this sequence
  insures that the [yellow] transform stanza gets applied first, then [blue],
  and then [red]:
    [source::color_logs]
    REPORT-colorchange = yellow, blue, red

EXTRACT-<class> = [<regex>|<regex> in <src_field>]
* Used to create extracted fields (search-time field extractions) that do
  not reference transforms.conf stanzas.
* Performs a regex-based field extraction from the value of the source
  field.
* <class> is a unique literal string that identifies the namespace of the
  field you're extracting.
  NOTE: <class> values do not have to follow field name syntax
  restrictions. You can use characters other than a-z, A-Z, and 0-9, and
  spaces are allowed. <class> values are not subject to key cleaning.
* The <regex> is required to have named capturing groups. When the <regex>
  matches, the named capturing groups and their values are added to the
  event.
* dotall (?s) and multi-line (?m) modifiers are added in front of the regex.
  So internally, the regex becomes (?ms)<regex>.
* Use '<regex> in <src_field>' to match the regex against the values of a
  specific field.  Otherwise it just matches against _raw (all raw event
  data).
* NOTE: <src_field> has the following restrictions:
  * It can only contain alphanumeric characters and underscore
    (a-z, A-Z, 0-9, and _).
  * It must already exist as a field that has either been extracted at
    index time or has been derived from an EXTRACT-<class> configuration
    whose <class> ASCII value is *lower* than the configuration in which
    you are attempting to extract the field. For example, if you
    have an EXTRACT-ZZZ configuration that extracts <src_field>, then
    you can only use 'in <src_field>' in an EXTRACT configuration with
    a <class> of 'aaa' or higher, as 'aaa' is higher in ASCII value
    than 'ZZZ'.
  * It cannot be a field that has been derived from a transform field
    extraction (REPORT-<class>), an automatic key-value field extraction
    (in which you configure the KV_MODE setting to be something other
    than 'none'), a field alias, a calculated field, or a lookup,
    as these operations occur after inline field extractions (EXTRACT-
    <class>) in the search-time operations sequence.
* If your regex needs to end with 'in <string>' where <string> is *not* a
  field name, change the regex to end with '[i]n <string>' to ensure that
  Splunk software doesn't try to match <string> to a field name.

KV_MODE = [none|auto|auto_escaped|multi|multi:<multikv.conf_stanza_name>json|xml]
* Used for search-time field extractions only.
* Specifies the field/value extraction mode for the data.
* Set KV_MODE to one of the following:
```

```
  * none - Disables field extraction for the host, source, or source type.
  * auto_escaped - Extracts fields/value pairs separated by equal signs and
                   honors \" and \\ as escaped sequences within quoted
                   values. For example: field="value with \"nested\" quotes"
  * multi - Invokes the 'multikv' search command, which extracts fields from
            table-formatted events.
  * multi:<multikv.conf_stanza_name> - Invokes a custom multikv.conf
    configuration to extract fields from a specific type of table-formatted
    event. Use this option in situations where the default behavior of the
    'multikv' search command is not meeting your needs.
  * xml - Automatically extracts fields from XML data.
  * json - Automatically extracts fields from JSON data.
* Setting to 'none' can ensure that one or more custom field extractions are not
  overridden by automatic field/value extraction for a particular host,
  source, or source type. You can also use 'none' to increase search
  performance by disabling extraction for common but nonessential fields.
* The 'xml' and 'json' modes do not extract any fields when used on data
  that isn't of the correct format (JSON or XML).
* If you set 'KV_MODE = json' for a source type, do not also set
  'INDEXED_EXTRACTIONS = JSON' for the same source type. This causes the Splunk
  software to extract the json fields twice: once at index time and again at
  search time.
* When KV_MODE is set to 'auto' or 'auto_escaped', automatic JSON field
  extraction can take place alongside other automatic field/value extractions.
  To disable JSON field extraction when 'KV_MODE' is set to 'auto' or
  'auto_escaped', add 'AUTO_KV_JSON = false' to the stanza.
* Default: auto

MATCH_LIMIT = <integer>
* Only set in props.conf for EXTRACT type field extractions.
  For REPORT and TRANSFORMS field extractions, set this in transforms.conf.
* Optional. Limits the amount of resources spent by PCRE
  when running patterns that do not match.
* Use this to set an upper bound on how many times PCRE calls an internal
  function, match(). If set too low, PCRE may fail to correctly match a pattern.
* Default: 100000

DEPTH_LIMIT = <integer>
* Only set in props.conf for EXTRACT type field extractions.
  For REPORT and TRANSFORMS field extractions, set this in transforms.conf.
* Optional. Limits the amount of resources spent by PCRE
  when running patterns that do not match.
* Use this to limit the depth of nested backtracking in an internal PCRE
  function, match(). If set too low, PCRE might fail to correctly
  match a pattern.
* Default: 1000

AUTO_KV_JSON = <boolean>
* Used only for search-time field extractions.
* Specifies whether to extract fields from JSON data when 'KV_MODE' is set to
  'auto' or 'auto_escaped'.
* To disable automatic extraction of fields from JSON data when 'KV_MODE' is
  set to 'auto' or 'auto_escaped', set 'AUTO_KV_JSON = false'.
* Setting 'AUTO_KV_JSON = false' when 'KV_MODE' is set to 'none', 'multi',
  'json', or 'xml' has no effect.
* Default: true

KV_TRIM_SPACES = <boolean>
* Modifies the behavior of KV_MODE when set to auto, and auto_escaped.
* Traditionally, automatically identified fields have leading and trailing
  whitespace removed from their values.
  * Example event: 2014-04-04 10:10:45 myfield=" apples "
```

    would result in a field called 'myfield' with a value of 'apples'.
* If this value is set to false, then external whitespace then this outer
  space is retained.
  * Example: 2014-04-04 10:10:45 myfield=" apples "
    would result in a field called 'myfield' with a value of ' apples '.
* The trimming logic applies only to space characters, not tabs, or other
  whitespace.
* NOTE: Splunk Web currently has limitations with displaying and
  interactively clicking on fields that have leading or trailing
  whitespace. Field values with leading or trailing spaces may not look
  distinct in the event viewer, and clicking on a field value typically
  inserts the term into the search string without its embedded spaces.
  * The limitations are not specific to this feature. Any embedded spaces
    behave this way.
  * The Splunk search language and included commands respect the spaces.
* Default: true

CHECK_FOR_HEADER = <boolean>
* Used for index-time field extractions only.
* Set to true to enable header-based field extraction for a file.
* If the file has a list of columns and each event contains a field value
  (without field name), Splunk software picks a suitable header line to
  use for extracting field names.
* Can only be used on the basis of [<sourcetype>] or [source::<spec>],
  not [host::<spec>].
* Disabled when LEARN_SOURCETYPE = false.
* Causes the indexed source type to have an appended numeral; for
  example, sourcetype-2, sourcetype-3, and so on.
* The field names are stored in etc/apps/learned/local/props.conf.
  * Because of this, this feature does not work in most environments where
    the data is forwarded.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: false

SEDCMD-<class> = <sed script>
* Only used at index time.
* Commonly used to anonymize incoming data at index time, such as credit
  card or social security numbers. For more information, search the online
  documentation for "anonymize data."
* Used to specify a sed script which Splunk software applies to the _raw
  field.
* A sed script is a space-separated list of sed commands. Currently the
  following subset of sed commands is supported:
    * replace (s) and character substitution (y).
* Syntax:
    * replace - s/regex/replacement/flags
      * regex is a perl regular expression (optionally containing capturing
        groups).
      * replacement is a string to replace the regex match. Use \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.
    * substitute - y/string1/string2/
      * substitutes the string1[i] with string2[i]
* No default.

FIELDALIAS-<class> = (<orig_field_name> AS|ASNEW <new_field_name>)+
* Use FIELDALIAS configurations to apply aliases to a field. This lets you
  search for the original field using one or more alias field names. For
  example, a search expression of <new_field_name>=<value> also

```
  finds events that match <orig_field_name>=<value>.
* <orig_field_name> is the original name of the field. It is not removed by
  this configuration.
* <new_field_name> is the alias to assign to the <orig_field_name>.
* You can create multiple aliases for the same field. For example, a single
  <orig_field_name> may have multiple <new_field_name>s as long as all of
  the <new_field_name>s are distinct.
  * Example of a valid configuration:
    FIELDALIAS-vendor = vendor_identifier AS vendor_id    \
                        vendor_identifier AS vendor_name
* You can include multiple field alias renames in the same stanza.
* Avoid applying the same alias field name to multiple original field
  names as a single alias cannot refer to multiple original source fields.
  Each alias can map to only one source field. If you attempt to create
  two field aliases that map two separate <orig_field_name>s onto the
  same <new_field_name>, only one of the aliases takes effect, not both.
  * For example, if you attempt to run the following configuration,
    which maps two <orig_field_name>s to the same <new_field_name>, only
    one of the aliases takes effect, not both. The following definition
    demonstrates an invalid configuration:
    FIELDALIAS-foo = userID AS user loginID AS user
  * If you must do this, set it up as a calculated field (an EVAL-* statement)
    that uses the 'coalesce' function to create a new field that takes the
    value of one or more existing fields. This method lets you be explicit
    about ordering of input field values in the case of NULL fields. For
    example: EVAL-ip = coalesce(clientip,ipaddress)
* The following is true if you use AS in this configuration:
  * If the alias field name <new_field_name> already exists, the Splunk
    software replaces its value with the value of <orig_field_name>.
  * If the <orig_field_name> field has no value or does not exist, the
    <new_field_name> is removed.
* The following is true if you use ASNEW in this configuration:
  * If the alias field name <new_field_name> already exists, the Splunk
    software does not change it.
  * If the <orig_field_name> field has no value or does not exist, the
    <new_field_name> is kept.
* Field aliasing is performed at search time, after field extraction, but
  before calculated fields (EVAL-* statements) and lookups.
  This means that:
  * Any field extracted at search time can be aliased.
  * You can specify a lookup based on a field alias.
  * You cannot alias a calculated field.
* No default.

EVAL-<fieldname> = <eval statement>
* Use this to automatically run the <eval statement> and assign the value of
  the output to <fieldname>. This creates a "calculated field."
* When multiple EVAL-* statements are specified, they behave as if they are
* run in parallel, rather than in any particular sequence.
  For example say you have two statements: EVAL-x = y*2 and EVAL-y=100. In
  this case, "x" is assigned the original value of "y * 2," not the
  value of "y" after it is set to 100.
* Splunk software processes calculated fields after field extraction and
  field aliasing but before lookups. This means that:
  * You can use a field alias in the eval statement for a calculated
    field.
  * You cannot use a field added through a lookup in an eval statement for a
    calculated field.
* No default.

LOOKUP-<class> = $TRANSFORM (<match_field> (AS <match_field_in_event>)?)+ (OUTPUT|OUTPUTNEW (<output_field>
(AS <output_field_in_event>)? )+ )?
```

```
* At search time, identifies a specific lookup table and describes how that
  lookup table should be applied to events.
* <match_field> specifies a field in the lookup table to match on.
  * By default Splunk software looks for a field with that same name in the
    event to match with (if <match_field_in_event> is not provided)
  * You must provide at least one match field. Multiple match fields are
    allowed.
* <output_field> specifies a field in the lookup entry to copy into each
  matching event in the field <output_field_in_event>.
  * If you do not specify an <output_field_in_event> value, Splunk software
    uses <output_field>.
  * A list of output fields is not required.
* If they are not provided, all fields in the lookup table except for the
  match fields (and the timestamp field if it is specified) are output
  for each matching event.
* If the output field list starts with the keyword "OUTPUTNEW" instead of
  "OUTPUT", then each output field is only written out if it did not previous
  exist. Otherwise, the output fields are always overridden. Any event that
  has all of the <match_field> values but no matching entry in the lookup
  table clears all of the output fields.  NOTE that OUTPUTNEW behavior has
  changed since 4.1.x (where *none* of the output fields were written to if
  *any* of the output fields previously existed).
* Splunk software processes lookups after it processes field extractions,
  field aliases, and calculated fields (EVAL-* statements). This means that you
  can use extracted fields, aliased fields, and calculated fields to specify
  lookups. But you can't use fields discovered by lookups in the
  configurations of extracted fields, aliased fields, or calculated fields.
* The LOOKUP- prefix is actually case-insensitive. Acceptable variants include:
   LOOKUP_<class> = [...]
   LOOKUP<class>  = [...]
   lookup_<class> = [...]
   lookup<class>  = [...]
* No default.
```

Binary file configuration

```
NO_BINARY_CHECK = <boolean>
* When set to true, Splunk software processes binary files.
* Can only be used on the basis of [<sourcetype>], or [source::<source>],
  not [host::<host>].
* Default: false (binary files are ignored).
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.

detect_trailing_nulls = [auto|true|false]
* When enabled, Splunk software tries to avoid reading in null bytes at
  the end of a file.
* When false, Splunk software assumes that all the bytes in the file should
  be read and indexed.
* Set this value to false for UTF-16 and other encodings (CHARSET) values
  that can have null bytes as part of the character text.
* Subtleties of 'true' vs 'auto':
  * 'true' is the historical behavior of trimming all null
            bytes when Splunk software runs on Windows.
  * 'auto' is currently a synonym for true but may be extended to be
            sensitive to the charset selected (i.e. quantized for multi-byte
            encodings, and disabled for unsafe variable-width encodings)
* This feature was introduced to work around programs which foolishly
```

```
  preallocate their log files with nulls and fill in data later.  The
  well-known case is Internet Information Server.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default (on *nix machines): false
* Default (on Windows machines): true
```

Segmentation configuration

```
SEGMENTATION = <segmenter>
* Specifies the segmenter from segmenters.conf to use at index time for the
  host, source, or sourcetype specified by <spec> in the stanza heading.
* Default: indexing

SEGMENTATION-<segment selection> = <segmenter>
* Specifies that Splunk Web should use the specific segmenter (from
  segmenters.conf) for the given <segment selection> choice.
* Default <segment selection> choices are: all, inner, outer, raw. For more
  information see the Admin Manual.
* Do not change the set of default <segment selection> choices, unless you
  have some overriding reason for doing so. In order for a changed set of
  <segment selection> choices to appear in Splunk Web, you need to edit
  the Splunk Web UI.
```

File checksum configuration

```
CHECK_METHOD = [endpoint_md5|entire_md5|modtime]
* Set CHECK_METHOD to "endpoint_md5" to have Splunk software perform a checksum
  of the first and last 256 bytes of a file. When it finds matches, Splunk
  software lists the file as already indexed and indexes only new data, or
  ignores it if there is no new data.
* Set CHECK_METHOD to "entire_md5" to use the checksum of the entire file.
* Set CHECK_METHOD to "modtime" to check only the modification time of the
  file.
* Settings other than "endpoint_md5" cause Splunk software to index the entire
  file for each detected change.
* This option is only valid for [source::<source>] stanzas.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: endpoint_md5

initCrcLength = <integer>
* See documentation in inputs.conf.spec.
```

Small file settings

```
PREFIX_SOURCETYPE = <boolean>
* NOTE: this setting is only relevant to the "[too_small]" sourcetype.
* Determines the source types that are given to files smaller than 100
```

```
  lines, and are therefore not classifiable.
* PREFIX_SOURCETYPE = false sets the source type to "too_small."
* PREFIX_SOURCETYPE = true sets the source type to "<sourcename>-too_small",
  where "<sourcename>" is a cleaned up version of the filename.
  * The advantage of PREFIX_SOURCETYPE = true is that not all small files
    are classified as the same source type, and wildcard searching is often
    effective.
  * For example, a Splunk search of "sourcetype=access*" retrieves
    "access" files as well as "access-too_small" files.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: true
```

Sourcetype configuration

```
sourcetype = <string>
* Can only be set for a [source::...] stanza.
* Anything from that <source> is assigned the specified source type.
* Is used by file-based inputs, at input time (when accessing logfiles) such
  as on a forwarder, or indexer monitoring local files.
* sourcetype assignment settings on a system receiving forwarded Splunk data
  are not be applied to forwarded data.
* For log files read locally, data from log files matching <source> is
  assigned the specified source type.
* Default: empty string

# The following setting/value pairs can only be set for a stanza that
# begins with [<sourcetype>]:

rename = <string>
* Renames [<sourcetype>] as <string> at search time
* With renaming, you can search for the [<sourcetype>] with
  sourcetype=<string>
* To search for the original source type without renaming it, use the
  field _sourcetype.
* Data from a renamed sourcetype only uses the search-time
  configuration for the target sourcetype. Field extractions
  (REPORTS/EXTRACT) for this stanza sourcetype are ignored.
* Default: empty string

invalid_cause = <string>
* Can only be set for a [<sourcetype>] stanza.
* If invalid_cause is set, the Tailing code (which handles uncompressed
  logfiles) does not read the data, but hands it off to other components or
  throws an error.
* Set <string> to "archive" to send the file to the archive processor
  (specified in unarchive_cmd).
* When set to "winevt", this causes the file to be handed off to the
  Event Log input processor.
* Set to any other string to throw an error in the splunkd.log if you are
  running Splunklogger in debug mode.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: empty string

is_valid = <boolean>
```

```
* Automatically set by invalid_cause.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* DO NOT SET THIS.
* Default: true

force_local_processing = <boolean>
* Forces a universal forwarder to process all data tagged with this sourcetype
  locally before forwarding it to the indexers.
* Data with this sourcetype is processed by the linebreaker,
  aggerator, and the regexreplacement processors in addition to the existing
  utf8 processor.
* Note that switching this property potentially increases the cpu
  and memory consumption of the forwarder.
* Applicable only on a universal forwarder.
* Default: false

unarchive_cmd = <string>
* Only called if invalid_cause is set to "archive".
* This field is only valid on [source::<source>] stanzas.
* <string> specifies the shell command to run to extract an archived source.
* Must be a shell command that takes input on stdin and produces output on
  stdout.
* Use _auto for Splunk software's automatic handling of archive files (tar,
  tar.gz, tgz, tbz, tbz2, zip)
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: empty string

unarchive_cmd_start_mode = [direct|shell]
* Determines how the Splunk platform runs the "unarchive_cmd" command.
* A value of "direct" means that the Splunk daemon runs the 'unarchive_cmd' command
  directly, and does not use a command shell. In this case, the Splunk daemon attempts
  to run the first 'unarchive_cmd' value that you specify as a command. Any subsequent
  values in the 'unarchive_cmd' are interpreted as the command's arguments.
  * When this setting has a value of "direct", command shell operators such
    as '&&' or ';' in the "unarchive_cmd" value cannot be used and will cause
    unexpected results. If you need to run multiple commands consecutively or
    conditionally using command shell syntax, give this setting a value of
    "shell" instead.
* A value of "shell" means that a shell process runs the "unarchive_cmd" commands.
  This allows for execution of a command pipeline that consists of multiple commands.
* Default: shell

unarchive_sourcetype = <string>
* Sets the source type of the contents of the matching archive file. Use
  this field instead of the sourcetype field to set the source type of
  archive files that have the following extensions: gz, bz, bz2, Z.
* If this field is empty (for a matching archive file props lookup) Splunk
  software strips off the archive file's extension (.gz, bz etc) and lookup
  another stanza to attempt to determine the sourcetype.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: empty string

LEARN_SOURCETYPE = <boolean>
* Determines whether learning of known or unknown sourcetypes is enabled.
  * For known sourcetypes, refer to LEARN_MODEL.
  * For unknown sourcetypes, refer to the rule:: and delayedrule::
```

```
    configuration (see below).
* Setting this field to false disables CHECK_FOR_HEADER as well (see above).
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: true

LEARN_MODEL = <boolean>
* For known source types, the file classifier adds a model file to the
  learned directory.
* To disable this behavior for diverse source types (such as source code,
  where there is no good example to make a sourcetype) set LEARN_MODEL =
  false.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: true

termFrequencyWeightedDist = <boolean>
* Whether or not the Splunk platform calculates distance between files by
  using the frequency at which unique terms appear in those files.
* The Splunk platform calculates file "distance", or how similar one file
  is to another, by analyzing patterns that it finds within each file.
* When this setting is the default of "false", the platform determines the
  file distance by using the number of unique terms that each file shares
  with another. This is the legacy behavior.
* To instead have the platform use the frequency in which those terms occur
  within a file to determine its distance from another file, set this to
  "true". This is a more accurate representation of file distance.
* Default: false

maxDist = <integer>
* Determines how different a source type model may be from the current file.
* The larger the 'maxDist' value, the more forgiving Splunk software is
  with differences.
  * For example, if the value is very small (for example, 10), then files
    of the specified sourcetype should not vary much.
  * A larger value indicates that files of the given source type can vary
    quite a bit.
* If you're finding that a source type model is matching too broadly, reduce
  its 'maxDist' value by about 100 and try again. If you're finding that a
  source type model is being too restrictive, increase its 'maxDist 'value by
  about 100 and try again.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: 300

# rule:: and delayedrule:: configuration

MORE_THAN<optional_unique_value>_<number> = <regular expression> (empty)
LESS_THAN<optional_unique_value>_<number> = <regular expression> (empty)

* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.

An example:

    [rule::bar_some]
    sourcetype = source_with_lots_of_bars
    # if more than 80% of lines have "----", but fewer than 70% have "####"
```

```
    # declare this a "source_with_lots_of_bars"
    MORE_THAN_80 = ----
    LESS_THAN_70 = ####

A rule can have many MORE_THAN and LESS_THAN patterns, and all are required
for the rule to match.
```

Annotation Processor configured

```
ANNOTATE_PUNCT = <boolean>
* Determines whether to index a special token starting with "punct::"
  * The "punct::" key contains punctuation in the text of the event.
    It can be useful for finding similar events
  * If it is not useful for your dataset, or if it ends up taking
    too much space in your index it is safe to disable it
* Default: true
```

Header Processor configuration

```
HEADER_MODE = <empty> | always | firstline | none
* Determines whether to use the inline ***SPLUNK*** directive to rewrite
  index-time fields.
  * If "always", any line with ***SPLUNK*** can be used to rewrite
    index-time fields.
  * If "firstline", only the first line can be used to rewrite
    index-time fields.
  * If "none", the string ***SPLUNK*** is treated as normal data.
  * If <empty>, scripted inputs take the value "always" and file inputs
    take the value "none".
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: <empty>
```

Internal settings

```
# NOT YOURS. DO NOT SET.

_actions = <string>
* Internal field used for user-interface control of objects.
* Default: "new,edit,delete".

pulldown_type = <boolean>
* Internal field used for user-interface control of source types.
* Default: empty

given_type = <string>
* Internal field used by the CHECK_FOR_HEADER feature to remember the
  original sourcetype.
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
```

```
  data.
* Default: not set
```

Sourcetype Category and Descriptions

```
description = <string>
* Field used to describe the sourcetype. Does not affect indexing behavior.
* Default: not set

category = <string>
* Field used to classify sourcetypes for organization in the front end. Case
  sensitive. Does not affect indexing behavior.
* Default: not set
```

props.conf.example

```
#   Version 9.2.2
#
# The following are example props.conf configurations. Configure properties for
# your data.
#
# To use one or more of these configurations, copy the configuration block into
# props.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


########
# Line merging settings
########

# The following example line-merges source data into multi-line events for
# apache_error sourcetype.

[apache_error]
SHOULD_LINEMERGE = True



########
# Settings for tuning
########

# The following example limits the amount of characters indexed per event from
# host::small_events.

[host::small_events]
TRUNCATE = 256

# The following example turns off DATETIME_CONFIG (which can speed up indexing)
# from any path that ends in /mylogs/*.log.
#
# In addition, the default splunk behavior of finding event boundaries
```

```
# via per-event timestamps can't work with NONE, so we disable
# SHOULD_LINEMERGE, essentially declaring that all events in this file are
# single-line.

[source::.../mylogs/*.log]
DATETIME_CONFIG = NONE
SHOULD_LINEMERGE = false



########
# Timestamp extraction configuration
########

# The following example sets Eastern Time Zone if host matches nyc*.

[host::nyc*]
TZ = US/Eastern


# The following example uses a custom datetime.xml that has been created and
# placed in a custom app directory. This sets all events coming in from hosts
# starting with dharma to use this custom file.

[host::dharma*]
DATETIME_CONFIG = <etc/apps/custom_time/datetime.xml>

########
## Timezone alias configuration
########

# The following example uses a custom alias to disambiguate the Australian
# meanings of EST/EDT

TZ_ALIAS = EST=GMT+10:00,EDT=GMT+11:00

# The following example gives a sample case wherein, one timezone field is
# being replaced by/interpreted as another.

TZ_ALIAS = EST=AEST,EDT=AEDT

########
# Transform configuration
########

# The following example creates a search field for host::foo if tied to a
# stanza in transforms.conf.

[host::foo]
TRANSFORMS-foo=foobar

# The following stanza extracts an ip address from _raw
[my_sourcetype]
EXTRACT-extract_ip = (?<ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})

# The following example shows how to configure lookup tables
[my_lookuptype]
LOOKUP-foo = mylookuptable userid AS myuserid OUTPUT username AS myusername

# The following shows how to specify field aliases
FIELDALIAS-foo = user AS myuser id AS myid
```

```
########
# Sourcetype configuration
########

# The following example sets a sourcetype for the file web_access.log for a
# unix path.

[source::.../web_access.log]
sourcetype = splunk_web_access

# The following example sets a sourcetype for the Windows file iis6.log.  Note:
# Backslashes within Windows file paths must be escaped.

[source::...\\iis\\iis6.log]
sourcetype = iis_access

# The following example extracts data from a .Z archive

[preprocess-Z]
invalid_cause = archive
is_valid = False
LEARN_MODEL = false

[source::....Z(.\d+)?]
unarchive_cmd = gzip -cd -
sourcetype = preprocess-Z
NO_BINARY_CHECK = true

# The following example learns a custom sourcetype and limits the range between
# different examples with a smaller than default maxDist.

[custom_sourcetype]
LEARN_MODEL = true
maxDist = 30


# rule:: and delayedrule:: configuration
# The following examples create sourcetype rules for custom sourcetypes with
# regex.


[rule::bar_some]
sourcetype = source_with_lots_of_bars
MORE_THAN_80 = ----


[delayedrule::baz_some]
sourcetype = my_sourcetype
LESS_THAN_70 = ####


########
# File configuration
########

# Binary file configuration
# The following example eats binary files from the sourcetype
# "imported_records".

[imported_records]
NO_BINARY_CHECK = true
```

```
# File checksum configuration
# The following example checks the entirety of every file in the web_access
# directory rather than skipping files that appear to be the same.

[source::.../web_access/*]
CHECK_METHOD = entire_md5

########
# Metric configuration
########

# A metric sourcetype of type statsd with 'regex_stanza1', 'regex_stanza2' to
# extract dimensions
[metric_sourcetype_name]
METRICS_PROTOCOL = statsd
STATSD-DIM-TRANSFORMS = regex_stanza1, regex_stanza2

#Convert a single log event into multiple metrics using METRIC-SCHEMA-TRANSFORMS
#and index time extraction feature.
[logtometrics]
METRIC-SCHEMA-TRANSFORMS = metric-schema:logtometrics
TRANSFORMS-group = extract_group
TRANSFORMS-name = extract_name
TRANSFORMS-max_size_kb = extract_max_size_kb
TRANSFORMS-current_size_kb = extract_current_size_kb
TRANSFORMS-current_size = extract_current_size
TRANSFORMS-largest_size = extract_largest_size
TRANSFORMS-smallest_size = extract_smallest_size
category = metrics
should_linemerge = false
```

pubsub.conf

The following are the spec and example files for `pubsub.conf`.

pubsub.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attributes and values for configuring a client of
# the PubSub system (broker).
#
# To set custom configurations, place a pubsub.conf in
# $SPLUNK_HOME/etc/system/local/.
# For examples, see pubsub.conf.example. You must restart Splunk to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
#     file wins.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

#*******************************************************************
# Configure the physical location where deploymentServer is running.
# This configuration is used by the clients of the pubsub system.
#*******************************************************************
```

[pubsub-server:deploymentServer]

```
disabled = <boolean>
* Default: false

targetUri = <IP:Port>|<hostname:Port>|direct
* Specify either the url of a remote server in case the broker is remote, or
  just the keyword "direct" when broker is in-process.
* It is usually a good idea to co-locate the broker and the Deployment Server
  on the same Splunk. In such a configuration, all
* deployment clients would have targetUri set to deploymentServer:port.

#*******************************************************************
# The following section is only relevant to Splunk developers.
#*******************************************************************

# This "direct" configuration is always available, and cannot be overridden.
```

[pubsub-server:direct]

```
disabled = false
targetUri = direct
```

[pubsub-server:<logicalName>]

```
* It is possible for any Splunk to be a broker. If you have multiple brokers,
  assign a logicalName that is used by the clients to refer to it.

disabled = <false or true>
* Default: false

targetUri = <IP:Port>|<hostname:Port>|direct
* The URI of a Splunk that is being used as a broker.
* The keyword "direct" implies that the client is running on the same Splunk
  instance as the broker.
```

pubsub.conf.example

```
#   Version 9.2.2

[pubsub-server:deploymentServer]
disabled=false
targetUri=somehost:8089

[pubsub-server:internalbroker]
disabled=false
targetUri=direct
```

restmap.conf

The following are the spec and example files for `restmap.conf`.

restmap.conf.spec

```
# Version 9.2.2
#
# This file contains possible attribute/value pairs for creating new
# Representational State Transfer (REST) endpoints.

# There is a restmap.conf in $SPLUNK_HOME/etc/system/default/. To set custom
# configurations, place a restmap.conf in $SPLUNK_HOME/etc/system/local/. For
# examples, see restmap.conf.example. You must restart Splunk software to
# enable configurations.
#
# To learn more about configuration files (including precedence), see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles.
#
# NOTE: You must register every REST endpoint using this file to make it available.
```

GLOBAL SETTINGS

```
# Use the [global] stanza to define any global settings.
#   * You can also define global settings outside of any stanza at the top
#     of the file.
#   * Each .conf file should have at most one global stanza. If there are
#     multiple global stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in
#     the file takes precedence.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

[global]

allowGetAuth = <boolean>
* Allows the username/password to be passed as a GET parameter to endpoint
  services/authorization/login.
* Setting to "true" might result in your username and password being
  logged as cleartext in Splunk logs and any proxy servers in between.
```

```
* Default: false

allowRestReplay = <boolean>
* Allows POST/PUT/DELETE requests to be replayed on other nodes in the deployment.
* Setting to "true" enables centralized management.
* You can also control replay at each endpoint level.
* CAUTION: This feature is currently internal. Do not enable it
  without consulting Splunk support.
* Default: false

defaultRestReplayStanza = <string>
* Points to the default or global REST replay configuration stanza.
* This setting is related to the 'allowRestReplay' setting.
* Default: restreplayshc

pythonHandlerPath = <path>
* Path to the 'main' python script handler.
* Used by the script handler to determine where the actual 'main' script is
  located.
* Typically you do not need to edit this setting.
* Default: $SPLUNK_HOME/bin/rest_handler.py

v1APIBlockGETSearchLaunch = <boolean>
* Triggers breaking changes in default and v1 variants of the endpoints:
* /search/jobs/export
* /search/jobs/{sid}/(events|results|results_preview)
* /search/jobs/oneshot
* /search/parser
* These changes involve removing the abilty to launch searches using
  HTTP GET requests.
* Default: false

[<rest endpoint name>:<endpoint description string>]
* Settings under this stanza are applicable to all REST stanzas.
* Settings in other stanzas might supply additional information.

match = <path>
* Specify the URI that calls the handler.
* For example, if match=/foo
  then https: //$SERVER:$PORT/services/foo
  calls this handler.
* NOTE: You must start your path with a "/".

requireAuthentication = <boolean>
* Determines if this endpoint requires authentication.
* (OPTIONAL)
* Default: true

authKeyStanza = <string>
* A list of comma or space separated stanza names that specifies the location
  of the pass4SymmKeys in the server.conf file to use for endpoint authentication.
* Tries to authenticate with all configured pass4SymmKeys.
* If no pass4SymmKey is available, authentication is done using the
  pass4SymmKey in the [general] stanza.
* This setting applies only if the 'requireAuthentication' setting is set to
  "true".
* (OPTIONAL) When not set, the endpoint will not be authenticated using
  pass4SymmKeys.
* Default: not set

restReplay = <boolean>
* Enables REST replay on this endpoint group.
```

```
* (OPTIONAL)
* Related to the 'allowRestReplay' setting.
* CAUTION: This feature is currently internal. Do not
  enable it without consulting Splunk support.
* Default: false

restReplayStanza = <string>
* This setting points to a stanza that can override the
  [global]/defaultRestReplayStanza value on a per-endpoint/regex basis.
* Default: empty string

capability = <capabilityName>
capability.<post|delete|get|put> = <capabilityName>
* Depending on the HTTP method, check capabilities on the authenticated session user.
* If you use the 'capability.<post|delete|get|put>' setting, the associated method is
  checked against the authenticated user's role.
* If you use the capability' setting, all calls are checked against this
  capability regardless of the HTTP method.
* You can also express capabilities as a boolean expression.
  Supported operators include: or, and, ()

acceptFrom = <comma-separated list>
* A list of networks or addresses from which to allow this endpoint to be accessed.
* Do not confuse this setting with the identical setting in the
  [httpServer] stanza of server.conf which controls whether a host can
  make HTTP requests at all.
* Each rule can be in the following forms:
    1. A single IPv4 or IPv6 address (examples: "10.1.2.3", "fe80::4a3")
    2. A CIDR block of addresses (examples: "10/8", "fe80:1234/32")
    3. A DNS name, possibly with a '*' used as a wildcard (examples:
       "myhost.example.com", "*.splunk.com")
    4. A single '*' which matches anything.
* You can also prefix entries with '!' to cause the rule to reject the
  connection. Rules are applied in order, and the first one to match is
  used. For example, "!10.1/16, *" allows connections from everywhere
  except the 10.1.*.* network.
* Default: "*" (accept from anywhere)

includeInAccessLog = <boolean>
* Whether to include requests to this endpoint in the splunkd_access.log.
* If set to "true", requests appear in splunkd_access.log.
* If set to "false", requests do not appear in splunkd_access.log.
* Default: true

[script:<uniqueName>]
* Per-endpoint stanza.
* Use this stanza to specify a handler and other handler-specific settings.
* The handler is responsible for implementing arbitrary namespace underneath
  each REST endpoint.
* NOTE: The uniqueName must be different for each handler.
* Call the specified handler when executing this endpoint.
* The attribute/value pairs below support the script handler.

scripttype = <string>
* Tells the system what type of script to run when using this endpoint.
* If set to "persist", it runs the script using a persistent process that
  uses the protocol from persistconn/appserver.py.
* Default: python

python.version={default|python|python2|python3}
* For Python scripts only, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
```

```
  version.
* (OPTIONAL)
* Default: Not set (Uses the system-wide Python version.)

handler=<SCRIPT>.<CLASSNAME>
* The name and class name of the file to execute.
* The file must be located in an application's bin subdirectory.
* For example, $SPLUNK_HOME/etc/apps/<APPNAME>/bin/TestHandler.py has a class
  called MyHandler (which, in the case of python must be derived from a base
  class called 'splunk.rest.BaseRestHandler'). The attribute/value pair for it is:
  "handler=TestHandler.MyHandler".

xsl = <string>
* The path to an XSL transform file.
* Perform an XSL transform on data returned from the handler.
* (OPTIONAL) Only use this setting if the data is in XML format.
* Does not apply if the 'scripttype' setting is set to "persist".

script = <string>
* The path to a script executable.
* (Optional). Use this setting only if the 'scripttype' setting is set to "python".
  This setting allows you to run a script which is *not* derived from
  'splunk.rest.BaseRestHandler'. This setting is rarely used.
* If the 'scripttype' setting is set to "persist", this setting is
  the path that is sent to the driver to run. In that case,
  environment variables are substituted.

script.arg.<N> = <string>
* A list of arguments that are passed to the driver to start the script.
* Only has effect if the 'scripttype' setting is set to "persist".
* The script can use this information however it wants.
* Environment variables are substituted.

script.param = <string>
* A free-form argument that is passed to the driver when it starts the script.
* (OPTIONAL)
* Only has effect if the 'scripttype' setting is set to "persist".
* The script can use this information however it wants.
* Environment variables are substituted.

output_modes = <comma-separated list>
* Specify which output formats this endpoint can request.
* Valid values: json, xml
* Default: xml

passSystemAuth = <boolean>
* Specifies whether or not to pass in a system-level
  authentication token on each request.
* Default: false

driver = <path>
* If the 'scripttype' setting is set to "persist", specifies
  the command to start a persistent server for this process.
* Endpoints that share the same driver configuration can share processes.
* Environment variables are substituted.
* Default: the persistconn/appserver.py server

driver.arg.<n> = <string>
* If the 'scripttype' setting is set to "persist", specifies
  the command to start a persistent server for this process.
* Environment variables are substituted.
* Only takes effect when "driver" is specifically set.
```

```
driver.env.<name> = <string>
* If the 'scripttype' setting is set to "persist", specifies
  an environment variable to set when running the driver process.

passConf = <boolean>
* If set, the script is sent the contents of this
  configuration stanza as part of the request.
* Only has effect if the 'scripttype' setting is set to "persist".
* Default: true

passPayload = [true|false|base64]
* If set to "true", sends the driver the raw, unparsed body of the
  POST/PUT as a "payload" string.
* If set to "base64", the same body is instead base64-encoded and
  sent as a "payload_base64" string.
* Only has effect if the 'scripttype' setting is set to "persist".
* Default: false

passSession = <boolean>
* If set to "true", sends the driver information about the user's
  session. This includes the user's name, an active authtoken,
  and other details.
* Only has effect if the 'scripttype' setting is set to "persist".
* Default: true

passHttpHeaders = <boolean>
* Determines whether splunkd passes HTTP request headers to the driver.
* A value of "true" means splunkd passes the HTTP request headers
  to the driver.
* Only has effect if the 'scripttype' setting is set to "persist".
* Default: false

passHttpCookies = <boolean>
* If set to "true", sends the driver the HTTP cookies of the request.
* Only has effect if the 'scripttype' setting is set to "persist".
* Default: false

stream = <boolean>
* Describes whether or not splunkd sends the payload in the
  request to the driver in a streaming fashion.
* A value of "true" means splunkd sends the payload in the
  request to the driver in a stream, or multiple sequential requests.
* A value of "false" means splunkd sends the payload in the
  request to the driver as a field of the original request.
* Only has effect if the 'scripttype' setting is set to "persist".
* Default: false

[admin:<uniqueName>]
* 'admin'
* The built-in handler for the Extensible Administration Interface (EAI).
* Exposes the listed EAI handlers at the given URL.

match = <string>
* A partial URL which, when accessed, displays the handlers listed below.

members = <comma-separated list>
* A list of handlers to expose at this URL.
* See https://localhost:8089/services/admin
  for a list of all possible handlers.

maxCacheTime = <interval>
```

```
* The maximum amount of time that the Splunk platform can cache a response for
  this REST endpoint.
* Determines how the endpoint handles the caching of HTTP responses.
  Specifically, controls the value of HTTP cache control headers, which is
  important for those who use a reverse proxy or an external client to access
  REST endpoints.
* You can specify the interval in seconds (s), minutes (m), hours (h), or days (d).
  For example: 60s, 1m, 1h, 1d; etc.
* Maximum accepted value is 1 year.
* Default: 0 (disabled)

capability = <string>
capability.<post|delete|get|put> = <string>

* One or more capabilities that an authenticated user must hold before they can
  execute an HTTP request against the REST endpoint URL that you specify in
  the stanza name.
* When a logged-in user submits an HTTP request to an endpoint, splunkd confirms
  that the user holds a minimum of the capabilities you specify in this setting
  before it lets the request act upon the endpoint. If the HTTP request is not submitted,
  splunkd rejects the attempt.
* This setting has two forms, which determine how capability checking occurs:
  * 'capability' on its own configures splunkd to confirm that the logged-in
     user holds the capabilities you specify to act upon the URL for any HTTP
     request method.
  * 'capability.<post|delete|get|put>' configures splunkd to confirm that the
    logged-in user holds the capabilities to act upon the URL through the HTTP
    method you specify after the period. You can only specify one method type
    after the period.
  * For example, if you specify "capability.get = admin_all_objects",
    splunkd confirms that the user holds the "admin_all_objects" capability before it
    lets them perform an HTTP GET operation on the endpoint.
* You can represent values for this setting in two ways:
  * As a single capability name, for example, "admin_all_objects".
  * As an expression for multiple capabilities, using the 'and' or 'or' operators.
    You can group capabilities together using parentheses ("()") to create
    complex expressions.
  * For example, if you specify "capability.post = (edit_monitor or edit_sourcetypes) and (edit_user and
edit_tcp)"
    then the user must hold one of 'edit_monitor' or 'edit_sourcetypes' and both
    'edit_user' and 'edit_tcp' before they can perform an HTTP POST operation on
    the endpoint.
  * Both setting formats can use either value format as long as the
    capabilities you specify are valid.
* Regardless of the HTTP request method that the user submits,
  the request can only act upon the handlers that this endpoint exposes
  with the 'members' setting. To set granular capability checking over
  multiple custom handlers, create multiple [admin:<uniqueName>]
  stanzas with the same name and use the 'members' setting to define different
  custom handlers within each stanza.
* No default.

maxRestResults = <unsigned integer>
* The maximum number of results that a REST API call can return.
* A REST API call fails if it tries to retrieve more results than you specify here.
* A value of 0, or no value, means that REST API calls can return any number of results, if possible.
* Do not change this setting without first consulting with Splunk Support.
* Default: 0

streamlineXmlSerialization = <boolean>
* Determines how the web server produces XML output for the results required by REST API calls.
* Only applies when a REST API call asks for XML outputs.
```

```
* A value of "true" means the web server produces XML output for each result one by one, which
  consumes less memory.
* A value of "false" means the web server produces XML output for all results at once, which
  consumes more memory.
* Do not change this setting without first consulting with Splunk Support.
* Default: true

[admin_external:<uniqueName>]
* 'admin_external'
* Register Python handlers for the Extensible Administration Interface (EAI).
* The handler is exposed via its "uniqueName".
* NOTE: Splunkd does not honor capability checks under this stanza.
  Define capability checks on endpoints under [admin:*] stanzas instead.
  handlertype = <string>
* The script type.
* Currently the only valid value is "python".

python.version={default|python|python2|python3}
* For Python scripts only, selects which Python version to use.
* Either "default" or "python" select the system-wide default Python version.
* Optional.
* Default: not set; uses the system-wide Python version.

handlerfile=<string>
* Script to execute.
* For bin/myAwesomeAppHandler.py, specify only myAwesomeAppHandler.py.

handlerpersistentmode = <boolean>
* Set to "true" to run the script in persistent mode and
  keep the process running between requests.

passHttpHeaders = <boolean>
* Determines whether splunkd passes HTTP request headers to the handler.
* A value of "true" means splunkd passes the HTTP request headers
  to the handler.
* Currently not supported. This setting is related to a feature that is
  still under development.
* Default: false

handleractions = <comma-separated list>
* a list of EAI actions supported by this handler.
* Valid values: create, edit, list, delete, _reload

[validation:<handler-name>]
* Validation stanzas.
* Add stanzas using the following definition to add argument
  validation to the appropriate EAI handlers.

<field> = <validation-rule>
* <field> is the name of the field whose value is validated when an
  object is being saved.
* <validation-rule> is an eval expression using the validate() function to
  evaluate argument correctness and return an error message. If you use a boolean
  returning function, a generic message is displayed.
* <handler-name> is the name of the REST endpoint that this stanza applies to.
  handler-name is what is used to access the handler via
  /servicesNS/<user>/<app/admin/<handler-name>.
* For example:
  action.email.sendresult = validate( isbool('action.email.sendresults'), "'action.email.sendresults' must
be a boolean value").
* NOTE: Use "'" or "$" to enclose field names that contain non-alphanumeric characters.
```

```
[eai:<EAI handler name>]
* 'eai'
* Settings to alter the behavior of EAI handlers in various ways.
* Users do not need to edit these settings.

showInDirSvc = <boolean>
* Whether configurations managed by this handler should be enumerated via the
  directory service, used by SplunkWeb's "All Configurations" management page.
* Default: false

desc = <string>
* Allows for renaming the configuration type of these objects
  when enumerated via the directory service.

[input:...]
* Miscellaneous parameters.
* The undescribed settings in these stanzas all operate according to the
  descriptions listed under the [script] stanza above.
* Users do not need to edit these settings. They only exist to quiet
  down the configuration checker.

dynamic = <boolean>
* If set to "true", listen on the socket for data.
* If set to "false", data is contained within the request body.
* Default: false

[peerupload:...]
path = <path>
* The path to search through to find configuration bundles from search peers.

untar = <boolean>
* Whether or not to untar a file once the transfer is complete.

[proxybundleupload:...]
path = <path>
* The path to search through to find proxy configuration bundles from search heads.

untar = <boolean>
* Whether or not to untar a file once the transfer is complete.

[proxybundleuploadrshcluster:...]
path = <path>
* The path to search through to find proxy configuration bundles from search heads.

untar = <boolean>
* Whether or not to untar a file once the transfer is complete.

[restreplayshc]
methods =  <comma-separated list>
* REST methods that are replayed.
* Available fields: POST, PUT, DELETE, HEAD, GET

nodelists = <comma-separated list>
* Strategies for replay.
* Available fields: shc, nodes, filternodes
* "shc" replays to all other nodes in a search head cluster.
* "nodes" provide raw comma-separated URIs in nodes variable.
* "filternodes" filters out specific nodes. It is always applied
  after other strategies.

nodes = <comma-separated list>
* A list of management URIs (specific nodes) that
```

```
  you want the REST call to be replayed to.

filternodes = <comma-separated list>
* A list of management URIs (specific nodes) that
  you do not want the REST call to be replayed to.

[proxy:appsbrowser]
destination = <URL>
* The protocol, subdomain, domain, port, and path
  of the Splunkbase API used to browse apps.
* Default: https://splunkbase.splunk.com/api
```

restmap.conf.example

```
#   Version 9.2.2
#
# This file contains example REST endpoint configurations.
#
# To use one or more of these configurations, copy the configuration block into
# restmap.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# The following are default REST configurations.  To create your own endpoints,
# modify the values by following the spec outlined in restmap.conf.spec.


# /////////////////////////////////////////////////////////////////////////////
#  global settings
# /////////////////////////////////////////////////////////////////////////////

[global]

# indicates if auths are allowed via GET params
allowGetAuth=false

#The default handler (assuming that we have PYTHONPATH set)
pythonHandlerPath=$SPLUNK_HOME/bin/rest_handler.py



# /////////////////////////////////////////////////////////////////////////////
#  internal C++ handlers
# NOTE: These are internal Splunk-created endpoints. 3rd party developers can
# only use script or search can be used as handlers.
# (Please see restmap.conf.spec for help with configurations.)
# /////////////////////////////////////////////////////////////////////////////

[SBA:sba]
match=/properties
capability=get_property_map

[asyncsearch:asyncsearch]
match=/search
capability=search
```

```
[indexing-preview:indexing-preview]
match=/indexing/preview
capability=(edit_monitor or edit_sourcetypes) and (edit_user and edit_tcp)
```

rolling_upgrade.conf

The following are the spec and example files for `rolling_upgrade.conf`.

rolling_upgrade.conf.spec

```
# This file contains descriptions of the settings you can use to configure the
# splunk-rolling-upgrade app.
```

[logging]

```
log_level = [DEBUG|INFO|WARN|ERROR]
* The severity level at which the splunk-rolling-upgrade app writes log file
  entries.
* splunk-rolling-upgrade writes logs at the level you set in 'log_level' and
  any higher levels. For example, if you set 'log_level = WARN',
  the app writes logs at both WARN and ERROR severity levels.
* Logging levels increase in order of severity, as follows (lower levels provide more
  information, but increase log file size):
  * DEBUG - Detailed information for diagnosing problems.
  * INFO  - Confirmation that things are working as expected.
  * WARN  - A warning of a recoverable fault or a problem that might
            occur in the future.
  * ERROR - A problem that is causing the software to not run as expected.
* Default: INFO
```

[requests]

```
# These settings control the client interface that connects to Splunk software
# via REST API.
retries = <positive integer>
* The maximum number of times the splunk-rolling-upgrade app retries a connection
  attempt before terminating the REST API request.
* Default: 2

delay = <non-negative integer>
* The initial amount of time, in seconds, splunk-rolling-upgrade app waits before
  retrying a REST call.
* Default: 1

timeout = <positive integer>
* The amount of time, in seconds, the splunk-rolling-upgrade app waits for a
  response from a REST API request before terminating the request.
* Default: 30
```

[process_runner]

```
# These settings manage internal sub-processes that the splunk-rolling-upgrade app creates.
timeout = <positive integer>
```

```
* The maximum about of time, in seconds, the splunk-rolling-upgrade app waits for a process to terminate.
  If the elapsed time exceeds the timeout value, the entire rolling upgrade fails.
* Default: 600
```

[kvstore_retry]

```
# Kvstore requires some time to be fully initialised. These settings are used to define
# how long an upgrade has to wait for kvstore initialisation.
max_tries = <positive integer>
* The maximum number of times the splunk-rolling-upgrade app checks for
  kvstore readiness.
* Default: 10

initial_delay_after_each_retry = <non-negative integer>
* The amount of time, in seconds, the splunk-rolling-upgrade app waits before
  checking again if the kvstore is properly initialized.
* Default: 20
```

[cluster_retry]

```
# When dealing with a Search Head Cluster there are few things that must happen in order to have
# a fully initialised cluster. Each peer must be ready (and their corresponding kvstore must be
# properly replicated), and a captain must be elected. This requires some time, depending on the
# cluster size.
max_tries = <positive integer>
* The maximum number of times the splunk-rolling-upgrade app checks for search
  head cluster readiness.
* Default: 10

initial_delay_after_each_retry = <non-negative integer>
* The amount of time, in seconds, the splunk-rolling-upgrade apps waits before
  rechecking search head cluster readiness.
* Default: 20
```

[peers_readiness_retry]

```
# Settings to control interaction with search head cluster members.
max_tries = <positive integer>
* The maximum number of times the splunk-rolling-upgrade app checks whether
  all historical searches are complete.
* Default: 20

initial_delay_after_each_retry = <non-negative integer>
* The amount of time, in seconds, the splunk-rolling-upgrade app waits before
  rechecking whether all searches are complete.
* Default: 20
```

[downloader]

```
# These settings control Splunk package source.
package_path = <string>
* Specifies the URI path to a new Splunk Enterprise installation pacakge.
  Authentication is not supported.
* Supported package file formats are .tgz, .rpm, and .deb.
* Supported URI schemas are file://, http:// and https:// only. All other schemas (such as
  ftp://) result in an error.
```

```
* Default: none
```

[hook]

```
# These settings specify a hook script that installs Splunk Enterprise on
# Linux machines during automated rolling upgrade. To install from a .tgz
# archive, use the default 'install.tgz.sh' script, which unpacks the archive
# under $SPLUNK_HOME and restarts Splunk. To install from .rpm or .deb
# packages, you must write a custom hook script. For instructions on how to
# write your own hook script, see Splunk Enterprise documentation.
install_script_path = <string>
* Specifies the path to the hook script that installs Splunk Enterprise.
* Default: $SPLUNK_HOME/etc/apps/splunk-rolling-upgrade/hooks/install_tgz.sh
```

rolling_upgrade.conf.example

```
No example
```

savedsearches.conf

The following are the spec and example files for `savedsearches.conf`.

savedsearches.conf.spec

```
#   Version 9.2.2
#
# This file contains possible setting/value pairs for saved search entries in
# the savedsearches.conf file.  You can configure saved searches by creating
# your own savedsearches.conf file.
#
# There is a default savedsearches.conf file in
# $SPLUNK_HOME/etc/system/default. To set custom configurations, place a
# savedsearches.conf file in $SPLUNK_HOME/etc/system/local/. For examples, see
# the savedsearches.conf.example file. You must restart Splunk to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top of
#    the file.
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, settings are combined. In the case of multiple
#    definitions of the same settings, the last definition in the file wins.
#  * If a setting is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.
```

The possible settings for the savedsearches.conf file are:

```
[<stanza name>]
* Create a unique stanza name for each saved search.
* Follow the stanza name with any number of the following settings.
* If you do not specify a setting, Splunk software uses the default.

disabled = <boolean>
* Disable your search by setting 'disabled=true'.
* You cannot run a disabled search.
* This setting is typically used to prevent a scheduled search from running
  on its schedule, without deleting the stanza for the search in the
  savedsearches.conf file.
* Default: false

search = <string>
* The actual search string for the saved search.
  * For example, 'search = index::sampledata http NOT 500'.
* Your search can include macro searches for substitution.
  * To learn more about creating a macro search, search the documentation for
    "macro search."
* Multi-line search strings currently have some limitations. For example, use
  with the search command '|savedsearch' does not currently work with multi-line
  search strings.
* No default.

dispatchAs = [user|owner]
* When the saved search is dispatched using the "saved/searches/{name}/dispatch"
  endpoint, this setting controls what user that search is dispatched as.
* This setting is only meaningful for shared saved searches.
* When dispatched as "user", the search is run as if the requesting user owned
  the search.
* When dispatched as "owner", the search is run as if the owner of the search
  dispatched the search, no matter which user requested it.
* If the 'force_saved_search_dispatch_as_user' setting, in the limits.conf
  file, is set to "true", then the 'dispatchAs' setting is reset to "user" while
  the saved search is dispatching.
* Default: owner
```

Scheduling options

```
enableSched = [0 | 1]
* Specifies whether or not to run the search on a schedule.
* The only acceptable values for this setting are 0 and 1.
* Set this to 1 (true) to run your search on a schedule.
* Default: 0

cron_schedule = <cron string>
* The cron schedule that is used to run this search.
* For example: */5 * * * *  causes the search to run every 5 minutes.
* You can use standard cron notation to define your scheduled search interval.
  In particular, cron can accept this type of notation: 00,20,40 * * * *, which
  runs the search every hour at hh:00, hh:20, hh:40.
```

```
  A cron of 03,23,43 * * * * runs the search every hour at hh:03, hh:23, hh:43.
* To reduce system load, schedule your searches so that they are staggered over
  time. Running all of the saved searches every 20 minutes (*/20) means all of
  the searches would launch at hh:00 (20, 40) and might slow your system every
  20 minutes.
* The Splunk cron implementation does not currently support names of months or
  days.
* No default.

schedule = <cron-style string>
* This setting is DEPRECATED as of version 4.0.
* For more information, see the pre-4.0 spec file.
* Use 'cron_schedule' to define your scheduled search interval.

allow_skew = <percentage>|<duration-specifier>
* Lets the search scheduler randomly distribute scheduled searches more evenly
  over the scheduled time periods.
* When set to non-zero for searches with the following cron_schedule values,
  the search scheduler randomly "skews" the second, minute, and hour that the
  search actually runs on:
    * * * * *     Every minute.
    */M * * * *   Every M minutes (M > 0).
    0 * * * *     Every hour.
    0 */H * * *   Every H hours (H > 0).
    0 0 * * *     Every day (at midnight).
* When set to non-zero for a search that has any other 'cron_schedule' setting,
  the search scheduler can only randomly skew the second that the search runs
  on.
* The amount of skew for a specific search remains constant between edits of
  the search.
* To specify a percentage: Use an integer value followed by the percent '%'
  symbol. This specifies the maximum amount of time to skew, as a percentage of
  the scheduled search period.
* To specify a duration: Use <integer><timescale> to specify a maximum duration.
  Supported units are:
    m, min, minute, mins, minutes
    h, hr, hour, hrs, hours
    d, day, days
  The <timescale> is required.
* Skew examples:
    100% (For an every-5-minute search = 5 minutes maximum)
    50%  (For an every-1-minute search = 30 seconds maximum)
    5m = 5 minutes maximum
    1h = 1 hour maximum
* A value of 0 does not allow a skew to occur.
* Default: 0

max_concurrent = <unsigned integer>
* The maximum number of concurrent instances of this search that the scheduler
  is allowed to run.
* Default: 1

realtime_schedule = <boolean>
* Controls the way the scheduler computes the next run time of a scheduled
  search.
* When set to 'true', the scheduler determines the next scheduled search run
  time based on the current time.
    * NOTE: When set to 'true', the scheduler might skip some execution periods
      to make sure that the scheduler is executing the searches that are running
      over the most recent time range.
* When set to 'false', the scheduler determines the next scheduled search run
  time based on the last run time for the search. This is called continuous
```

```
   scheduling.
    * NOTE: When set to 'false', the scheduler never skips scheduled execution
      periods. However, the execution of the saved search might fall behind
      depending on the scheduler's load.
    * Use continuous scheduling whenever you enable the 'summary index' option.
* The scheduler tries to run searches that have 'realtime_schedule' set to true
  before it runs searches that have continuous scheduling
  (realtime_schedule = false).
* Default: true

schedule_priority = [default | higher | highest]
* Raises the scheduling priority of a search:
  * When set to "default", this setting specifies that there is no increase to
    the scheduling priority.
  * When set to "higher", this setting specifies that the scheduling priority
    is higher than other searches of the same scheduling tier. While there are
    four tiers of priority for scheduled searches, only the following are
    affected by this setting:
      1. Real-Time-Scheduled (realtime_schedule=1).
      2. Continuous-Scheduled (realtime_schedule=0).
  * When set to "highest", this setting specifies that the scheduling priority
    is higher than other searches regardless of scheduling tier. However,
    real-time-scheduled searches with 'schedule_priority = highest' always have
    priority over continuous scheduled searches with 'schedule_priority =
    highest'.
  * The high-to-low order is:
      RTSS(H) > CSS(H) > RTSS(h) > RTSS(d) > CSS(h) > CSS(d)
    Where:
        RTSS = real-time-scheduled search
        CSS = continuous-scheduled search
        d = default
        h = higher
        H = highest
* The scheduler honors a non-default priority only when the search owner has
  the 'edit_search_schedule_priority' capability.
  * A non-default priority is mutually exclusive with a non-zero
    'schedule_window' (see below). If a user specifies both for a scheduled
    search, the scheduler honors the priority only.
  * However, if a user specifies both settings for a search, but the search
    owner does not have the 'edit_search_scheduler_priority' capability, then
    the scheduler ignores the priority setting and honors the 'schedule_window'.
* CAUTION: Having too many searches with a non-default priority impedes the
  ability of the scheduler to minimize search starvation. Use this setting
  only for mission-critical searches.
* Default: default

schedule_window = <unsigned integer> | auto
* When 'schedule_window' is non-zero, it indicates to the scheduler that the
  search does not require a precise start time. This gives the scheduler
  greater flexibility when it prioritizes searches.
* When 'schedule_window' is set to an integer greater than 0, it specifies the
  "window" of time (in minutes) that a search may start within.
    * The 'schedule_window' must be shorter than the period of the search.
    * Schedule windows are not recommended for searches that run every minute.
* When set to 0, there is no schedule window. The scheduler starts the search
  as close to its scheduled time as possible.
* When set to "auto," the scheduler calculates the 'schedule_window' value
  automatically.
    * For more information about this calculation, see the search scheduler
      documentation.
* A non-zero 'schedule_window' is mutually exclusive with a non-default
  'schedule_priority'. See 'schedule_priority' for details.
```

```
* Default: 0 for searches that are owned by users with the
           'edit_search_schedule_window' capability.
           For these searches, this value can be changed.
* Default: auto for searches that are owned by users that do not have the
           'edit_search_schedule_window' capability.
           For these searches, this setting cannot be changed.

schedule_as = [auto|classic|prjob]
* Specifies whether a scheduled search should use parallel reduce search
  processing each time it runs.
* When set to 'auto', the Splunk software determines automatically whether
  this scheduled search should use parallel reduce search processing, each time
  it runs. This means it might not use parallel reduce processing some of the
  time or all of the time. For details, please check 'autoAppliedPercentage' in
  'parallelreduce' stanza.
* When set to 'classic', the Splunk software is forced to NOT use parallel reduce
  search processing for this scheduled search, each time it runs.
* When set to 'prjob', the Splunk software is forced to use parallel reduce
  search processing for this scheduled search, each time it runs.
* Default: 'auto'
```

Workload management options

```
workload_pool = <name of workload pool>
* Specifies the name of the workload pool to be used by this search.
* There are multiple workload pools defined in the workload_pools.conf file.
  Each workload pool has different resource limits associated with it, for
  example, CPU, Memory, etc.
* The search process of this search is launched into the 'workload_pool'
  specified above.
* The 'workload_pool' used should be defined in the workload_pools.conf file.
* If workload management is enabled and a explicit 'workload_pool' is not
  specified, the 'default_pool' defined in the workload_pools.conf file is used.
```

Notification options

```
counttype = number of events | number of hosts | number of sources | custom | always
* Set the type of count for alerting.
* Used with the 'relation' and 'quantity' settings.
* NOTE: If you specify "always," do not set 'relation' or 'quantity'.
* Default: always

relation = greater than | less than | equal to | not equal to | drops by | rises by
* Specifies how to compare against 'counttype'.
* Default: empty string

quantity = <integer>
* Specifies a value for the 'counttype' and 'relation' settings, to determine
  the condition under which an alert is triggered by a saved search.
* Think of it as a sentence constructed like this: <counttype> <relation>
  <quantity>.
  * For example, "number of events [is] greater than 10" sends an alert when the
    count of events is larger than by 10.
  * For example, "number of events drops by 10%" sends an alert when the count
    of events drops by 10%.
```

```
* Default: empty string

alert_condition = <search string>
* Contains a conditional search that is evaluated against the results of the
  saved search. Alerts are triggered if the specified search yields a
  non-empty search result list.
* Default: empty string


#*******
# Generic action settings.
# For a comprehensive list of actions and their arguments, refer to the
# alert_actions.conf file.
#*******

action.<action_name> = <boolean>
* Indicates whether the action is enabled for a particular saved
  search.
* The 'action_name' can be: email | populate_lookup | script | summary_index
* For more about your defined alert actions see the alert_actions.conf file.
* Default: empty string

action.<action_name>.<parameter> = <value>
* Overrides an action's <parameter> as defined in the alert_actions.conf file,
  with a new <value> for this saved search only.
* Default: empty string
```

Settings for email action

```
action.email = <boolean>
* Specifies whether the email action is enabled for this search.
* Default: false

action.email.to = <email list>
* REQUIRED. This setting is not defined in the alert_actions.conf file.
* Set a comma-delimited list of recipient email addresses.
* Default: empty string

* NOTE: When configured in Splunk Web, the following email settings
  are written to this conf file only if their values differ
  from the settings in the alert_actions.conf file.

action.email.from = <email address>
* Set an email address to use as the sender's address.
* Default: splunk@<LOCALHOST>
    (or the 'from' setting in the alert_actions.conf file)

action.email.subject = <string>
* Set the subject of the email delivered to recipients.
* Default: SplunkAlert-<savedsearchname>
    (or the 'subject' setting in the alert_actions.conf file)

action.email.mailserver = <string>
* Set the address of the MTA server to be used to send the emails.
* Default: <LOCALHOST>
    (or the 'mailserver' setting in alert_actions.conf file)

action.email.maxresults = <integer>
```

```
* Set the maximum number of results to email.
* Any alert-level results threshold greater than this number is capped at this
  level.
* This value affects all methods of result inclusion by email alert: inline,
  CSV, and PDF.
* NOTE: This setting is affected globally by the 'maxresults' setting in the
  [email] stanza of the alert_actions.conf file.
* Default: 10000

action.email.include.results_link = [1|0]
* Specify whether to include a link to search results in the alert notification
  email.
* Default: 1 (true)
    (or the 'include.result.link' setting in the alert_actions.conf file)

action.email.include.search = [1|0]
* Specify whether to include the query whose results triggered the email.
* Default: 0 (false)
    (or the 'include.search' setting in the alert_actions.conf file)

action.email.include.trigger = [1|0]
* Specify whether to include the alert trigger condition.
* Default: 0 (false)
    (or the 'include.trigger' setting in the alert_actions.conf file)

action.email.include.trigger_time = [1|0]
* Specify whether to include the alert trigger time.
* Default: 0 (false) or whatever is set in the alert_actions.conf file

action.email.include.view_link = [1|0]
* Specify whether to include saved search title and a link for editing the
  saved search.
* Default: 1 (true)
    (or the 'include.view_link' setting in the alert_actions.conf file)

action.email.inline = [1|0]
* Specify whether to include search results in the body of the alert
  notification email.
* Default: 0 (false)
    (or the 'inline' setting in the alert_actions.conf file)

action.email.sendcsv = [1|0]
* Specify whether to send results as a CSV file.
* Default: 0
    (or the 'sendcsv' setting in the alert_actions.conf file)

action.email.allow_empty_attachment = <boolean>
* Specifies whether the Splunk software attaches a CSV or PDF file to an
  alert email even when the triggering alert search does not have results.
* Use this setting to override for specific alerts the default set for
  email alert actions in 'alert_actions.conf'.
* Default: set by the 'allow_empty_attachment' setting in
           'alert_actions.conf'

action.email.sendpdf = [1|0]
* Specify whether to send results as a PDF file.
* Default: 0 (false)
    (or the 'sendpdf' setting in the alert_actions.conf file)

action.email.sendresults = [1|0]
* Specify whether to include search results in the alert notification email.
* Default: 0 (false)
```

```
    (or the 'sendresults' setting in the alert_actions.conf file)
```

Settings for script action

```
action.script = <boolean>
* Specifies whether the script action is enabled for this search.
* Default: false

action.script.filename = <script filename>
* The filename, with no path, of the shell script to run.
* The script should be located in: $SPLUNK_HOME/bin/scripts/
* For system shell scripts on UNIX, or .bat or .cmd file on Windows, there
  are no further requirements.
* For other types of scripts, the first line should begin with a #! marker,
  followed by a path to the interpreter that will run the script.
  * Example: #!C:\Python27\python.exe
* Default: empty string
```

Settings for lookup action

```
action.lookup = <boolean>
* Specifies whether the lookup action is enabled for this search.
* Default: false

action.lookup.filename = <lookup filename>
* Provide the name of the CSV lookup file to write search results to.
  Do not provide a file path.
* Lookup actions can only be applied to CSV lookups.

action.lookup.append = <boolean>
* Specifies whether to append results to the lookup file defined for the
  'action.lookup.filename' setting.
* Default: false
```

Settings for summary index action

```
action.summary_index = <boolean>
* Specifies whether the summary index action is enabled for this search.
* Default: false.

action.summary_index._name = <index>
* Specifies the name of the summary index where the results of the scheduled
  search are saved.
* Default: summary

action.summary_index._type = [event | metric]
* Specifies the data type of the summary index where the Splunk software saves
  the results of the scheduled search.
* Default: event
```

```
action.summary_index._metric_dims = <comma-delimited-field-list>
* Optional
* Identify one or more fields with numeric values that the Splunk software
  should convert into dimensions during the summary indexing process.
* The Splunk software converts all fields with numeric values that are not in
  this list into measures.
* If you provide a list of fields, separate them with commas.
* Default: empty string

action.summary_index.inline = <boolean>
* Specify whether to run the summary indexing action as part of the
  scheduled search.
* NOTE: This option is considered only if the summary index action is enabled
  and is always run (in other words, if 'counttype = always').
* Default: 1 (true)

action.summary_index.<field> = <string>
* Specifies a field/value pair to add to every event that gets summary indexed
  by this search.
* You can define multiple field/value pairs for a single summary index search.

action.summary_index.force_realtime_schedule = <boolean>
* By default 'realtime_schedule' is false for a report configured for
  summary indexing. Set this attribute to 'true' or '1' to override the
  default behavior.
* CAUTION: Setting this to 'true' can cause gaps in summary data as a
  realtime_schedule
  search is skipped if search concurrency limits are violated.
* Default: 0 (false)
```

Settings for lookup table population parameters

```
action.populate_lookup = <boolean>
* Specifies whether the lookup population action is enabled for this search.
* Default: false

action.populate_lookup.dest = <string>
* Can be one of the following two options:
  * A lookup name from transforms.conf. The lookup name cannot be associated
    with KV store.
  * A path to a lookup .csv file that the search results should be copied to,
    relative to $SPLUNK_HOME.
    * NOTE: This path must point to a .csv file in either of the following
            directories:
      * etc/system/lookups/
      * etc/apps/<app-name>/lookups
      * NOTE: the destination directories of the above files must already exist.
* Default: empty string

run_on_startup = <boolean>
* Specifies whether this search runs when the Splunk platform starts
  or any edit that changes search related arguments happen. This includes search
  and dispatch.* arguments.
* If set to "true", the search is run as soon as possible during startup or
  after edit. Otherwise the search is run at the next scheduled time.
* Set 'run_on_startup' to "true" for scheduled searches that populate
  lookup tables or generate artifacts used by dashboards.
* Default: false
```

```
run_n_times = <unsigned integer>
* Runs this search exactly the specified number of times. The search is not run
  again until the Splunk platform is restarted.
* Default: 0 (infinite)
```

dispatch search options

```
dispatch.ttl = <integer>[p]
* Indicates the time to live (ttl), in seconds, for the search job artifacts
  produced by the scheduled search, if no actions are triggered.
* If the integer is followed by the letter 'p', the ttl is calculated as a
  multiple of the execution period for the scheduled search.
  * For example, if the search is scheduled to run hourly and ttl is set to 2p,
    the ttl of the artifacts is set to 2 hours.
* If an action is triggered for the scheduled search, the ttl changes to the
  ttl for the action. If multiple actions are triggered, the action with the
  largest ttl is applied to the artifacts. To set the ttl for an action, refer
  to the alert_actions.conf.spec file.
* If the scheduled search is configured as an alert, the alert must have a
  minimum ttl of 1p. At all times an alert must have an artifact accessible
  within its cron schedule.
  * If the alert has "rises by" or "drops by" in its trigger condition, the
    alert must have a minimum ttl of 2p to make this trigger condition
    possible. The "rises by" and "drops by" trigger condition elements require
    that at any given point in time the most recent two artifacts of the alert
    are available.
* For more information on the ttl for a search, see the limits.conf.spec file
  [search] stanza ttl setting.
* Default: 2p

dispatch.buckets  = <integer>
* The maximum number of timeline buckets.
* Default: 0

dispatch.max_count = <integer>
* The maximum number of results before finalizing the search.
* Default: 500000

dispatch.max_time = <integer>
* The maximum amount of time, in seconds, before finalizing the search.
* Default: 0

dispatch.lookups = 1| 0
* Enables or disables lookups for this search.
* Specify 1 to enable, 0 to disable.
* Default: 1

dispatch.earliest_time = <time-str>
* Specifies the earliest time for this search. Can be a relative or absolute
  time.
* If this value is an absolute time, use the 'dispatch.time_format' setting
  to format the value.
* Default: empty string

dispatch.latest_time = <time-str>
* Specifies the latest time for this saved search. Can be a relative or
  absolute time.
* If this value is an absolute time, use the 'dispatch.time_format' setting
```

```
  to format the value.
* Default: empty string

dispatch.index_earliest = <time-str>
* Specifies the earliest index time for this search. Can be a relative or
  absolute time.
* If this value is an absolute time, use the 'dispatch.time_format' setting
  to format the value.
* Default: empty string

dispatch.index_latest= <time-str>
* Specifies the latest index time for this saved search. Can be a relative or
  absolute time.
* If this value is an absolute time, use the 'dispatch.time_format' setting
  to format the value.
* Default: empty string

dispatch.time_format = <time format str>
* Defines the time format that is used to specify the earliest and latest
  time.
* Default: %FT%T.%Q%:z

dispatch.spawn_process = 1 | 0
* Specifies whether a new search process is started when this saved search
  is run.
* Default: 1 (true)

dispatch.auto_cancel = <integer>
* Specifies the amount of inactive time, in seconds, after which the job
  is automatically canceled.
* 0 means to never auto-cancel the job.
* Default: 0

dispatch.auto_pause = <integer>
* Specifies the amount of inactive time, in seconds, after which the
  search job is automatically paused.
* 0 means to never auto-pause the job.
* To restart a paused search job, specify 'unpause' as an action to POST
  search/jobs/{search_id}/control.
* auto_pause only goes into effect once. Unpausing after auto_pause does not
  put auto_pause into effect again.
* Default: 0

dispatch.reduce_freq = <integer>
* Specifies the frequency, in number of intermediary results chunks, that
  the MapReduce reduce phase should run on the accumulated map values.
* Default: 10

dispatch.allow_partial_results = <boolean>
* Specifies whether the search job can proceed to provide partial results if a search
  peer fails. When set to false, the search job fails if a search peer providing
  results for the search job fails.
* Default: true

dispatch.rt_backfill = <boolean>
* Specifies whether to do real-time window backfilling for scheduled real-time
  searches.
* Default: false

dispatch.indexedRealtime = <boolean>
* Specifies whether to use 'indexed-realtime' mode when doing real-time
  searches.
```

```
* Overrides the setting in the limits.conf file for the
  'indexed_realtime_use_by_default' setting in the [realtime] stanza.
* This setting applies to each job.
* See the [realtime] stanza in the limits.conf.spec file for more information.
* Default: The value for 'indexed_realtime_use_by_default' in the limits.conf
  file.

dispatch.indexedRealtimeOffset = <integer>
* Controls the number of seconds to wait for disk flushes to finish.
* Overrides the setting in the limits.conf file for the
  'indexed_realtime_disk_sync_delay' setting in the [realtime] stanza.
* This setting applies to each job.
* See the [realtime] stanza in the limits.conf.spec file for more information.
* Default: The value for 'indexed_realtime_disk_sync_delay' in the limits.conf
  file.

dispatch.indexedRealtimeMinSpan = <integer>
* Minimum seconds to wait between component index searches.
* Overrides the setting in the limits.conf file for the
  'indexed_realtime_default_span' setting in the [realtime] stanza.
* This setting applies to each job.
* See the [realtime] stanza in the limits.conf.spec file for more information.
* Default: The value for 'indexed_realtime_default_span' in the limits.conf
  file.

dispatch.rt_maximum_span = <integer>
* The max seconds allowed to search data which falls behind realtime.
* Use this setting to set a limit, after which events are not longer considered
  for the result set. The search catches back up to the specified delay from
  realtime and uses the default span.
* Overrides the setting in the limits.conf file for the
  'indexed_realtime_maximum_span' setting in the [realtime] stanza.
* This setting applies to each job.
* See the [realtime] stanza in the limits.conf.spec file for more information.
* Default: the value for 'indexed_realtime_maximum_span' in the limits.conf
  file.

dispatch.sample_ratio = <integer>
* The integer value used to calculate the sample ratio. The formula is
  1 / <integer>.
* The sample ratio specifies the likelihood of any event being included in the
  sample.
* For example, if sample_ratio = 500, each event has a 1/500 chance of being
  included in the sample result set.
* Default: 1

dispatch.rate_limit_retry = <boolean>
* Specifies whether the search job will be re-run in case of failure caused by
  search requests throttling on remote peers.
* Currently this setting only applies when used in SHC.
* Overrides value of 'allow_partial_results'.
* Does not apply to real time searches.
* Default: false

restart_on_searchpeer_add = 1 | 0
* Specifies whether to restart a real-time search managed by the scheduler when
  a search peer becomes available for this saved search.
* NOTE: The peer can be a newly added peer or a peer that has been down and has
        become available.
* Default: 1 (true)
```

durable search options

```
durable.track_time_type = [ _time | _indextime | none ]
* Indicates that a scheduled search is durable and specifies how the search
  tracks events.
  * A durable search is a search that tries to ensure the delivery of all
    results, even when the search process is slowed or stopped by runtime
    issues like rolling restarts, network bottlenecks, and even downed servers.
  * When durable searches encounter search errors that they cannot recover
    from, they do not return any results.
  * When a durable scheduled search job fails in this manner, the Splunk
    software reschedules a new run of the durable search over the same period
    of time to backfill the missing data. See the 'durable.backfill_type' and
    'durable.max_backfill_intervals' settings for more information.
  * This setting cannot be applied to real-time and ad hoc searches.
  * For searches of metric data, only the '_time' setting is available.
* If set to '_time', the durable search tracks each event by its original
  timestamp.
* If set to '_indextime', the durable search tracks each event by the the time
  that it is indexed.
* If this setting is set to 'none' or not set, the search is not durable.
* Default: Not set

durable.lag_time = <unsigned integer>
* Specifies the search time delay, in seconds, that a durable search uses to catch
  events that are ingested or indexed late.
* This setting takes effect only for searches that have a setting for
  'durable.track_time_type'.
* In most cases, '60' (1 minute) is a good 'lag_time' for durable searches that
  track '_indextime'.
* If your durable search tracks '_time', check to see how long the events for
  the search are delayed at indexing before setting a 'lag_time' for it.
* Default: 0

durable.backfill_type = [ auto | time_interval | time_whole ]
* Specifies how the Splunk software backfills the lost search results of failed
  scheduled search jobs.
* When set to 'time_whole', the Splunk software schedules a single backfill
  search job with a time range that spans the combined time ranges of all
  failed scheduled search jobs. The 'time_whole' setting can be applied only to
  searches that are streaming, where the results are raw events without
  additional aggregation.
* When set to 'time_interval', the Splunk software schedules multiple backfill
  search jobs, one for each failed scheduled search job. The backfill jobs have
  time ranges that match those of the failed jobs. The 'time_interval' setting
  can be applied to both streaming and non-streaming searches,
* When set to 'auto', the Splunk software decides the backfill type by checking
  whether the search is streaming or not. If the search is streaming, the
  Splunk software uses the 'time_whole' backfill type. Otherwise, it uses the
  'time_interval' backfill type.
* This setting takes effect only for searches that have a setting for
  'durable.track_time_type'.
* Default: auto

durable.max_backfill_intervals = <unsigned integer>
* Specifies the maximum number of cron intervals (previous scheduled search
  jobs) that the Splunk software can attempt to backfill for this search, when
  those jobs have incomplete events.
* This setting takes effect only for searches that have a setting for
  'durable.track_time_type'.
* For example, if 'durable.max_backfill_intervals' is set to '100', the maximum
```

```
  backfill time range for a search is 100 multiplied by the cron interval for
  the scheduled search.
* Default: 0 (unlimited)
```

auto summarization options

```
auto_summarize  = <boolean>
* Specifies if the scheduler should ensure that the data for this search is
  automatically summarized.
* Default: false

auto_summarize.command = <string>
* A search template to use to construct the auto summarization for this search.
* DO NOT change this setting unless you know what you're doing.

auto_summarize.timespan = <time-specifier> (, <time-specifier>)*
* Comma-delimited list of time ranges that each summarized chunk should span.
  This comprises the list of available granularity levels for which summaries
  would be available. For example, a timechart over the last month whose
  granularity is at the day level should set this to "1d". If you need
  the same data summarized at the hour level because you need to have weekly
  charts then use: "1h,1d".
* This setting does not support "1w" timespans.

auto_summarize.cron_schedule = <cron-string>
* Cron schedule to use to probe or generate the summaries for this search.

auto_summarize.dispatch.<arg-name> = <string>
* Any dispatch.* options that need to be overridden when running the summary
  search.

auto_summarize.suspend_period = <time-specifier>
* The amount of time to suspend summarization of this search if the
  summarization is deemed unhelpful.
* Default: 24h

auto_summarize.max_summary_size = <unsigned integer>
* The minimum summary size when to start testing its helpfulness.
* Default: 52428800 (5MB)

auto_summarize.max_summary_ratio = <positive decimal>
* The maximum ratio of summary_size/bucket_size when to stop summarization and
  deem it unhelpful for a bucket.
* NOTE: The test is only performed if the summary size is larger
  than the 'auto_summarize.max_summary_size' setting.
* Default: 0.1

auto_summarize.max_disabled_buckets = <unsigned integer>
* The maximum number of buckets with the suspended summarization before the
  summarization search is completely stopped and the summarization of the
  search is suspended for the value specified in the
  'auto_summarize.suspend_period' setting.
* Default: 2

auto_summarize.max_time = <unsigned integer>
* The maximum amount of time that the summary search is allowed to run.
* NOTE: This is an approximate time and the summarize search will be stopped at
  clean bucket boundaries.
* Default: 3600
```

```
auto_summarize.hash = <string>
* An auto generated setting.

auto_summarize.normalized_hash = <string>
* An auto generated setting.

auto_summarize.max_concurrent = <unsigned integer>
* The maximum number of concurrent instances of this auto summarizing search,
  that the scheduler is allowed to run.
* Defaults: 1

auto_summarize.workload_pool = <name of workload pool>
* Sets the name of the workload pool that is used by this auto summarization.
* There are multiple workload pools defined in workload_pools.conf.
  Each workload pool has different resource limits associated with it,
  for example, CPU, Memory, etc.
* The search process of this auto summarization are launched into the
  workload_pool specified above.
* The workload_pool used should be defined in workload_pools.conf.
* If workload management is enabled and an explicit workload_pool is not
  specified, the workload rules defined in workload_rules.conf try to put the
  search into a proper pool as specified in some rule. If there is no rule
  defined for this search, the default_pool defined in workload_pools.conf is
  used.
```

alert suppression/severity/expiration/tracking/viewing settings

```
alert.suppress = <boolean>
* Specifies whether alert suppression is enabled for this scheduled search.
* Default: false

alert.suppress.period = <time-specifier>
* Sets the suppression period. Use [number][time-unit] to specify a time.
* For example: 60 = 60 seconds, 1m = 1 minute, 1h = 60 minutes.
* Honored if and only if 'alert.suppress = 1'.
* Default: empty string

alert.suppress.fields = <comma-delimited-field-list>
* List of fields to use when suppressing per-result alerts. This field *must*
  be specified if the digest mode is disabled and suppression is enabled.
* Default: empty string.

alert.suppress.group_name = <string>
* Optional.
* Use this setting to define an alert suppression group for a set of alerts
  that are running over the same or very similar datasets. Do this to avoid
  getting multiple triggered alert notifications for the same data.
* All alerts with the same 'alert.suppress.group_name' value are in the same
  alert suppression group, as long as they are all owned by the same user.
  * Alerts belonging to different users cannot be included in the same
    suppression group, even if they all have the same 'group_name'.
* When an alert within an alert suppression group is triggered, all of the
  alerts in the group are suppressed for a period of time defined by the
  'alert.suppress.period' of the triggered alert. The triggered alert performs
  its alert actions, if it has any. The other alerts in the group do not
  perform their alert actions.
  * For example, say you have an alert suppression group with five alerts. Each
    of these alerts has a different 'alert.suppress.period' and a different
    alert action. If one alert from the group with an 'alert.suppress.period'
```

```
    of 5m and an email alert action is triggered, all of the alerts in the
    group are suppressed for 5m. However, only one alert action happens: the
    email for the triggering alert.
* Default: empty string.

alert.severity = <integer>
* Sets the alert severity level.
* Valid values are: 1-debug, 2-info, 3-warn, 4-error, 5-severe, 6-fatal
* Default: 3

alert.expires = <time-specifier>
* Sets the period of time to show the alert on the Triggered Alerts page.
  * Use [number][time-unit] to specify a time.
  * For example: 60s = 60 seconds, 1m = 1 minute, 1h = 60 minutes = 1 hour etc
* This setting is only honored when 'alert.track = true' (when the "Add to
  Triggered Alerts" action is selected for the alert in Splunk Web).
* This property is valid until splunkd restarts. Restart clears the listing of
  triggered alerts.
* Default: 24h

alert.digest_mode = <boolean>
* Specifies whether Splunk applies the alert actions to the entire result set
  or to each individual result.
* Default: true

alert.track = <boolean> | auto
* Specifies whether to track the actions triggered by this scheduled search.
  * auto - determine whether to track or not based on the tracking setting of
    each action, do not track scheduled searches that always trigger actions.
  * true - force alert tracking.
  * false - disable alert tracking for this search.
* Default: auto

alert.display_view = <string>
* Name of the UI view where the emailed link for each result alerts should
  point to.
* If not specified, the value of the 'request.ui_dispatch_app' setting is used.
  If the 'request.ui_dispatch_app' setting is missing then "search" is used.
* Default: empty string

alert.managedBy = <string>
* Specifies the feature or component that created the alert.
* Default: empty string
```

UI-specific settings

```
displayview =<string>
* Defines the default UI view name (not label) in which to load the results.
* Accessibility is subject to the user having sufficient permissions.
* Default: empty string

vsid = <string>
* Defines the view state ID associated with the UI view listed in the
  'displayview' setting.
* Must match up to a stanza in the viewstates.conf file.
* Default: empty string

is_visible = <boolean>
```

```
* Specifies whether this saved search should be listed in the visible saved
  search list within apps.
* Saved searches are still visible when accessing the "Searches, reports,
  and alerts" page in Splunk Web.
* Default: true

description = <string>
* Human-readable description of this saved search.
* Default: empty string

request.ui_dispatch_app  = <string>
* Specifies a field used by Splunk UI to denote the app that this search
  should be dispatched in.
* Default: empty string

request.ui_dispatch_view = <string>
* Specifies a field used by Splunk UI to denote the view this search should be
  displayed in.
* Default: empty string
```

Display Formatting Options

```
# General options
display.general.enablePreview = [0 | 1]
display.general.type = [events|statistics|visualizations]
display.general.timeRangePicker.show = [0 | 1]
display.general.migratedFromViewState = [0 | 1]
display.general.locale = <string>

# Event options
display.events.fields = [<string>(, <string>)*]
display.events.type = [raw|list|table]
display.events.rowNumbers = [0 | 1]
display.events.maxLines = <integer>
display.events.raw.drilldown = [inner|outer|full|none]
display.events.list.drilldown = [inner|outer|full|none]
display.events.list.wrap = [0 | 1]
display.events.table.drilldown = [0 | 1]
display.events.table.wrap = [0 | 1]

# Statistics options
display.statistics.rowNumbers = [0 | 1]
display.statistics.wrap = [0 | 1]
display.statistics.overlay = [none|heatmap|highlow]
display.statistics.drilldown = [row|cell|none]
display.statistics.totalsRow = [0 | 1]
display.statistics.percentagesRow = [0 | 1]
display.statistics.show = [0 | 1]

# Visualization options
display.visualizations.trellis.enabled = [0 | 1]
display.visualizations.trellis.scales.shared = [0 | 1]
display.visualizations.trellis.size = [small|medium|large]
display.visualizations.trellis.splitBy = <string>
display.visualizations.show = [0 | 1]
display.visualizations.type = [charting|singlevalue|mapping|custom]
display.visualizations.chartHeight = <integer>
display.visualizations.charting.chart =
[line|area|column|bar|pie|scatter|bubble|radialGauge|fillerGauge|markerGauge]
```

```
display.visualizations.charting.chart.stackMode = [default|stacked|stacked100]
display.visualizations.charting.chart.nullValueMode = [gaps|zero|connect]
display.visualizations.charting.chart.overlayFields = <string>
display.visualizations.charting.drilldown = [all|none]
display.visualizations.charting.chart.style = [minimal|shiny]
display.visualizations.charting.layout.splitSeries = [0 | 1]
display.visualizations.charting.layout.splitSeries.allowIndependentYRanges = [0 | 1]
display.visualizations.charting.legend.mode = [standard|seriesCompare]
display.visualizations.charting.legend.placement = [right|bottom|top|left|none]
display.visualizations.charting.legend.labelStyle.overflowMode = [ellipsisEnd|ellipsisMiddle|ellipsisStart]
display.visualizations.charting.axisTitleX.text = <string>
display.visualizations.charting.axisTitleY.text = <string>
display.visualizations.charting.axisTitleY2.text = <string>
display.visualizations.charting.axisTitleX.visibility = [visible|collapsed]
display.visualizations.charting.axisTitleY.visibility = [visible|collapsed]
display.visualizations.charting.axisTitleY2.visibility = [visible|collapsed]
display.visualizations.charting.axisX.scale = linear|log
display.visualizations.charting.axisY.scale = linear|log
display.visualizations.charting.axisY2.scale = linear|log|inherit
display.visualizations.charting.axisX.abbreviation = none|auto
display.visualizations.charting.axisY.abbreviation = none|auto
display.visualizations.charting.axisY2.abbreviation = none|auto
display.visualizations.charting.axisLabelsX.majorLabelStyle.overflowMode = [ellipsisMiddle|ellipsisNone]
display.visualizations.charting.axisLabelsX.majorLabelStyle.rotation = [-90|-45|0|45|90]
display.visualizations.charting.axisLabelsX.majorUnit = <decimal> | auto
display.visualizations.charting.axisLabelsY.majorUnit = <decimal> | auto
display.visualizations.charting.axisLabelsY2.majorUnit = <decimal> | auto
display.visualizations.charting.axisX.minimumNumber = <decimal> | auto
display.visualizations.charting.axisY.minimumNumber = <decimal> | auto
display.visualizations.charting.axisY2.minimumNumber = <decimal> | auto
display.visualizations.charting.axisX.maximumNumber = <decimal> | auto
display.visualizations.charting.axisY.maximumNumber = <decimal> | auto
display.visualizations.charting.axisY2.maximumNumber = <decimal> | auto
display.visualizations.charting.axisY2.enabled = [0 | 1]
display.visualizations.charting.chart.sliceCollapsingThreshold = <decimal>
display.visualizations.charting.chart.showDataLabels = [all|none|minmax]
display.visualizations.charting.gaugeColors = [<hex>(, <hex>)*]
display.visualizations.charting.chart.rangeValues = [<string>(, <string>)*]
display.visualizations.charting.chart.bubbleMaximumSize = <integer>
display.visualizations.charting.chart.bubbleMinimumSize = <integer>
display.visualizations.charting.chart.bubbleSizeBy = [area|diameter]
display.visualizations.charting.fieldColors = <string>
display.visualizations.charting.fieldDashStyles = <string>
display.visualizations.charting.lineWidth = <decimal>
display.visualizations.custom.drilldown = [all|none]
display.visualizations.custom.height = <integer>
display.visualizations.custom.type = <string>
display.visualizations.singlevalueHeight = <integer>
display.visualizations.singlevalue.beforeLabel = <string>
display.visualizations.singlevalue.afterLabel = <string>
display.visualizations.singlevalue.underLabel = <string>
display.visualizations.singlevalue.unit = <string>
display.visualizations.singlevalue.unitPosition = [before|after]
display.visualizations.singlevalue.drilldown = [all|none]
display.visualizations.singlevalue.colorMode = [block|none]
display.visualizations.singlevalue.rangeValues = [<string>(, <string>)*]
display.visualizations.singlevalue.rangeColors = [<string>(, <string>)*]
display.visualizations.singlevalue.trendInterval = <string>
display.visualizations.singlevalue.trendColorInterpretation = [standard|inverse]
display.visualizations.singlevalue.showTrendIndicator = [0 | 1]
display.visualizations.singlevalue.showSparkline = [0 | 1]
display.visualizations.singlevalue.trendDisplayMode = [percent|absolute]
```

```
display.visualizations.singlevalue.colorBy = [value|trend]
display.visualizations.singlevalue.useColors = [0 | 1]
display.visualizations.singlevalue.numberPrecision = [0|0.0|0.00|0.000|0.0000]
display.visualizations.singlevalue.useThousandSeparators = [0 | 1]
display.visualizations.mapHeight = <integer>
display.visualizations.mapping.type = [marker|choropleth]
display.visualizations.mapping.drilldown = [all|none]
display.visualizations.mapping.map.center = (<decimal>,<decimal>)
display.visualizations.mapping.map.zoom = <integer>
display.visualizations.mapping.map.scrollZoom = [0 | 1]
display.visualizations.mapping.map.panning    = [0 | 1]
display.visualizations.mapping.choroplethLayer.colorMode = [auto|sequential|divergent|categorical]
display.visualizations.mapping.choroplethLayer.maximumColor = <string>
display.visualizations.mapping.choroplethLayer.minimumColor = <string>
display.visualizations.mapping.choroplethLayer.colorBins = <integer>
display.visualizations.mapping.choroplethLayer.neutralPoint = <decimal>
display.visualizations.mapping.choroplethLayer.shapeOpacity = <decimal>
display.visualizations.mapping.choroplethLayer.showBorder = [0 | 1]
display.visualizations.mapping.markerLayer.markerOpacity = <decimal>
display.visualizations.mapping.markerLayer.markerMinSize = <integer>
display.visualizations.mapping.markerLayer.markerMaxSize = <integer>
display.visualizations.mapping.legend.placement = [bottomright|none]
display.visualizations.mapping.data.maxClusters = <integer>
display.visualizations.mapping.showTiles = [0 | 1]
display.visualizations.mapping.tileLayer.tileOpacity = <decimal>
display.visualizations.mapping.tileLayer.url = <string>
display.visualizations.mapping.tileLayer.minZoom = <integer>
display.visualizations.mapping.tileLayer.maxZoom = <integer>

# Patterns options
display.page.search.patterns.sensitivity = <decimal>

# Page options
display.page.search.mode = [fast|smart|verbose]
* This setting has no effect on saved search execution when dispatched by the
  scheduler. It only comes into effect when the search is opened in the UI and
  run manually.

display.page.search.timeline.format = [hidden|compact|full]
display.page.search.timeline.scale = [linear|log]
display.page.search.showFields = [0 | 1]
display.page.search.tab = [events|statistics|visualizations|patterns]
# Deprecated
display.page.pivot.dataModel = <string>
```

Table format settings

```
# Format options
display.statistics.format.<index> = [color|number]
display.statistics.format.<index>.field = <string>
display.statistics.format.<index>.fields = [<string>(, <string>)*]

# Color format options
display.statistics.format.<index>.scale = [category|linear|log|minMidMax|sharedCategory|threshold]
display.statistics.format.<index>.colorPalette = [expression|list|map|minMidMax|sharedList]

# Number format options
display.statistics.format.<index>.precision = <integer>
display.statistics.format.<index>.useThousandSeparators = <boolean>
```

```
display.statistics.format.<index>.unit = <string>
display.statistics.format.<index>.unitPosition = [before|after]

# Scale options for 'category'
display.statistics.format.<index>.scale.categories = [<string>(, <string>)*]

# Scale options for 'log'
display.statistics.format.<index>.scale.base = <integer>

# Scale options for 'minMidMax'
display.statistics.format.<index>.scale.minType = [number|percent|percentile]
display.statistics.format.<index>.scale.minValue = <decimal>
display.statistics.format.<index>.scale.midType = [number|percent|percentile]
display.statistics.format.<index>.scale.midValue = <decimal>
display.statistics.format.<index>.scale.maxType = [number|percent|percentile]
display.statistics.format.<index>.scale.maxValue = <decimal>

# Scale options for 'threshold'
display.statistics.format.<index>.scale.thresholds = [<decimal>(, <decimal>)*]

# Color palette options for 'expression'
display.statistics.format.<index>.colorPalette.rule = <string>

# Color palette options for 'list'
display.statistics.format.<index>.colorPalette.colors = [<hex>(, <hex>)*]
display.statistics.format.<index>.colorPalette.interpolate = <boolean>

# Color palette options for 'map'
display.statistics.format.<index>.colorPalette.colors = {<string>:<hex>(, <string>:<hex>)*}

# Color palette options for 'minMidMax'
display.statistics.format.<index>.colorPalette.minColor = <hex>
display.statistics.format.<index>.colorPalette.midColor = <hex>
display.statistics.format.<index>.colorPalette.maxColor = <hex>
```

Other settings

```
embed.enabled = [0 | 1]
* Specifies whether a saved search is shared for access with a guestpass.
* The only acceptable values for this setting are 0 and 1.
* Search artifacts of a search can be viewed using a guestpass only if:
  * A token has been generated that is associated with this saved search.
    The token is associated with a particular user and app context.
  * The user to whom the token belongs has permissions to view that search.
  * The saved search has been scheduled and there are artifacts available.
    Only artifacts are available using guestpass. A search is never dispatched.
  * The saved search is not disabled, it is scheduled.
  * The saved search is not real-time.
  * The saved search is not an alert.

defer_scheduled_searchable_idxc = <boolean>
* Specifies whether to defer a continuous saved search during a searchable
  rolling restart or searchable rolling upgrade of an indexer cluster.
* Note: When disabled, a continuous saved search might return partial results.
* Default: false (disabled)

skip_scheduled_realtime_idxc = <boolean>
* Specifies whether to skip a continuous saved realtime search during a searchable
  rolling restart or searchable rolling upgrade of an indexer cluster.
```

```
* Note: When set to false, a continuous saved search might return partial results.
* Default: false (does not skip)

precalculate_required_fields_for_alerts = <boolean>
* Specifies whether to precalculate the required fields from the alert
  condition search and use the result in the main search. Giving the required
  fields to the main search may decrease performance in some cases where the
  system is bottlenecked on the search scheduler.
* If "false", the required fields are not precalculated, which may free up the
  search scheduler and improve performance, but at the cost of potentially more
  work in the main search.
* Note: Do not change unless instructed to do so by Splunk Support.
* Default: true
```

Deprecated settings

```
sendresults = <boolean>
* Use the 'action.email.sendresult' setting.

action_rss = <boolean>
* Use the 'action.rss' setting.

action_email = <string>
* Use the 'action.email' and 'action.email.to' settings.

role = <string>
* See saved search permissions.

userid = <string>
* See saved search permissions.

query = <string>
* Use the 'search' setting.

nextrun  = <integer>
* Not used anymore. The scheduler maintains this info internally.

qualifiedSearch = <string>
* Not used anymore. Splunk software computes this value during runtime.
```

savedsearches.conf.example

```
#   Version 9.2.2
#
# This file contains example saved searches and alerts.
#
# To use one or more of these configurations, copy the configuration block into
# savedsearches.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# The following searches are example searches.  To create your own search,
```

```
# modify the values by following the spec outlined in savedsearches.conf.spec.


[Daily indexing volume by server]
search = index=_internal todaysBytesIndexed LicenseManager-Audit NOT source=*web_service.log NOT
source=*web_access.log | eval Daily
_Indexing_Volume_in_MBs = todaysBytesIndexed/1024/1024 | timechart avg(Daily_Indexing_Volume_in_MBs) by
host
dispatch.earliest_time = -7d

[Errors in the last 24 hours]
search = error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )
dispatch.earliest_time = -1d

[Errors in the last hour]
search = error OR failed OR severe OR ( sourcetype=access_* ( 404 OR 500 OR 503 ) )
dispatch.earliest_time = -1h

[KB indexed per hour last 24 hours]
search = index=_internal metrics group=per_index_thruput NOT debug NOT sourcetype=splunk_web_access |
timechart fixedrange=t span=1h
 sum(kb) | rename sum(kb) as totalKB
dispatch.earliest_time = -1d

[Messages by minute last 3 hours]
search = index=_internal eps "group=per_source_thruput" NOT filetracker | eval events=eps*kb/kbps |
timechart fixedrange=t span=1m s
um(events) by series
dispatch.earliest_time = -3h

[Splunk errors last 24 hours]
search = index=_internal " error " NOT debug source=*/splunkd.log*
dispatch.earliest_time = -24h

[stats with durable search]
search = index=_internal eps | stats avg(eps) as avg, max(eps) as max, min(eps) as min
dispatch.indexed_earliest = -30m
dispatch.indexed_latest   = now

durable.track_time_type   = _indextime
durable.lag_time          = 60
durable.backfill_type     = time_interval
durable.max_backfill_intervals = 100
```

searchbnf.conf

The following are the spec and example files for `searchbnf.conf`.

searchbnf.conf.spec

```
#   Version 9.2.2
#
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
```

```
# configure the search assistant to display information in the
# UI about custom search commands.
#
# Each stanza in your local searchbnf.conf file controls a separate
# custom search command or an option to a command.
#
# There is a searchbnf.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# which is used to display information about the built-in search commands
# in the UI through the search assistant.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their
# original location.
#
# To set custom configurations, create a new file with the name
# "searchbnf.conf" in the
# $SPLUNK_HOME/etc/apps/<appname>/default/ directory.
# Then add the custom commands to the custom configuration file.
# For examples, see the "searchbnf.conf.example" file.
#
# You must restart the Splunk instance to enable configuration changes.
#
# To learn more about configuration files, including precedence, see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GENERAL FORMATTING

```
# * Adjacent tokens implicitly allow no whitespace.
# * All literals are case-insensitive.
# * Whitespace (including newlines) match \s+
#
```

DESCRIPTION FORMATTING

```
# * For the command description, when automatically converted to html
#   multiple whitespaces are removed.
# * For descriptions that extend to multiple lines end each line with
#   a backslash "\", except the last line.
# * To create a multi-paragraph description, use \p\ to cause a paragraph
#   break.
# * To force a new line and indent, use \i\ to cause a newline and
#   indent 4 spaces (<br>    )
# * <terms> are italicized.
# * UPPERCASETERMS and quoted terms are put into <code/> and render in
#   green text, in a slightly smaller font.
#
```

SYNTAX FORMATTING

```
# * Reserved characters are ("<>()|?*+") and <tokens>, everything else
#   is taken literally. Those characters need to be quoted.
#   Use \"  to represent a quote.
# * This file uses regex-like grouping and nomenclature for the syntax:
```

```
#       (): grouping
#       <term> : <term> is required
#       (<term>)? : <term> is optional
#       (<term>)* : <term> is optional and repeated 0 or more times
#       (<term>)+ : <term> is required and repeated 1 or more times
#
# * <terms> can be named for readability with a colon and a default value
#   For example, if you have a term called "field", instead of the
#   syntax "...<field> AS <field>" you can add a qualifer to the term
#   name, such as "<field:fromfield> AS <field:tofield>" and then define
#   "field" as a <string>.
```

STANZAS

```
# There are two types of stanzas, search command stanzas and options stanzas.
#
#[<command-name>-command]
# * The command stanza contains the name of the custom search command
#   and "-command" enclosed in square brackets.
#   For example, "geocode-command".
# * A searchbnf.conf file can contain multiple command stanzas,
#   one command stanza for each command.
# * Follow the command stanza with attribute/value pairs that define
#   the properties for the custom search command.
#   Some attributes are required. See ATTRIBUTES.
# * If you do not set an attribute for a given <spec>, the default
#   is used. The default values are empty.
# * Search command syntax can refer to command options. These options
#   must be defined below the command stanza in separate options stanzas.
#   It is possible to use nested options stanzas.
#   For example:
#
#   [geocode-command]
#   syntax = geocode (geocode-options)*
#   ...
#   [geocode-options]
#   syntax = (maxcount=<int>) | (maxhops=<int>) | (coordinate-options)+
#   ...
#   [coordinate-options]
#   syntax = (latitude-field=<string>) | (longitude-field=<string>)
#   ...
#
```

COMMAND STANZA STRUCTURE

```
#
# [<command-name>-command]
# syntax (Required)
# simplesyntax (Optional)
# alias (Optional)
# description (Required)
# shortdesc (Optional)
# example<number> (Optional)
# comment<number> (Optional)
# usage (Required)
# tags (Optional)
# maintainer (Deprecated)
```

```
# appears-in (Deprecated)
# related (Optional)
```

ATTRIBUTES

```
# The attribute/value pair descriptions for custom search commands.

syntax = <string>
* The syntax of the custom search command. The format is:
  syntax=<command-name> (attribute-name=<datatype>) (attribute-name=<datatype>)
* See SYNTAX FORMATTING.
* Required

simplesyntax = <string>
* Simpler version of the syntax to make it easier to understand,
  at the expense of completeness. Use only if the syntax is complex.
* Typically the simplesyntax removes rarely used options or alternate
  ways of saying the same thing.
* For example, a search command might accept values such as
  "m|min|mins|minute|minutes", but that would unnecessarily clutter
  the syntax description for the user. For the simplesyntax you can
  use one value such as "minute".
* Optional

alias = <alias list>
* Alternative names for the search command.
  Specifying an alias is discouraged.
  Users might get confused when more than one name is used for the
  same command.
* Optional

description = <string>
* A detailed description of the search command.
  See DESCRIPTION FORMATTING.
* If a shortdesc is specified, the description appears only in the
  search assistant "Full" mode. Displays under the heading "Details"
  when users click "More".
* See the "searchbnf.conf.example" file for an example.
* Required

shortdesc = <string>
* A one sentence description of the search command. If specified,
  appears in both the "Full" and "Compact" search assistant modes.
* Specify a shortdesc when the description is multiple sentences long.
* Optional

example<number> = <string>
comment<number> = <string>
* The "example" should show a common example of using the search command,
   with 1 or more attributes.
* The "comment" should explain what the command is doing in the example.
* You can specify multiple examples by appending a matching number to
  the example and corresponding comment.
* For example:
    example1 = geocode maxcount=4
    comment1 = run geocode on up to four values
    example2 = geocode maxcount=-1
    comment2 = run geocode on all values
* In Compact mode, only the first example displays in the search assistant.
```

```
* In Full mode, the top three examples display in the search assistant.
* Optional, but recommended

usage = public | private | deprecated
* Specifies if a command is public, private, or deprecated.
* The search assistant only operates on public commands.
* Required

tags = <tag list>
* One or more words that users might type into the search bar which are
  similar to the command name. The UI displays the command names
  associated with the tags.
* For example, when a user types "graph" or "report" for the "chart"
  command.
* Optional

maintainer = <name>
* The name of person who originally worked on the command or who is
  responsible for the command now.
* Does not appear in the search assistant.
* Deprecated

appears-in = <version>
* The version that the custom command first appeared in.
* Does not appear in the search assistant.
* Deprecated

related = <command list>
* List of SPL commands related to this command.
* Might help users learn about other, related commands.
* Displays in the search assistant Full mode when users click "More".
* Optional
```

searchbnf.conf.example

```
#   Version 9.2.2
#
# The following are example stanzas for searchbnf.conf configurations.
#

##################
# selfjoin
##################
[selfjoin-command]
syntax = selfjoin (<selfjoin-options>)* <field-list>
shortdesc = Join results with itself.
description = Join results with itself.  Must specify at least one field to join on.
usage = public
example1 = selfjoin id
comment1 = Joins results with itself on 'id' field.
related = join
tags = join combine unite

[selfjoin-options]
syntax = overwrite=<bool> | max=<int> | keepsingle=<int>
description = The selfjoin joins each result with other results that\
  have the same value for the join fields.  'overwrite' controls if\
  fields from these 'other' results should overwrite fields of the\
  result used as the basis for the join (default=true).  max indicates\
```

```
  the maximum number of 'other' results each main result can join with.\
  (default = 1, 0 means no limit).  'keepsingle' controls whether or not\
  results with a unique value for the join fields (and thus no other\
  results to join with) should be retained.  (default = false)
```

segmenters.conf

The following are the spec and example files for `segmenters.conf`.

segmenters.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure the segmentation of events.
#
# There is a segmenters.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name segmenters.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see segmenters.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# NOTE: Keep in mind the following limitations when working with event segmentation:
#       1) The segmenters.conf file must not have conflicting definitions for
#          different installed apps. This means that definitions within a
#          segmenters.conf that is installed in one app cannot directly conflict
#          with definitions within a segmenters.conf that is installed
#          in another app.
#       2) Definitions within segmenters.conf must match between search heads
#          and search peers.
#       3) Definitions in segmenters.conf must be visible in the global context,
#          either within a [default] stanza, or outside of any stanza.
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
```

```
#     file takes precedence.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

[<SegmenterName>]

```
* Name your stanza.
* Follow this stanza name with any number of the following setting/value
  pairs.
* If you don't specify a setting/value pair, Splunk will use the default.

MAJOR = <space separated list of breaking characters>
* Set major breakers.
* Major breakers are words, phrases, or terms in your data that are surrounded
  by set breaking characters.
* By default, major breakers are set to most characters and blank spaces.
* Typically, major breakers are single characters.
* Note: \s represents a space; \n, a newline; \r, a carriage return; and
  \t, a tab.
* Default is [ ] < > ( ) { } | ! ; , ' " * \n \r \s \t & ? + %21 %26 %2526 %3B %7C %20 %2B %3D
-- %2520 %5D %5B %3A %0A %2C %28 %29


MINOR = <space separated list of strings>
* Specifies minor breakers.
* In addition to the segments specified by the major breakers, for each minor
  breaker found, Splunk indexes the token from the last major breaker to the
  current minor breaker and from the last minor breaker to the current minor
  breaker.
* Default: / : = @ . - $ # % \\ _

INTERMEDIATE_MAJORS =  true | false
* Set this to "true" if you want an IP address to appear in typeahead as
  a, a.b, a.b.c, a.b.c.d
* The typical negative effect on performance by setting to "true" is 30%.
* Default: false

FILTER = <regular expression>
* If specified, segmentation will only take place if the regular expression matches.
* Furthermore, segmentation will only take place on the first group of the
  matching regex.
* Default: None

LOOKAHEAD = <integer>
* Specifies how far into a given event, in characters, the Splunk segments.
* LOOKAHEAD is applied after any FILTER rules.
* To disable segmentation, set to 0.
* Default: -1 (read the whole event)

MINOR_LEN = <integer>
* Specifies how long a minor token can be.
* Longer minor tokens are discarded without prejudice.
* Default: -1

MAJOR_LEN = <integer>
* Specifies how long a major token can be.
* Longer major tokens are discarded without prejudice.
* Default: -1.

MINOR_COUNT = <integer>
* Specifies how many minor segments to create for each event.
```

```
* After the specified number of minor segments are created, later minor segments are
  discarded without prejudice.
* Default: -1

MAJOR_COUNT = <integer>
* Specifies how many major segments are created for each event.
* After the specified number of major segments are created, later segments
  are discarded without prejudice.
* Default: -1
```

segmenters.conf.example

```
#   Version 9.2.2
#
# The following are examples of segmentation configurations.
#
# To use one or more of these configurations, copy the configuration block into
# segmenters.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


# Example of a segmenter that doesn't index the date as segments in syslog
# data:

[syslog]
FILTER = ^.*?\d\d:\d\d:\d\d\s+\S+\s+(.*)$


# Example of a segmenter that only indexes the first 256b of events:

[limited-reach]
LOOKAHEAD = 256


# Example of a segmenter that only indexes the first line of an event:

[first-line]
FILTER = ^(.*?)(\n|$)


# Turn segmentation off completely:

[no-segmentation]
LOOKAHEAD = 0
```

server.conf

The following are the spec and example files for `server.conf`.

server.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains settings and values to configure server options
# in server.conf.
#
# Each stanza controls different search commands settings.
#
# There is a server.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name server.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see server.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza at the top
#     of the file.
#   * Each configuration file should have at most one default stanza.
#     If you have multiple default stanzas, settings are combined. If you
#     have multiple definitions of the same settings, the last definition
#     in the file wins.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

General Server Configuration

```
[general]
serverName = <ASCII string>
* The name that identifies this Splunk software instance for features such as
  distributed search.
* Cannot be an empty string.
* Can contain environment variables.
* After any environment variables are expanded, the server name
  (if not an IPv6 address) can only contain letters, numbers, underscores,
  dots, and dashes. The server name must start with a letter, number, or an
  underscore.
```

```
* Default: $HOSTNAME

hostnameOption = [ fullyqualifiedname | clustername | shortname ]
* The type of information to use to determine how splunkd sets the 'host'
  value for a Windows
  Splunk platform instance when you specify an input stanza with
  'host = $decideOnStartup'.
* Applies only to Windows hosts, and only for input stanzas that use the
  "host = $decideOnStartup" setting and value.
* Valid values are "fullyqualifiedname", "clustername", and "shortname".
* The value returned for the 'host' field depends on Windows DNS, NETBIOS,
  and what the name of the host is.
  * 'fullyqualifiedname' uses Windows DNS to return the fully qualified
    host name as the value.
  * 'clustername' also uses Windows DNS, but sets the value to the domain
    and machine name.
  * 'shortname' returns the NETBIOS name of the machine.
* Cannot be an empty string.
* Default: shortname

sessionTimeout = <nonnegative integer>[s|m|h|d]
* The amount of time before a user session times out, expressed as a
  search-like time range.
* Examples include "24h" (24 hours), "3d" (3 days),
  "7200s" (7200 seconds, or two hours)
* Default: "1" (1 hour)


invalidateSessionTokensOnLogout = <boolean>
* A value of "true" means the SHC invalidates any tokens associated with a logged-out session
  across all nodes in the cluster.
* This setting has an effect only if search head clustering and App Key Value store are enabled.
* Splunkd on each node tries to keep the logout information in sync with other nodes in the
  cluster within the specified 'logoutCacheRefreshInterval'.
* Default: false

logoutCacheRefreshInterval = <nonnegative integer>[s|m|h|d]
* This setting controls how often splunkd on a given node updates its local cache from the
  App Key Value store when 'invalidateSessionTokensOnLogout' is enabled.
* This setting has no effect when 'invalidateSessionTokensOnLogout' is disabled.
* In normal scenarios, maximum time for changes to propogate across the cluster can be upto
  this interval, plus a few seconds; minimum can be a second or two.
* There is no guarantee that this sync will always happen within this time. If the system is
  blocked because of load or other issues like network partition, the information may not be
  propogated within the specified interval.
* Default: 30s

trustedIP = <IP address>
* Only a single IP address is allowed.
* All logins from specified IP addresses are trusted. This means a
  password is no longer required.
* Only set this if you are using Single Sign-On (SSO).

allowRemoteLogin = always|never|requireSetPassword
* Controls remote management by restricting general login. Note that this
  does not apply to trusted SSO logins from a trustedIP.
* When set to "always", all remote login attempts are allowed.
* When set to "never", only local logins to splunkd are allowed. Note that this
  still allows remote management through Splunk Web if Splunk Web is on
  the same server.
* If set to "requireSetPassword":
  * In the free license, remote login is disabled.
```

```
  * In the pro license, remote login is disabled for the "admin" user if
    the default password of "admin" has not been changed.
* NOTE: As of version 7.1, Splunk software does not support the use of default
  passwords. The "requireSetPassword" value is deprecated and might be removed
  in the future.
* Default: requireSetPassword

tar_format = gnutar|ustar
* Sets the default TAR format.
* Default: gnutar

access_logging_for_phonehome = <boolean>
* Enables/disables logging to the splunkd_access.log file for client phonehomes.
* Default: true (logging enabled)

hangup_after_phonehome = <boolean>
* Controls whether or not the deployment server hangs up the connection
  after the phonehome is done.
* By default, persistent HTTP 1.1 connections are used with the server to
  handle phonehomes. This might show higher memory usage if you have a large
  number of clients.
* If you have more than the maximum recommended concurrent TCP connection
  deployment clients, persistent connections can not help with the reuse of
  connections. Setting this setting to true helps bring down memory usage.
* Default: false (persistent connections for phonehome)

pass4SymmKey = <string>
* Authenticates traffic between:
  * A license manager and its license peers.
  * Members of a cluster.
  * A deployment server (DS) and its deployment clients (DCs).
* When authenticating members of a cluster, clustering might override the
  passphrase specified in the clustering stanza. A clustering search head
  connecting to multiple managers might further override in the
  [clustermanager:<cm-nameX>] stanza.
* When authenticating deployment servers and clients, by default, DS-DCs
  passphrase authentication is disabled. To enable DS-DCs passphrase
  authentication, you must also add the following line to the [broker:broker]
  stanza in the restmap.conf file: requireAuthentication = true
* In all scenarios, every node involved must set the same passphrase in
  the same stanzas. For example in the [general] stanza and/or
  [clustering] stanza. Otherwise, the respective communication does not proceed:
    - licensing and deployment in the case of the [general] stanza
    - clustering in case of the [clustering] stanza)
* Unencrypted passwords must not begin with "$1$". This is used by
  Splunk software to determine if the password is already encrypted.

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' can be for a particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length than
  what you specify with this setting, the platform warns you and advises that you
  change the pass4SymmKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* Default: 12

listenOnIPv6 = no|yes|only
* By default, splunkd listens for incoming connections (both REST and
  TCP inputs) using IPv4 only.
* When you set this value to "yes", splunkd simultaneously listens for
  connections on both IPv4 and IPv6.
```

```
* To disable IPv4 entirely, set listenOnIPv6 to "only". This causes splunkd
  to exclusively accept connections over IPv6. You might need to change
  the mgmtHostPort setting in the web.conf file. Use '[::1]' instead of
  '127.0.0.1'.
* Any setting of SPLUNK_BINDIP in your environment or the
  splunk-launch.conf file overrides the listenOnIPv6 value.
  In this case splunkd listens on the exact address specified.

connectUsingIpVersion = auto|4-first|6-first|4-only|6-only
* When making outbound TCP connections for forwarding event data, making
  distributed search requests, etc., this setting controls whether the
  connections are made using IPv4 or IPv6.
* Connections to literal addresses are unaffected by this setting. For
  example, if a forwarder is configured to connect to "10.1.2.3" the
  connection is made over IPv4, regardless of what the value of this setting is.
* A value of "auto" means the following:
    * If 'listenOnIPv6' is set to "no", the Splunk server follows the
      "4-only" behavior.
    * If 'listenOnIPv6' is set to "yes", the Splunk server follows "6-first"
    * If 'listenOnIPv6' is set to "only", the Splunk server follow
      "6-only" behavior.
* A value of "4-first" means, if a host is available over both IPv4 and IPv6, then
  the Splunk server connects over IPv4 first and falls back to IPv6 if the
  IPv4 connection fails.
* A value of "6-first" means splunkd tries IPv6 first and falls back to
  IPv4 on failure.
* A value of "4-only" means splunkd only attempts to make connections over IPv4.
* A value of "6-only" means splunkd only attempts to connect to the IPv6 address.
* Default: auto (the Splunk server selects a reasonable
  value based on the listenOnIPv6 setting.)

guid = <globally unique identifier for this instance>
* This setting (as of version 5.0) belongs in the [general] stanza of
  SPLUNK_HOME/etc/instance.cfg file. See the .spec file of instance.cfg for
  more information.

useHTTPServerCompression = <boolean>
* Specifies whether the splunkd HTTP server should support gzip content
  encoding. For more info on how content encoding works, see Section 14.3
  of Request for Comments: 2616 (RFC2616) on the World Wide Web Consortium
  (W3C) website.
* Default: true

defaultHTTPServerCompressionLevel = <integer>
* If the useHTTPServerCompression setting is enabled (it is enabled
  by default), this setting controls the compression level that the
  Splunk server attempts to use.
* This number must be between 1 and 9.
* Higher numbers produce smaller compressed results but require more CPU
  usage.
* Default: 6 (This is appropriate for most environments)

skipHTTPCompressionAcl = <comma- or space-separated list>
* Lists a set of networks or addresses to skip data compression.
  These are addresses that are considered so close that network speed is
  never an issue, so any CPU time spent compressing a response is wasteful.
* Note that the server might still respond with compressed data if it
  already has a compressed version of the data available.
* These rules are separated by commas or spaces.
* The accepted formats for network and address rules are:
    1. A single IPv4 or IPv6 address (examples: "192.0.2.3", "2001:db8::2:1")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
```

```
       (examples: "192.0.2/24", "2001:DB8::/32")
    3. A DNS name. Use "*" as a wildcard.
       (examples: "myhost.example.com", "*.example.org")
    4. The wildcard "*" matches anything.
* Entries can also be prefixed with '!' to negate their meaning.
* Default: localhost addresses

legacyCiphers = decryptOnly|disabled
* This setting controls how Splunk software handles support for
  legacy encryption ciphers.
* If set to "decryptOnly", Splunk software supports decryption of
  configurations that have been encrypted with legacy ciphers.
  It encrypts all new configurations with newer and stronger cyphers.
* If set to "disabled", Splunk software neither encrypts nor decrypts
  configurations that have been encrypted with legacy ciphers.
* Default: decryptOnly

site = <string>
* Specifies the site that this Splunk instance belongs to when multisite is
  enabled.
* Valid values for site-id include site0 to site63
* The special value "site0" can be set only on search heads or on forwarders
  that are participating in indexer discovery.
  * For a search head, "site0" disables search affinity.
  * For a forwarder participating in indexer discovery, "site0" causes the
    forwarder to send data to all peer nodes across all sites.

useHTTPClientCompression = true|false|on-http|on-https
* Specifies whether gzip compression should be supported when splunkd acts
  as a client (including distributed searches). Note: For the content to
  be compressed, the HTTP server that the client is connecting to should
  also support compression.
* If the connection is being made over https and
  "useClientSSLCompression=true", then setting "useHTTPClientCompression=true"
  results in double compression work without much compression gain. To
  mitigate this, set this value to "on-http" (or to "true", and
  useClientSSLCompression to "false").
* Default: true

embedSecret = <string>
* When using report embedding, normally the generated URLs can only
  be used on the search head that they were generated on.
* If "embedSecret" is set, then the token in the URL is encrypted
  with this key.  Then other search heads with the exact same setting
  can also use the same URL.
* This is needed if you want to use report embedding across multiple
  nodes on a search head pool.

parallelIngestionPipelines = <integer>
* The number of discrete data ingestion pipeline sets to create for this
  instance.
* A pipeline set handles the processing of data, from receiving streams
  of events through event processing and writing the events to disk.
* An indexer that operates multiple pipeline sets can achieve improved
  performance with data parsing and disk writing, at the cost of additional
  CPU cores.
* For most installations, the default setting of "1" is optimal.
* Use caution when changing this setting. Increasing the CPU usage for data
  ingestion reduces available CPU cores for other tasks like searching.
* If the data source is streamed over TCP or UDP, such as syslog sources,
  only one pipeline will be used.
* NOTE: Enabling multiple ingestion pipelines can change the behavior of some
```

```
  settings in other configuration files. Each ingestion pipeline enforces
  the limits of the following settings independently:
    1. maxKBps (in the limits.conf file)
    2. max_fd (in the limits.conf file)
    3. maxHotBuckets (in the indexes.conf file)
    4. maxHotSpanSecs (in the indexes.conf file)
* Default: 1

pipelineSetSelectionPolicy = round_robin|weighted_random
* Specifies the pipeline set selection policy to use while selecting pipeline
  sets for new inputs.
* If set to round_robin, the incoming inputs are assigned to pipeline sets in a
  round robin fashion.
* If set to weighted_random, the incoming inputs are assigned to pipeline sets
  using a weighted random scheme designed to even out the CPU usage of each
  pipeline set.
* NOTE: This setting only takes effect when parallelIngestionPipelines is
  greater than 1.
* Default: round_robin

pipelineSetWeightsUpdatePeriod = <integer>
* The interval, in seconds, when pipeline set weights are recalculated for the
  weighted_random pipeline set selection policy.
* Reducing this interval causes pipeline set weights to be re-evaluated more
  frequently, thereby enabling the system to react more quickly to changes in
  dutycycle estimation.
* Increasing this interval causes pipeline set weights to be re-evaluated less
  frequently, thereby reducing the likelihood of the system responding to
  bursty events.
* Default: 30

pipelineSetNumTrackingPeriods = <integer>
* The number of look-back periods, of interval pipelineSetWeightsUpdatePeriod,
  that are used to keep track of incoming ingestion requests for pipeline sets.
* This information is used as a heuristic to calculate the pipeline set weights
  at every expiry of pipelineSetWeightsUpdatePeriod.
* Default: 5

pipelineSetChannelSetCacheSize = <integer>
* Maximum number of inactive channels to be stored in the per-pipeline set
  cache to reduce load in the configuration management system.
* Currently only affects ingestion via the HTTP Event Collector.
* Increasing this setting should reduce the number of created channels
  reported in metrics.log under the 'channel_cache' group. If neither that
  group nor the 'created' field exists in metrics.log, increasing this
  value has no effect.
* Default: 12

instanceType = <string>
* Should not be modified by users.
* Informs components (such as the Splunk Web Manager section) which
  environment the Splunk server is running in, to allow for more
  customized behaviors.
* Default: download

requireBootPassphrase = <boolean>
* Prompt the user for a boot passphrase when starting splunkd.
* Splunkd uses this passphrase to grant itself access to platform-provided
  secret storage facilities, like the GNOME keyring.
* For more information about secret storage, see the [secrets] stanza in
  $SPLUNK_HOME/etc/system/README/authentication.conf.spec.
* Default (if Common Criteria mode is enabled): true
```

```
* Default (if Common Criteria mode is disabled): false

numThreadsForIndexInitExecutor = <positive integer>
* Number of threads that can be used by the index init thread pool.
* Maximum accepted value for this setting is 32.
* Default: 16

remoteStorageRecreateIndexesInStandalone = <boolean>
* Controls re-creation of remote storage enabled indexes in standalone mode.
* Default: true

cleanRemoteStorageByDefault = <boolean>
* Allows 'splunk clean eventdata' to clean the remote indexes when set to true.
* Default: false

is_remote_queue_accounting_batched = <boolean>
* Allows indexer to maintain a batched count of events that have been uploaded to
  remote storage when set to true.
* This count is subsequently used to delete corresponding messages from remote queue.
* Default: false

recreate_index_fetch_bucket_batch_size = <positive integer>
* Controls the maximum number of bucket IDs to fetch from remote storage
  as part of a single transaction for a remote storage enabled index.
* Only valid for standalone mode.
* Default: 500

recreate_bucket_fetch_manifest_batch_size = <positive integer>
* Controls the maximum number of bucket manifests to fetch in parallel
  from remote storage.
* Only valid for standalone mode.
* Default: 100

splunkd_stop_timeout = <positive integer>
* The maximum time, in seconds, that splunkd waits for a graceful shutdown to
  complete before splunkd forces a stop.
* Default: 360 (6 minutes)

decommission_search_jobs_wait_secs = <unsigned integer>
* The maximum time, in seconds, that splunkd waits for running searches to complete
  during a shutdown_decommission_search.
* To trigger this type of shutdown, post to
  'services/server/control/shutdown_decommission_search'
* If set to 0, splunkd does not wait, and all searches in progress will fail.
* If this search head is a member of a search head cluster, use
  'decommission_search_jobs_wait_secs' in the [shclustering] stanza instead.
* NOTE: If this search head is a node of an indexer cluster, use
  'decommission_search_jobs_wait_secs' in the [clustering] stanza instead.
* Default: 0

decommission_search_jobs_min_wait_ratio = <decimal>
* Fraction of the decommission_search_jobs_wait_secs that splunkd will always
  wait during a shutdown_decommission_search.
* This wait is not contingent on whether or not there are any actively running searches
* Once this minimum wait time has elapsed, splunkd will wait the remainder of
  decommission_search_jobs_wait_secs contingent on the presence of actively running
  search processes on this indexer.
* Default: 0.15

python.version = python3|force_python3|unspecified
* For Python scripts only, sets the default Python version to use.
* Can be overridden by other 'python.version' values elsewhere, with the
```

```
  following exception:
* If you set to "force_python3", the system always uses Python 3, and ignores
  'python.version' values that you set elsewhere.
* If you set to "unspecified", the system calls the python interpreter 'python'
  to run scripts. Used on universal forwarders when calling an external instance
  of python. This setting value is not supported.
* Default: python3

roll_and_wait_for_uploads_at_shutdown_secs = <non-negative integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Default: 0 (disabled)

preShutdownCleanup = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies if indexer waits to complete any indexing activities before
  continuing with shutdown.
* Default: true

reset_manifests_on_startup = <boolean>
* Whether or not the Splunk platform instance regenerates size retention
  information for index bucket summaries that have been stored in the
  manifest.csv files.
* Configuring this setting lets the platform instance have the most
  up-to-date size retention information immediately after startup.
* When set to true, the size retention information for summaries stored
  in the manifest.csv files are removed and regenerated during startup.
* When set to false, manifest.csv files are not reset during startup.
* Default: true

percent_manifests_to_reset = <integer>
* In order to minimize the cost of resetting all manifest.csv files at once
  the manifest.csv files are separated in groups that are processed separately.
* This percentage defines how many manifest.csv files each group will reset.
* For example, a setting of 20 means each group resets 20% of all manifests
  resulting in 5 groups with 20% each.
* The minimum of one manifest.csv file will be processed per group.
* Legal values are between 0 and 100.
* Default: 10

regex_cache_hiwater = <integer>
* A threshold for the number of entries in the regex cache. If the regex cache
  grows larger than this, splunkd server will purge some of the older entries.
* When set to a negative value, no purge occurs, no matter how large
  the cache.
* Default: 2500

enable_search_process_long_lifespan = <boolean>
* Controls whether the search process can have a long lifespan.
* Configuring a long lifespan on a search process can optimize performance
  by reducing the number of new processes that are launched and old
  processes that are reaped, and is a more efficient use of system
  resources.
* When set to "true": Splunk software does the following:
  * Suppresses increases in the configuration generation. See the
    'conf_generation_include' setting for more information.
  * Avoids unnecessary replication of search configuration bundles.
  * Allows a certain number of idle search processes to live.
  * Sets the size of the pool of search processes.
  * Checks memory usage before a search process is reused.
* When set to "false": The lifespan of a search process at the 50th
```

```
  percentile is approximately 30 seconds.
* NOTE: Do not change this setting unless instructed to do so by Splunk Support.
* Default: true

conf_generation_include.<conf_file_name> = <boolean>
* Controls whether conf generation bumps at a property change in a particular
  type of *.conf file, mainly used on search head.
* In general, do not bump when a property change needs to restart Splunk server
  or is not related to search execution.
* If set properly, Splunk server skips unnecessary generation increments to
  maximize reuse of preforked search processes at search head. As a result,
  overall search performance is improved in shorter execution time and better
  system resource utilization.
* Has no effect if 'enable_search_process_long_lifespan' is set to "false".
* Default: false

encrypt_fields = <comma-separated list>
* A list of the fields that need to be re-encrypted when a search head
  cluster performs a first-time run on syncing all members with a new
  splunk.secret key, and when a bundle is created and applied in an indexer cluster.
* Provide each field as a three-element entry. Separate each field element with
  colons, and each field with commas, for example:
  <conf-file>:<stanza-prefix>:<setting>, <conf-file>:<stanza-prefix>:<setting>...
* Do not include brackets when you specify a stanza-prefix.
* To match all stanzas from a configuration file, leave the stanza-prefix
  empty. For example: "server: :pass4SymmKey" matches all stanzas
  with 'pass4SymmKey' as the key in the server.conf file.
* Default: a default list of fields containing passwords, secret keys, and identifiers:
  "server: :sslKeysfilePassword", "server: :sslPassword", "server: :pass4SymmKey",...
```

Configuration Change Tracker

```
[config_change_tracker]
disabled = <boolean>
* Whether or not splunkd writes configuration changes to the
  configuration change log at $SPLUNK_HOME/var/log/splunk/configuration_change.log.
* If set to "false", configuration changes are captured in
  $SPLUNK_HOME/var/log/splunk/configuration_change.log.
* If set to "true", configuration changes are not captured
  in $SPLUNK_HOME/var/log/splunk/configuration_change.log.
* Default: false

mode = [auto|diff|track-only]
* Determines the method used by 'config_change_tracker' to track and record
  changes to .conf files.
* A value of "auto" or "diff" means splunkd logs all configuration changes made to
  .conf files, including changes to setting values. In this mode, config change
  tracking only includes changes that could have an effect on your environment.
  For example, if a file with a stanza and setting-value pair is created, updated,
  or deleted, splunkd logs the change. But if an empty file or a stanza without any
  setting-value pairs is added or deleted, splunkd does not log the change since it
  will not have an impact. Similarly, splunkd does not track any comments that are
  added to or removed from files.
* A value of "track-only" means splunkd logs .conf file changes, but excludes
  configuration setting values. In this mode, config change tracking includes
  changes whether or not they can have an effect on your environment. For example,
  splunkd logs a change for any updates to file content, or that come from a change
  by the operating system. Splunkd also sees a comment that has been added to a .conf
  file as a change, because that change results in a different file checksum.
```

```
* Splunkd tracks all .conf files under the following directories:
  * $SPLUNK_HOME/etc/system
  * $SPLUNK_HOME/etc/apps
  * $SPLUNK_HOME/etc/users
  * $SPLUNK_HOME/etc/peer-apps
  It also tracks changes to the following:
  * $SPLUNK_HOME/etc/instance.cfg
* The values "auto" and "diff" have the same behavior at this time. Setting the
  value to "auto" ensures that the instance will always use the latest feature set.
* Default: auto

denylist = <regular expression>
* If set, splunkd does not monitor files for configuration change tracker if
  their path matches the specified regex.
* No default.

log_throttling_disabled = <boolean>
* Describes whether or not splunkd logs config changes to a .conf file
  that occur within the 'log_throttling_threshold_ms' time span as a single event.
* A value of "false" means that splunkd logs all changes to a conf file within
  the time span 'log_throttling_threshold_ms' as a single event.
* A value of "true" means that splunkd logs all changes individually as
  soon as it detects them.
* This setting requires a Linux system with the "inotify" API for
  file system event monitoring.
* Do not change this setting without first consulting with Splunk Support.
* Default: true

log_throttling_threshold_ms = <positive integer>
* The span of time, in milliseconds, during which splunkd logs multiple
  changes to a .conf file as a single configuration change event.
* If multiple changes are made to a conf file within the time span
  'log_throttling_threshold_ms' milliseconds, splunkd logs those changes
  as a single event.
* Default: 10000

exclude_fields = <comma-separated list>
* If set, splunkd excludes the stanza key that you specify when it writes to the
  configuration_change.log file.
* The format for each entry is '<conf-file>:<stanza>:<key>'. Separate multiple
  entries with commas.
* To exclude all keys under a stanza, use the '<conf-file>:<stanza>:*' format.
* This setting has no effect when mode is set to "track-only".
* Example setting:
  'server.conf:general:pass4SymmKey, authentication.conf:authentication:*'
* No default.

* NOTE: The [config_change_audit] stanza, which was previously mentioned in
  the Splunk version 8.2.0 documentation and configuration specification files,
  is now DEPRECATED.
```

Deployment Configuration details

```
[deployment]
pass4SymmKey = <passphrase string>
    * Authenticates traffic between the deployment server (DS) and its
      deployment clients (DCs).
    * By default, DS-DCs passphrase authentication key is disabled. To enable
```

```
      DS-DCs passphrase authentication, you must *also* add the following
      line to the [broker:broker] stanza in the restmap.conf file:
          requireAuthentication = true
    * If the key is not set in the [deployment] stanza, the key is looked
      for in the [general] stanza.
    * NOTE: Unencrypted passwords must not begin with "$1$", because this is
            used by Splunk software to determine if the password is already
            encrypted.

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' should be for a
  particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length than
  what you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* Default: 12
```

SSL/TLS Configuration details

```
[sslConfig]
* Set SSL for communications on Splunk back-end under this stanza name.
  * NOTE: To set SSL (for example HTTPS) for Splunk Web and the browser,
   use the web.conf file.
* Follow this stanza name with any number of the following setting/value
  pairs.
* If you do not specify an entry for each setting, the default value
  is used.

enableSplunkdSSL = <boolean>
* Enables/disables SSL on the splunkd management port (8089) and KV store
  port (8191).
* NOTE: Running splunkd without SSL is not recommended.
* Distributed search often performs better with SSL enabled.
* Default: true

useClientSSLCompression = <boolean>
* Turns on HTTP client compression.
* Server-side compression is turned on by default. Setting this on the
  client-side enables compression between server and client.
* Enabling this potentially gives you much faster distributed searches
  across multiple Splunk instances.
* CAUTION: There are known performance issues due to SSL compression.
  Confirm that 'conf_deploy_precompress_bundles',
  'precompress_cluster_bundle', 'precompress_artifacts',
  'preCompressKnowledgeBundlesClassicMode',
  'preCompressKnowledgeBundlesCascadeMode',
  and 'useHTTPClientCompression' are set to "false" before setting
 'useClientSSLCompression' to "true" to avoid double compression.
* Default: false

useSplunkdClientSSLCompression = <boolean>
* Controls whether SSL compression is used when splunkd is acting as
  an HTTP client, usually during certificate exchange, bundle replication,
  remote calls, etc.
* This setting is effective if, and only if, useClientSSLCompression
  is set to "true".
```

```
* NOTE: splunkd is not involved in data transfer in distributed search, the
  search in a separate process is.
* Default: true

sslVersions = <comma-separated list>
* Comma-separated list of SSL versions to support for incoming connections.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.
  The version "tls" selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version
  list but does nothing.
* When configured in FIPS mode, "ssl3" is always disabled regardless
  of this configuration.
* Default: The default can vary (see the 'sslVersions' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the
  current default)

sslVersionsForClient = <comma-separated list>
* A comma-separated list of SSL versions to support for outgoing HTTP connections
  from splunkd.  This includes distributed search, deployment client, etc.
* This is usually less critical, since SSL/TLS always picks the highest
  version both sides support.  However, you can use this setting to prohibit
  making connections to remote servers that only support older protocols.
* The syntax is the same as the 'sslVersions' setting.
* NOTE: For forwarder connections, there is a separate 'sslVersions'
  setting in the outputs.conf file. For connections to SAML servers, there
  is a separate 'sslVersions' setting in the authentication.conf file.
* Default: The default can vary (see the 'sslVersionsForClient' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the
  current default)

supportSSLV3Only = <boolean>
* DEPRECATED.  SSLv2 is disabled.  The exact set of SSL versions
  allowed is configurable using the 'sslVersions' setting.

sslVerifyServerCert = <boolean>
* This setting is used by distributed search and distributed
  deployment clients.
  * For distributed search: Used when making a search request
    to another server in the search cluster.
  * For distributed deployment clients: Used when polling a
    deployment server.
* A value of "true" means make sure that the connected server is
  authenticated. Both the common name and the alternate name
  of the server are checked for a match if they are specified
  in this configuration file. A certificate is considered
  verified if either is matched.
* Default: false

sslCommonNameToCheck = <commonName1>, <commonName2>, ...
* If set, and 'sslVerifyServerCert' is set to "true",
  splunkd limits most outbound HTTPS connections to hosts which
  use a certificate with one of the listed common names.
* The most important scenario is distributed search.
* Optional.
* No default (no common name checking.)

sslCommonNameList = <commonName1>, <commonName2>, ...
* DEPRECATED. Use the 'sslCommonNameToCheck' setting instead.

sslAltNameToCheck = <alternateName1>, <alternateName2>, ...
```

```
* If this value is set, and 'sslVerifyServerCert' is set to true,
  splunkd also verifies certificates which have a so-called
  "Subject Alternate Name" that matches any of the alternate
  names in this list.
  * Subject Alternate Names are effectively extended descriptive
    fields in SSL certificates beyond the commonName. A common
    practice for HTTPS certificates is to use these values to
    store additional valid hostnames or domains where the
    certificate should be considered valid.
* Accepts a comma-separated list of Subject Alternate Names to
  consider as valid.
* Items in this list are never validated against the SSL Common Name.
* Optional.
* No default (no alternate name checking.)

requireClientCert = <boolean>
* Requires that any HTTPS client that connects to a splunkd
  internal HTTPS server has a certificate that was signed by a
  CA (Certificate Authority) specified by the 'sslRootCAPath' setting.
  * Used by distributed search: Splunk indexing instances must be
    authenticated to connect to another splunk indexing instance.
  * Used by distributed deployment: The deployment server requires that
    deployment clients are authenticated before allowing them to poll
    for new configurations/applications.
* If set to "true", a client can connect ONLY if a certificate
  created by our certificate authority was used on that client.
* Default: false

sslVerifyServerName = <boolean>
* Whether or not splunkd, as a client, performs a TLS hostname validation check
  on an SSL certificate that it receives upon an initial connection
  to a server.
* A TLS hostname validation check ensures that a client
  communicates with the correct server, and has not been redirected to
  another by a machine-in-the-middle attack, where a malicious party inserts
  themselves between the client and the target server, and impersonates
  that server during the session.
* Specifically, the validation check forces splunkd to verify that either
  the Common Name or the Subject Alternate Name in the certificate that the
  server presents to the client matches the host name portion of the URL that
  the client used to connect to the server.
* For this setting to have any effect, the 'sslVerifyServerCert' setting must
  have a value of "true". If it doesn't, TLS hostname validation is not possible
  because certificate verification is not on.
* A value of "true" for this setting means that splunkd performs a TLS hostname
  validation check, in effect, verifying the server's name in the certificate.
  If that check fails, splunkd terminates the SSL handshake immediately. This
  terminates the connection between the client and the server. Splunkd logs
  this failure at the ERROR logging level.
* A value of "false" means that splunkd does not perform the TLS hostname
  validation check. If the server presents an otherwise valid certificate, the
  client-to-server connection proceeds normally.
* Default: false

caTrustStore = <[splunk],[OS]>
* The type of trust store that the Splunk platform accesses to validate
  connections over TLS.
* The Splunk platform uses this setting to load certificate authority
  certificates for this kind of validation.
* A value of "splunk" means the platform only uses the certificate authority
  certificates in the trust store that the 'sslRootCAPath' setting references.
* A value of "OS" means the platform only uses the CA certificates in
```

```
  the trust store that the operating system on the instance defines.
* Splunk provides support for OS trust store usage on the Linux
  operating system.
  There is currently no support for loading certificate trust stores on macOS
  or Windows.
* Providing both values ("splunk,OS") means that the platform uses CA
  certificates within both the Splunk platform and operating system
  trust stores.
* If a duplicate certificate exists in both types of trust store, the platform
  prioritizes using the certificate in the Splunk platform trust store.
* This values for this setting are not case sensitive.
* Default: splunk

caTrustStorePath = <string>
* The path to the location of the certificate authority trust store on
  a machine that runs a distribution of Linux.
* Different Linux distributions use different locations for the CA
  trust store. This setting lets you configure where the
  Splunk platform looks for the trust store, based on the distribution
  of Linux you run.
* If 'caTrustStore' has a value of "OS", but this setting has either
  no value or an invalid value, then the Splunk platform does not
  attempt to load any certificates from the OS trust store to
  validate TLS, and logs an error message in the splunkd.log log file.
* Following are example trust store locations for popular
  Linux distributions:
  Debian/Ubuntu/Gentoo: /etc/ssl/certs/ca-certificates.crt
  Fedora/RHEL 6:        /etc/pki/tls/certs/ca-bundle.crt
  CentOS/RHEL 7:        /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
* No default.

cipherSuite = <cipher suite string>
* If set, Splunk uses the specified cipher string for the HTTP server.
* If not set, Splunk uses the default cipher string provided by OpenSSL.
  This is used to ensure that the server does not accept connections using
  weak encryption protocols.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Default: The default can vary (See the 'cipherSuite' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the
  current default)

ecdhCurveName = <string>
* DEPRECATED.
* Use the 'ecdhCurves' setting instead.
* This setting specifies the Elliptic Curve Diffie-Hellman (ECDH) curve to
  use for ECDH key negotiation.
* Splunk only supports named curves that have been specified by their
  SHORT name.
* The list of valid named curves by their short and long names
  can be obtained by running this CLI command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Default: empty string.

ecdhCurves = <comma-separated list>
* A list of ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of an SSL Client Hello.
* The server supports only the curves specified in the list.
* Splunk software only supports named curves that have been specified
  by their SHORT names.
* The list of valid named curves by their short and long names can be obtained
  by running this CLI command:
```

```
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Example setting: "ecdhCurves = prime256v1,secp384r1,secp521r1"
* Default: The default can vary (See the 'ecdhCurves' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the
  current default)

serverCert = <path>
* The full path to the PEM (Privacy-Enhanced Mail) format server
  certificate file.
* Certificates are auto-generated by splunkd on starting Splunk Enterprise.
* You can replace the default certificate with your own PEM
  format file.
* Default: $SPLUNK_HOME/etc/auth/server.pem

sslKeysfile = <filename>
* DEPRECATED. Use the 'serverCert' setting instead.
* This file is in the directory specified by the 'caPath' setting
  (see below).
* Default: server.pem

sslPassword = <string>
* Server certificate password.
* Default: password

sslKeysfilePassword = <string>
* DEPRECATED. Use the 'sslPassword' setting instead.

sslRootCAPath = <path>
* Full path to the root CA (Certificate Authority) certificate store
  on the operating system.
* The <path> must refer to a PEM (Privacy-Enhanced Mail) format
  file containing one or more root CA certificates concatenated
  together.
* Required for Common Criteria.
* This setting is valid on Windows machines only if you have not set
  'sslRootCAPathHonoredOnWindows' to "false".
* No default.

sslRootCAPathHonoredOnWindows = <boolean>
* DEPRECATED.
* Whether or not the Splunk instance respects the 'sslRootCAPath' setting on
  Windows machines.
* If you set this setting to "false", then the instance does not respect the
  'sslRootCAPath' setting on Windows machines.
* This setting is valid only on Windows, and only if you have set
  'sslRootCAPath'.
* When the 'sslRootCAPath' setting is respected, the instance expects to find
  a valid PEM file with valid root certificates that are referenced by that
  path. If a valid file is not present, SSL communication fails.
* Default: true

caCertFile = <filename>
* DEPRECATED. Use the 'sslRootCAPath' setting instead.
* Used only if 'sslRootCAPath' is not set.
* File name (relative to 'caPath') of the CA (Certificate Authority)
  certificate PEM format file containing one or more certificates
  concatenated together.
* Default: cacert.pem

dhFile = <path>
* PEM (Privacy-Enhanced Mail) format Diffie-Hellman(DH) parameter file name.
* DH group size should be no less than 2048bits.
```

```
* This file is required in order to enable any Diffie-Hellman ciphers.
* No default.

caPath = <path>
* DEPRECATED. Use absolute paths for all certificate files.
* If certificate files given by other settings in this stanza are not absolute
  paths, then they are relative to this path.
* Default: $SPLUNK_HOME/etc/auth

certCreateScript = <script name>
* Creation script for generating certificates on startup of Splunk Enterprise.

sendStrictTransportSecurityHeader = <boolean>
* If set to "true", the REST interface sends a "Strict-Transport-Security"
  header with all responses to requests made over SSL.
* This can help avoid a client being tricked later by a
  Man-In-The-Middle attack to accept a non-SSL request.
  However, this requires a commitment that no non-SSL web hosts
  ever run on this hostname on any port. For
  example, if Splunk Web is in default non-SSL mode this can break the
  ability of a browser to connect to it.
* NOTE: Enable with caution.
* Default: false

allowSslCompression = <boolean>
* If set to "true", the server allows clients to negotiate
  SSL-layer data compression.
* KV Store also observes this setting.
* If set to "false", KV Store disables TLS compression.
* Default: true

allowSslRenegotiation = <boolean>
* In the SSL protocol, a client may request renegotiation of the
  connection settings from time to time.
* If set to "false", causes the server to reject all renegotiation
  attempts, breaking the connection.  This limits the amount of CPU a
  single TCP connection can use, but it can cause connectivity problems
  especially for long-lived connections.
* Default: true

sslClientSessionPath = <path>
* Path where all client sessions are stored for session re-use.
* Used if 'useSslClientSessionCache' is set to "true".
* No default.

useSslClientSessionCache = <boolean>
* Specifies whether to re-use client session.
* When set to "true", client sessions are stored in memory for
  session re-use. This reduces handshake time, latency and
  computation time to improve SSL performance.
* When set to "false", each SSL connection performs a full
  SSL handshake.
* Default: false

sslServerSessionTimeout = <integer>
* Timeout, in seconds, for newly created session.
* If set to "0", disables Server side session cache.
* The openssl default is 300 seconds.
* Default: 300 (5 minutes)

sslServerHandshakeTimeout = <integer>
* The timeout, in seconds, for an SSL handshake to complete between an
```

```
  SSL client and the Splunk SSL server.
* If the SSL server does not receive a "Client Hello" from the SSL client within
  'sslServerHandshakeTimeout' seconds, the server terminates
  the connection.
* Default: 60

certificateStatusValidationMethod = crl
* Specifies the certificate status validation method that splunkd is to use.
* Certificate status validation checks the status of a digital certificate
  upon its presentation during a network connection.
* When certificate status validation is active, it is active for any kind of
  Splunk platform-related network communication that uses SSL, including
  the Splunk Web, splunkd, and Splunk-to-Splunk (S2S) communication channels.
* Currently, the only acceptable value for this setting is "crl".
  * If you do not give this setting a value of "crl", the setting, and thus
    certificate status validation checks, are turned off.
  * "crl" stands for Certificate Revocation List, which is a list of
    digital certificates that have been revoked by the issuing Certificate
    Authority (CA) before their scheduled expiration date and can no
    longer be trusted.
  * The default path for CRL files in a Splunk platform instance is
    at $SPLUNK_HOME/auth/crl. Any CRL files must reside there, in
    privacy-enhanced mail (PEM) format.
* For more information on using CRLs and configuring CRL files on a Splunk
  platform instance, see the '[kvstore]:sslCRLPath' setting.
* Default: empty string (certificate status validation checks are off)

cliVerifyServerName = <boolean>
* Whether or not the Splunk CLI must validate the server name in the splunkd
  server certificate when you use the "--uri" argument to connect to a remote
  splunkd server.
* Server certificates are validated to be issued to the same host name or IP address
  which is specified in the command line argument.
* A value of "true" means that the CLI validates server certificates. The
  certificates must be issued to have the same host or IP address that you
  specify in the command line argument to connect to the server.
* When this setting is "true", you can temporarily disable enforcement for that
  particular invocation of the Splunk CLI by providing the "--no-server-name-check"
  flag on the command line.
* The CLI uses the certificate authority certificate or certificate chain that you specify in
  the server.conf file, at '[sslConfig]/sslRootCAPath', to verify incoming server certificates.
* A value of "false" means that the CLI does not validate server certificates.
* Default: false
```

Python SSL Client Configuration details

```
[pythonSslClientConfig]
* SSL settings for Splunk Python client connections.
* Follow this stanza name with any number of the following setting/value
  pairs.
* If you do not specify an entry for each setting, splunkd uses
  the values from the settings in the [sslConfig] stanza.

sslVerifyServerCert = <boolean>
* See the description of 'sslVerifyServerCert' under the [sslConfig] stanza
  for details on this setting.
* If you give this setting a value of "true", confirm that you have also set
  '[sslConfig]/sslRootCAPath' with the correct value.
* Default: false
```

```
sslVerifyServerName = <boolean>
* See the description of 'sslVerifyServerName' under the [sslConfig] stanza
  for details on this setting.
* Default: false
```

Splunkd http proxy configuration

```
[proxyConfig]
* NOTE: Splunkd does not support Transport Layer Security (TLS) as a protocol
  for proxying connections. It only supports using the HTTP CONNECT method
  for HTTPS requests. The proxy server cannot listen on an SSL port.

http_proxy = <string>
* If set, splunkd sends all HTTP requests through the proxy server
  that you specify.
* No default.

https_proxy = <string>
* If set, splunkd sends all HTTPS requests through the proxy server
  that you specify.
* If not set, splunkd uses the 'http_proxy' setting instead.
* No default.

proxy_rules = <string>
* One or more host names or IP addresses for which splunkd should route
  HTTPS requests only through the proxy server.
* If set, splunkd uses the proxy server only for endpoints that match the
  hosts or IP addresses in this value.
* Splunkd does not route requests to either the localhost or loopback addresses
  through the proxy server.
* Separate multiple entries with commas.
* This setting accepts the following values:
  * '*' (asterisk): Proxy all requests. This is the only wildcard, and it can
    be used only by itself.
  * <IPv4 or IPv6 address>: Route the request through the proxy if the
     request is intended for that address.
  * <hostname>/<domain name>: Route the request through the proxy if
     the request is intended for that host name or domain name.
  * Examples:
    * proxy_rules = "wimpy": This matches the host name "wimpy".
    * proxy_rules = "splunk.com": Matches all host names in the splunk.com
       domain (such as apps.splunk.com, www.splunk.com, etc.)
* Default: *

no_proxy = <string>
* One or more host names or IP addresses for which splunkd should
  explicitly bypass the proxy server for HTTPS requests.
* If set, splunkd does not route requests to matching host names and
  IP addresses through the proxy server.
* This setting overrides the 'proxy_rules' setting. If a host name or IP
  address is in both settings, splunkd does not route requests for that
  host name or IP address through the proxy server.
* Splunkd does not route requests to either the localhost or loopback addresses
  through the proxy server.
  addresses.
* Separate multiple entries with commas.
* This setting accepts the following values:
  * '*' (asterisk): Proxy all requests. This is the only wildcard, and it can
    be used only by itself.
```

```
  * <IPv4 or IPv6 address>: Route the request through the proxy if the
     request is intended for that address.
  * <hostname>/<domain name>: Route the request through the proxy if
     the request is intended for that host name or domain name.
  * Examples:
    * no_proxy = "wimpy": This matches the host name "wimpy".
    * no_proxy = "splunk.com": Matches all host names in the splunk.com
       domain (such as apps.splunk.com, www.splunk.com, etc.)
* Default: localhost, 127.0.0.1, ::1
```

Splunkd HTTP server configuration

```
[httpServer]
* Set stand-alone HTTP settings for splunkd under this stanza name.
* Follow this stanza name with any number of the following setting/value
  pairs.
* If you do not specify an entry for each setting, splunkd uses the default
  value.

atomFeedStylesheet = <string>
* Defines the stylesheet relative URL to apply to default Atom feeds.
* Set to 'none' to stop writing out xsl-stylesheet directive.
* Default: /static/atom.xsl

max-age = <nonnegative integer>
* Set the maximum time, in seconds, to cache a static asset served off of
  the '/static' directory.
* This value is passed along in the 'Cache-Control' HTTP header.
* Default: 3600 (60 minutes)

follow-symlinks = <boolean>
* Specifies whether the static file handler (serving the '/static'
  directory) follows filesystem symlinks when serving files.
* Default: false

disableDefaultPort = <boolean>
* If set to "true", turns off listening on the splunkd management port,
  which is 8089 by default.
* NOTE: Changing this setting is not recommended.
  * This is the general communication path to splunkd.  If it is disabled,
    there is no way to communicate with a running splunk instance.
  * This means many command line splunk invocations cannot function,
    Splunk Web cannot function, the REST interface cannot function, etc.
  * If you choose to disable the port anyway, understand that you are
    selecting reduced Splunk functionality.
* Default: false

mgmtMode = none|auto|tcp
* Sets the transport layer protocol mode for Splunk CLI management commands.
* A value of "none" means that only Splunk CLI commands that can be run
  offline are available for use on the instance.
* A value of "auto" means that CLI commands execute over a Unix Domain Socket (UDS),
  which represents as $SPLUNK_HOME/var/run/splunk/cli.socket on the file system.
  * If the OS does not support UDS, and if 'disableDefaultPort' has a value of "false",
    the CLI executes over the splunkd management port.
  * If 'disableDefaultPort' has a value of "true", only CLI commands that can be
    run offline are available for use on the instance.
* A value of "tcp" means the CLI commands execute over the splunkd
  management port.
```

```
* This setting is only available on the universal forwarder.
* NOTE: There is a path length limit of 104-108 characters for the UDS socket file. This includes
  whatever the $SPLUNK_HOME environment variable expands to. If you exceed this length for the
  socket file, UDS does not work, and you must reinstall the universal forwarder, because you cannot
  take corrective action to fix the UDS path. Verify the UDS path length when configuring this setting.
* Default: auto

acceptFrom = <network_acl> ...
* Lists a set of networks or addresses from which to accept connections.
* Separate multiple rules with commas or spaces.
* Each rule can be in one of the following formats:
    1. A single IPv4 or IPv6 address (examples: "10.1.2.3", "fe80::4a3")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
       (examples: "10/8", "192.168.1/24", "fe80:1234/32")
    3. A DNS name, possibly with a "*" used as a wildcard
       (examples: "myhost.example.com", "*.splunk.com")
    4. "*", which matches anything
* You can also prefix an entry with '!' to cause the rule to reject the
  connection. The input applies rules in order, and uses the first one that
  matches.
  For example, "!10.1/16, *" allows connections from everywhere except
  the 10.1.*.* network.
* Default: "*" (accept from anywhere)

streamInWriteTimeout = <positive number>
* The timeout, in seconds, for uploading data to the http server.
* When uploading data to http server, if the http server is unable
  to write data to the receiver for the specified value, the operation
  aborts.
* Default: 5

max_content_length = <integer>
* Maximum content length, in bytes.
* HTTP requests over the size specified are rejected.
* This setting exists to avoid allocating an unreasonable amount
  of memory from web requests.
* In environments where indexers have enormous amounts of RAM, this
  number can be reasonably increased to handle large quantities of
  bundle data.
* Default: 2147483648 (2GB)

maxSockets = <integer>
* The number of simultaneous HTTP connections that Splunk Enterprise accepts
  simultaneously. You can limit this number to constrain resource usage.
* If set to "0", Splunk Enterprise automatically sets maxSockets to
  one third of the maximum allowable open files on the host.
* If this number is less than 50, it is set to 50.
* If this number is greater than 400000, it is set to 400000.
* If set to a negative number, no limit is enforced.
* Default: 0

maxThreads = <integer>
* The number of threads that can be used by active HTTP transactions.
  You can limit this number to constrain resource usage.
* If set to 0, Splunk Enterprise automatically sets the limit to
  one third of the maximum allowable threads on the host.
* If this number is less than 20, it is set to 20. If this number is
  greater than 150000, it is set to 150000.
* If maxSockets is not negative and maxThreads is greater than maxSockets, then
  Splunk Enterprise sets maxThreads to be equal to maxSockets.
* If set to a negative number, no limit is enforced.
* Default: 0
```

```
keepAliveIdleTimeout = <integer>
* How long, in seconds, that the Splunkd HTTP server allows a keep-alive
  connection to remain idle before forcibly disconnecting it.
* If this number is less than 7200, it is set to 7200.
* Default: 7200 (120 minutes)

busyKeepAliveIdleTimeout = <integer>
* How long, in seconds, that the Splunkd HTTP server allows a keep-alive
  connection to remain idle while in a busy state before forcibly
  disconnecting it.
* Use caution when configuring this setting as a value that is too large
  can result in file descriptor exhaustion due to idling connections.
* If this number is less than 12, it is set to 12.
* Default: 12

forceHttp10 = auto|never|always
* When set to "always", the REST HTTP server does not use some
  HTTP 1.1 features such as persistent connections or chunked
  transfer encoding.
* When set to "auto" it does this only if the client sent no
  User-Agent header, or if the user agent is known to have bugs
  in its HTTP/1.1 support.
* When set to "never" it always allows HTTP 1.1, even to
  clients it suspects may be buggy.
* Default: auto

crossOriginSharingPolicy = <origin_acl> ...
* List of the HTTP Origins for which to return Access-Control-Allow-* (CORS)
  headers.
* These headers tell browsers that web applications are trusted at those sites
  to make requests to the REST interface.
* The origin is passed as a URL without a path component (for example
  "https://app.example.com:8000").
* This setting can take a list of acceptable origins, separated
  by spaces and/or commas.
* Each origin can also contain wildcards for any part.  Examples:
    *://app.example.com:*  (either HTTP or HTTPS on any port)
    https://*.example.com  (any host under example.com, including
    example.com itself)
* An address can be prefixed with a '!' to negate the match, with
  the first matching origin taking precedence.  For example,
  "!*://evil.example.com:* *://*.example.com:*" to not avoid
  matching one host in a domain
* A single "*" can also be used to match all origins
* No default.

crossOriginSharingHeaders = <string>
* A list of the HTTP headers to which splunkd sets
  "Access-Control-Allow-Headers" when replying to
  Cross-Origin Resource Sharing (CORS) preflight requests.
* The "Access-Control-Allow-Headers" header is used in response to
  a CORS preflight request to tell browsers which HTTP headers can be
  used during the actual request.
* A CORS preflight request is a CORS request that checks to see if
  the CORS protocol is understood and a server is aware of using
  specific methods and headers.
* This setting can take a list of acceptable HTTP headers, separated
  by commas.
* A single "*" can also be used to match all headers.
* Default: Empty string.
```

```
x_frame_options_sameorigin = <boolean>
* Adds a X-Frame-Options header set to "SAMEORIGIN" to every response
  served by splunkd.
* Default: true

allowEmbedTokenAuth = <boolean>
* A value of "false" means splunkd does not allow any access to artifacts
  that previously had been explicitly shared to anonymous users.
* This effectively disables all use of the "embed" feature.
* Default: true

cliLoginBanner = <string>
* Sets a message which is added to the HTTP reply headers
  of requests for authentication, and to the "server/info" endpoint
* This is printed by the Splunk CLI before it prompts
  for authentication credentials.  This can be used to print
  access policy information.
* If this string starts with a '"' character, it is treated as a
  CSV-style list with each line comprising a line of the message.
  For example: "Line 1","Line 2","Line 3"
* No default.

allowBasicAuth = <boolean>
* Allows clients to make authenticated requests to the splunk
  server using "HTTP Basic" authentication in addition to the
  normal "authtoken" system
* This is useful for programmatic access to REST endpoints and
  for accessing the REST API from a web browser.  It is not
  required for the UI or CLI.
* Default: true

allowWwwAuthHeader = <boolean>
* Describes whether or not Splunk Web can include a "www-authenticate" header
  in a response to a request from a web client to access a management endpoint.
* When Splunk Web sends the "www-authenticate" header in response to such
  a request, the client forces its user to provide credentials to authenticate.
* A value of "true" means that Splunk Web sends a "www-authenticate" header
  in its response to the web client request. This means that the user of that
  client will be prompted to enter valid credentials to access the instance,
  even if they provide those credentials as part of the request.
* A value of "false" means that Splunk Web does not send the "www-authenticate"
  header in its response to the web client request. This means that the
  user of that client will not be prompted to provide valid credentials to
  access the instance.
* Giving this setting a value of "false" reduces the attack surface in the
  management API when you access it through Splunk Web.
* This setting is not valid for the CLI. It works only with Splunk Web.
* Default: true

basicAuthRealm = <string>
* When using "HTTP Basic" authentication, the 'realm' is a
  human-readable string describing the server.  Typically, a web
  browser presents this string as part of its dialog box when
  asking for the username and password.
* This can be used to display a short message describing the
  server and/or its access policy.
* Default: /splunk

allowCookieAuth = <boolean>
* Allows clients to request an HTTP cookie from the /services/auth/login
  endpoint which can then be used to authenticate future requests
* Default: true
```

```
cookieAuthHttpOnly = <boolean>
* When using cookie based authentication, mark returned cookies
  with the "httponly" flag to tell the client not to allow javascript
  code to access its value
* NOTE: has no effect if allowCookieAuth=false
* Default: true

cookieSameSiteSecure = <boolean>
* DEPRECATED.
* Describes whether or not the Splunk REST server sets all Splunk cookies
  with the "SameSite=None" cookie attribute.
* A value of "true" means that the Splunk REST server sets the "SameSite=None"
  attribute for all cookies.
* A value of "false" means that the REST server does not set cookies with
  the "SameSite=None" attribute.
* NOTE: The REST server does not change the 'secure' cookie attribute with
  this setting. Use the 'cookieAuthSecure' setting to perform this task.
  The Splunk web server and the Splunk REST server use different
  settings to make cookie modifications. To modify cookies that the
  Splunk web server sets, use the 'cookieSameSite' setting in the
  web.conf configuration file.
* Default: false

cookieAuthSecure = <boolean>
* When using cookie based authentication, mark returned cookies
  with the "secure" flag to tell the client never to send it over
  an unencrypted HTTP channel
* NOTE: has no effect if allowCookieAuth=false OR the splunkd REST
  interface has SSL disabled
* Default: true

dedicatedIoThreads = [<integer>|auto]
* The number of threads that splunkd dedicates to handling HTTP I/O requests.
* This setting controls thread usage for all HTTP requests through splunkd,
  including SSL encryption.
* If you set this to "0", splunkd uses the same thread that accepted the initial
  connection over TCP to perform the HTTP I/O.
* If you set this to a number other than "0", splunkd creates that number of
  threads to handle HTTP I/O.
* If you set this to "auto", splunkd uses the number of CPU cores on the
  machine to determine the number of threads available for HTTP I/O as
  follows:
    * Number of CPU cores available  | 'dedicatedIoThreads'
                  0 - 16             |    0
                 17 - 48             |    2
                 49 - 128            |    4
                129 - 192            |    6
                193 and higher       |    8

* You do not usually need to change this setting. However, for an instance
  running as a dedicated deployment server, the best practice is to set
  'dedicatedIoThreads' to a value of CPU cores/2, or CPU cores-1, in the case
  of a very active deployment server. For example, in the case of a deployment
  server with 8 CPU cores, set 'dedicatedIoThreads' to 4 or 7.
* Default: auto

dedicatedIoThreadsSelectionPolicy = <round_robin | weighted_random>
* Specifies the I/O threads selection policy to use while selecting I/O thread
  for new connection.
* If set to "round_robin", the incoming connections are assigned to I/O threads
  in a round robin fashion.
```

```
* If set to "weighted_random", the connections are assigned to I/O threads using
  a weighted random scheme designed to even out the CPU usage of each I/O thread.
* NOTE: This setting only takes effect when dedicatedIoThreads is greater than 1.
* Default: round_robin

dedicatedIoThreadsWeightsUpdatePeriod = <number>
* The interval, in seconds, when I/O thread weights are recalculated for the
  "weighted_random" selection policy.
* Reducing this interval causes the weights to be re-evaluated more
  frequently, thereby enabling the system to react more quickly to changes
  in relative thread load.
* Increasing this interval causes the weights to be re-evaluated less
  frequently, thereby reducing the ability of the system to respond to
  bursty events.
* Default: 30

replyHeader.<name> = <string>
* Add a static header to all HTTP responses this server generates
* For example, "replyHeader.My-Header = value" causes the
  response header "My-Header: value" to be included in the reply to
  every HTTP request to the REST server
```

Splunkd HTTPServer listener configuration

```
[httpServerListener:<ip:><port>]
* Enable the splunkd REST HTTP server to listen on an additional port number
  specified by <port>.  If a non-empty <ip> is included (for example:
  "[httpServerListener:127.0.0.1:8090]") the listening port is
  bound only to a specific interface.
* Multiple "httpServerListener" stanzas can be specified to listen on
  more ports.
* Normally, splunkd listens only on the single REST port specified in
  the web.conf "mgmtHostPort" setting, and none of these stanzas need to
  be present. Add these stanzas only if you want the REST HTTP server
  to listen to more than one port.

ssl = <boolean>
* Toggle whether this listening ip:port uses SSL or not.
* If the main REST port is SSL (the "enableSplunkdSSL" setting in this
  file's [sslConfig] stanza) and this stanza is set to "ssl=false" then
  clients on the local machine such as the CLI may connect to this port.
* Default: true

listenOnIPv6 = no|yes|only
* Toggle whether this listening ip:port listens on IPv4, IPv6, or both.
* If not present, the setting in the [general] stanza is used

acceptFrom = <network_acl> ...
* Lists a set of networks or addresses from which to accept connections.
* Separate multiple rules with commas or spaces.
* Each rule can be in one of the following formats:
    1. A single IPv4 or IPv6 address (examples: "10.1.2.3", "fe80::4a3")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
       (examples: "10/8", "192.168.1/24", "fe80:1234/32")
    3. A DNS name, possibly with a "*" used as a wildcard
       (examples: "myhost.example.com", "*.splunk.com")
    4. "*", which matches anything
* You can also prefix an entry with '!' to cause the rule to reject the
  connection. The input applies rules in order, and uses the first one that
```

```
  matches.
  For example, "!10.1/16, *" allows connections from everywhere except
  the 10.1.*.* network.
* Default: The setting in the [httpServer] stanza
```

Static file handler MIME-type map

```
[mimetype-extension-map]
* Map filename extensions to MIME type for files served from the static file
  handler under this stanza name.

<file-extension> = <MIME-type>
* Instructs the HTTP static file server to mark any files ending
  in 'file-extension' with a header of 'Content-Type: <MIME-type>'.
* Default:
    [mimetype-extension-map]
    gif = image/gif
    htm = text/html
    jpg = image/jpg
    png = image/png
    txt = text/plain
    xml = text/xml
    xsl = text/xml
```

Log rotation of splunkd_stderr.log & splunkd_stdout.log

```
# These stanzas apply only on UNIX.  splunkd on Windows has no
# stdout.log or stderr.log files.

[stderr_log_rotation]
* Controls the data retention of the file containing all messages written to
  splunkd's stderr file descriptor (fd 2).
* Typically this is extremely small, or mostly errors and warnings from
  linked libraries.

maxFileSize = <bytes>
* When splunkd_stderr.log grows larger than this value, it is rotated.
* maxFileSize is expressed in bytes.
* You might want to increase this if you are working on a problem
  that involves large amounts of output to the splunkd_stderr.log file.
* You might want to reduce this to allocate less storage to this log category.
* Default: 10000000 (10 si-megabytes)

BackupIndex = <non-negative integer>
* How many rolled copies to keep.
* For example, if this setting is 2, the splunkd_stderr.log.1 and
  splunkd_stderr.log.2 file might exist. Further rolls delete the
  current splunkd_stderr.log.2 file.
* You might want to increase this value if you are working on a problem
  that involves large amounts of output to the splunkd_stderr.log fils
* You might want to reduce this to allocate less storage to this log category.
* Default: 2

checkFrequency = <seconds>
* How often. in seconds, to check the size of splunkd_stderr.log
```

```
* Larger values may result in larger rolled file sizes but take less resources.
* Smaller values may take more resources but more accurately constrain the
  file size.
* Default: 10

[stdout_log_rotation]
* Controls the data retention of the file containing all messages written to
  splunkd's stdout file descriptor (fd 1).
* Almost always, there is nothing in this file.

* This stanza can have the same settings as the [stderr_log_rotation]
  stanza with the same defaults.  See above for definitions.

maxFileSize = <bytes>
BackupIndex = <non-negative integer>
checkFrequency = <seconds>
```

Remote applications configuration (e.g. SplunkBase)

```
[applicationsManagement]
* Set remote applications settings for Splunk under this stanza name.
* Follow this stanza name with any number of the following setting/value
  pairs.
* If you do not specify an entry for each setting, the Splunk platform instance
  uses the default value.

allowInternetAccess = <boolean>
* Lets the Splunk platform instance access the remote applications repository.

url = <string>
* Applications repository URL.
* Default: https://apps.splunk.com/api/apps

loginUrl = <string>
* Applications repository login URL.
* Default: https://apps.splunk.com/api/account:login/

detailsUrl = <string>
* Base URL for application information, keyed off of app ID.
* Default: https://apps.splunk.com/apps/id

useragent = <splunk-version>-<splunk-build-num>-<platform>
* User-agent string to use when contacting applications repository.
* <platform> includes information like operating system and CPU architecture.

updateHost = <string>
* Host section of URL to check for app updates, e.g. https://apps.splunk.com

updatePath = <string>
* Path section of URL to check for app updates
  For example: /api/apps:resolve/checkforupgrade

updateTimeout = <time range string>
* The minimum amount of time Splunk software waits between checks for
  app updates.
* Examples include '24h' (24 hours), '3d' (3 days),
  '7200s' (7200 seconds, or two hours)
* Default: 24h
```

```
sslVersions = <comma-separated list>
* Comma-separated list of SSL versions to connect to 'url'
  (https://apps.splunk.com).
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list
  but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Default: The default can vary (See the 'sslVersions' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the
  current default)

sslVerifyServerCert = <boolean>
* If this is set to true, Splunk verifies that the remote server (
  specified in 'url') being connected to is a valid one (authenticated).
  Both the common name and the alternate name of the server are then
  checked for a match if they are specified in 'sslCommonNameToCheck' and
  'sslAltNameToCheck'. A certificate is considered verified if either
  is matched.
* Default: true

sslVerifyServerName = <boolean>
* See the description of 'sslVerifyServerName' under the [sslConfig] stanza
  for details on this setting.
* Default: false

caCertFile = <path>
* The full path to a CA (Certificate Authority) certificate(s) PEM format file.
* The <path> must refer to a PEM format file containing one or more root CA
  certificates concatenated together.
* Used only if 'sslRootCAPath' is not set.
* Used for validating SSL certificate from https://apps.splunk.com/

caTrustStore = <[splunk],[OS]>
* See the description of 'caTrustStore' under the [sslConfig] stanza
  for details on this setting.
* Default: splunk

caTrustStorePath = <string>
* See the description of 'caTrustStorePath' under the [sslConfig] stanza
  for details on this setting.
* No default.

sslCommonNameToCheck = <commonName1>, <commonName2>, ...
* If this value is set, and 'sslVerifyServerCert' is set to true,
  splunkd checks the common name(s) of the certificate presented by
  the remote server (specified in 'url') against this list of common names.
* Default: splunkbase.splunk.com, apps.splunk.com, cdn.apps.splunk.com

sslCommonNameList = <commonName1>, <commonName2>, ...
* DEPRECATED. Use the 'sslCommonNameToCheck' setting instead.

sslAltNameToCheck =  <alternateName1>, <alternateName2>, ...
* If this value is set, and 'sslVerifyServerCert' is set to true,
  splunkd checks the alternate name(s) of the certificate presented by
  the remote server (specified in 'url') against this list of subject
  alternate names.
* Default: splunkbase.splunk.com, apps.splunk.com, cdn.apps.splunk.com
```

```
cipherSuite = <cipher suite string>
* Uses the specified cipher string for making outbound HTTPS connection.
* The default can vary. See the 'cipherSuite' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the current default.

ecdhCurves = <comma separated list of ec curves>
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only supports named curves specified
  by their SHORT names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: The default can vary (See the 'ecdhCurves' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the
  current default)
```

Misc. configuration

```
[scripts]

initialNumberOfScriptProcesses = <num>
* The number of pre-forked script processes that are launched when the
  system comes up. These scripts are reused when script REST endpoints
  *and* search scripts are executed.
  The idea is to eliminate the performance overhead of launching the script
  interpreter every time it is invoked.  These processes are put in a pool.
  If the pool is completely busy when a script gets invoked, a new processes
  is fired up to handle the new invocation - but it disappears when that
  invocation is finished.
```

Disk usage settings (for the indexer, not for Splunk log files)

```
[diskUsage]

minFreeSpace = <num>|<percentage>
* Minimum free space for a partition.
* Specified as an integer that represents a size in binary
  megabytes (ie MiB) or as a percentage, written as a decimal
  between 0 and 100 followed by a '%' sign, for example "10%"
  or "10.5%"
* If specified as a percentage, this is taken to be a percentage of
  the size of the partition. Therefore, the absolute free space required
  varies for each partition depending on the size of that partition.
* Specifies a safe amount of space that must exist for splunkd to continue
  operating.
* Note that this affects search and indexing
* For search:
  * Before attempting to launch a search, Splunk software requires this
    amount of free space on the filesystem where the dispatch directory
    is stored, $SPLUNK_HOME/var/run/splunk/dispatch
```

```
  * Applied similarly to the search quota values in authorize.conf and
    limits.conf.
* For indexing:
  * Periodically, the indexer checks space on all partitions
    that contain splunk indexes as specified by indexes.conf. Indexing
    is paused and a ui banner + splunkd warning posted to indicate
    need to clear more disk space.
* Default: 5000 (approx 5GB)

pollingFrequency = <num>
* Specifies that after every 'pollingFrequency' events are indexed,
  the disk usage is checked.
* Default: 100000

pollingTimerFrequency = <num>
* Minimum time, in seconds, between two disk usage checks.
* Default: 10
```

Queue settings

```
[queue]

maxSize = [<integer>|<integer>[KB|MB|GB]]
* Specifies default capacity of a queue.
* If specified as a lone integer (for example, maxSize=1000), maxSize
  indicates the maximum number of events allowed in the queue.
* If specified as an integer followed by KB, MB, or GB (for example,
  maxSize=100MB), it indicates the maximum RAM allocated for queue.
* Default: 500KB

cntr_1_lookback_time = [<integer>[s|m]]
* The lookback counters are used to track the size and count (number of
  elements in the queue) variation of the queues using an exponentially
  moving weighted average technique. Both size and count variation
  has 3 sets of counters each. The set of 3 counters is provided to be able
  to track short, medium and long term history of size/count variation. The
  user can customize the value of these counters or lookback time.
* Specifies how far into history should the size/count variation be tracked
  for counter 1.
* It must be an integer followed by [s|m] which stands for seconds and
  minutes respectively.
* Default: 60s

cntr_2_lookback_time = [<integer>[s|m]]
* Specifies how far into history should the size/count variation be tracked
  for counter 2.
* See the 'cntr_1_lookback_time' setting description for explanation and usage
  of the lookback counter.
* Default: 600s (10 minutes)

cntr_3_lookback_time = [<integer>[s|m]]
* Specifies how far into history should the size/count variation be tracked
  for counter 3.
* See the 'cntr_1_lookback_time' setting description for explanation and usage
  of the lookback counter.
* Default: 900s (15 minutes)

sampling_interval = [<integer>[s|m]]
* The lookback counters described earlier collect the size and count
  measurements for the queues. This setting specifies at what interval the
```

```
  measurement collection happens.
* NOTE: The counter sampling interval is the same for all counters in
  a particular queue.
* Specify this value using integer followed by [s|m] which stands for
  seconds and minutes, respectively.
* Default: 1s

[queue=<queueName>]

maxSize = [<integer>|<integer>[KB|MB|GB]]
* Specifies the capacity of a queue. It overrides the default capacity
  specified in the [queue] stanza.
* If specified as a lone integer (for example, maxSize=1000), maxSize
  indicates the maximum number of events allowed in the queue.
* If specified as an integer followed by KB, MB, or GB (for example,
  maxSize=100MB), it indicates the maximum RAM allocated for queue.
* Default: The default is inherited from the 'maxSize' value specified
  in the [queue] stanza

cntr_1_lookback_time = [<integer>[s|m]]
* Same explanation as mentioned in the [queue] stanza.
* Specifies the lookback time for the specific queue for counter 1.
* Default: The default value is inherited from the 'cntr_1_lookback_time'
  value that is specified in the [queue] stanza

cntr_2_lookback_time = [<integer>[s|m]]
* Specifies the lookback time for the specific queue for counter 2.
* Default: The default value is inherited from the 'cntr_2_lookback_time'
  value that is specified in the [queue] stanza.

cntr_3_lookback_time = [<integer>[s|m]]
* Specifies the lookback time for the specific queue for counter 3.
* Default: The default value is inherited from the 'cntr_3_lookback_time' value
  that is specified in the [queue] stanza.

sampling_interval = [<integer>[s|m]]
* Specifies the sampling interval for the specific queue.
* Default: The default value is inherited from the 'sampling_interval' value
  specified in the [queue] stanza.
```

PubSub server settings for the http endpoint.

```
[pubsubsvr-http]

disabled = <boolean>
* If disabled, then http endpoint is not registered. Set this value to
  'false' to expose PubSub server on http.
* Default: true

stateIntervalInSecs = <seconds>
* The number of seconds before a connection is flushed due to inactivity.
  The connection is not closed, only messages for that connection are
  flushed.
* Default: 300 (5 minutes)
```

*General file input settings. ** NOT SUPPORTED **

```
# [fileInput]
# outputQueue = <queue name>
* REMOVED. Historically this allowed the user to set the target queue for the
  file-input (tailing) processor, but there was no valid reason to modify this.
* This setting is now removed, and has no effect.
* Tailing always uses the parsingQueue.
```

*Settings controlling the behavior of 'splunk diag', the diagnostic tool

```
[diag]

# These settings provide defaults for invocations of the splunk diag
# command. Generally these can be further modified by command line flags to
# the diag command.

EXCLUDE-<class> = <glob expression>
* Specifies a glob / shell pattern to be excluded from diags generated on
  this Splunk instance.
  * Example: */etc/secret_app/local/*.conf
* Further excludes can be added at the splunk diag command line, but there
  is no facility to disable configuration-based excludes at the command
  line.
* There is one exclude by default, for the splunk.secret file.

# the following commands can be overridden entirely by their command-line
# equivalents.

components = <comma-separated list>
* Specifies which components of the diag should be gathered.
* This allows the disabling and enabling, categorically, of entire portions
  of diag functionality.
* All of these components are further subject to the EXCLUDE-<class> setting
  and component-specific filters (see the following component list.)
* Currently, with no configuration, all components except "rest" are enabled
  by default.
* Available components are:
  * index_files   : Files from the index that indicate their health
                    (Hosts|Sources|Sourcetypes.data and bucketManifests).
                    User data is not collected.
  * index_listing : Directory listings of the index contents are
                    gathered, in order to see filenames, directory names,
                    sizes, timestamps and the like.
  * etc           : The entire contents of the $SPLUNK_HOME/etc
                    directory.  In other words, the configuration files.
  * log           : The contents of $SPLUNK_HOME/var/log/...
  * pool          : If search head pooling is enabled, the contents of the
                    pool dir.
  * dispatch      : Search artifacts, without the actual results,
                    In other words var/run/splunk/dispatch, but not the
                    results or events files
  * searchpeers   : Directory listings of knowledge bundles replicated for
                    distributed search
                    In other words: $SPLUNK_HOME/var/run/searchpeers
  * consensus     : Consensus protocol files produced by search head clustering
```

```
                    In other words: $SPLUNK_HOME/var/run/splunk/_raft
  * conf_replication_summary : Directory listing of configuration
                    replication summaries produced by search head clustering
                    In other words: $SPLUNK_HOME/var/run/splunk/snapshot
  * rest           : The contents of a variety of splunkd endpoints
                    Includes server status messages (system banners),
                    licenser banners, configured monitor inputs & tailing
                    file status (progress reading input files).
                    * On cluster managers, also gathers manager info, fixups,
                      current peer list, clustered index info, current
                      generation, & buckets in bad stats
                    * On cluster peers, also gathers local buckets & local
                      peer info, and the manager information remotely from
                      the configured manager.
  * kvstore        : Directory listings of the KV Store data directory
                    contents are gathered, in order to see filenames,
                    directory names, sizes, and timestamps.
  * file_validate : Produce list of files that were in the install media
                    which have been changed.  Generally this should be an
                    empty list.
  * profiler       : The profiler directory at $SPLUNK_HOME/var/run/profiler

* The special value "all" is also supported, enabling everything explicitly.
* Further controlling the components from the command line:
    * The switch --collect replaces this list entirely.
        * Example: --collect log,etc
          This would set the components to log and etc only, regardless of
          config
    * The switch --enable adds a specific component to this list.
        * Example: --enable pool
          This would ensure that pool data is collected, regardless of
          config
    * The switch --disable removes a specific component from this list.
        * Example: --disable pool
          This would ensure that pool data is *NOT* collected, regardless of
          config
* Default: All components except "rest"

# Data filters:
# These filters further refine what the diag tool collects.
# Most of the existing ones are designed to limit the size and collection
# time to acceptable values.

# NOTE: Most values here use underscores, while the command line uses
# hyphens.

all_dumps = <boolean>
* This setting currently is not applicable on UNIX platforms.
* Affects the 'log' component of diag. (dumps are written to the log directory
  on Windows)
* Can be overridden with the --all-dumps command line argument.
* Normally, Splunk diag gathers only three .DMP (crash dump) files on
  Windows to limit diag size.
* If this is set to true, splunk diag collects *all* .DMP files from
  the log directory.
* No default. (false equivalent)

index_files = [full|manifests]
* Selects a detail level for the 'index_files' component.
* Can be overridden with the --index-files command line flag.
* If set to "manifests", limits the index file-content collection to just
  .bucketManifest files which give some information about the general state of
```

```
  buckets in an index.
* If set to "full", adds the collection of Hosts.data, Sources.data, and
  Sourcetypes.data which indicate the breakdown of count of items by those
  categories per-bucket, and the timespans of those category entries
    * "full" can take quite some time on very large index sizes, especially
      when slower remote storage is involved.
* Default: manifests

index_listing = [full|light]
* Selects a detail level for the 'index_listing' component.
* Can be overridden with the --index-listing command line flag.
* "light" gets directory listings (ls, or dir) of the hot/warm and cold
  container directory locations of the indexes, as well as listings of each
  hot bucket.
* "full" gets a recursive directory listing of all the contents of every
  index location, which should mean all contents of all buckets.
  * "full" can take a significant amount of time with very large
    bucket counts, especially on slower storage.
* Default: light

etc_filesize_limit = <non-negative integer>
* This filters the 'etc' component.
* Can be overridden with the --etc-filesize-limit command line flag
* This value is specified in kilobytes.
    * Example: 2000 - this would be approximately 2MB.
* Files in the $SPLUNK_HOME/etc directory which are larger than this limit
  is not collected in the diag.
* Diag produces a message stating that a file has been skipped for size
  to the console. (In practice, large files have been found to oftentimes
  be a surprise to the administrator, and indicate problems).
* You can disable this filter by setting the value to 0.
* Currently, as a special exception, the file
  $SPLUNK_HOME/system/replication/ops.json is permitted to be 10x the
  size of this limit.
* Default: 10000 (10MB)

log_age = <non-negative integer>
* This filters the 'log' component.
* Can be overridden with the --log-age command line flag
* This value is specified in days.
  * Example: 75 - this would be 75 days, or about 2.5 months.
* You can disable this filter by setting the value to 0.
* The idea of this default filter is that data older than this is rarely
  helpful in troubleshooting cases in any event.
* Default: 60 (or approximately 2 months)

upload_proto_host_port = <protocol://host:port>|disabled
* The URI base to use for uploading files/diags to Splunk support.
* If set to "disabled" (override in a local/server.conf file), effectively
  disables diag upload functionality for this Splunk instance.
* Modification can theoretically permit operations with some forms of
  proxies, but diag is not specifically designed for such, and support of proxy
  configurations that do not currently work is considered an Enhancement
  Request.
* The communication path with api.splunk.com is over a simple but not
  documented protocol. If you want to accept diag uploads into
  your own systems, it probably is simpler to run diag and then upload via
  your own means independently. However if you have business reasons that you
  want this built-in, get in touch.
* Do not upload using unencrypted HTTP protocol unless you have no other choice.
* Default: https://api.splunk.com
```

```
SEARCHFILTERSIMPLE-<class> = regex
SEARCHFILTERLUHN-<class> = regex
* Redacts strings from ad-hoc searches logged in the audit.log and
  remote_searches.log files.
* Substrings which match these regexes *inside* a search string in one of those
  two files is replaced by sequences of the character X, as in XXXXXXXX.
* Substrings which match a SEARCHFILTERLUHN regex has the contained
  numbers further tested against the Luhn algorithm, used for data integrity
  in mostly financial circles, such as credit card numbers. This permits more
  accurate identification of that type of data, relying less heavily on regex
  precision. See the Wikipedia article on the "Luhn algorithm" for additional
  information.
* Search string filtering is disabled if --no-filter-searchstrings is
  used on the command line.
* NOTE: That matching regexes must match only the bytes of the
  term. Each match "consumes" a portion of the search string, so matches that
  extend beyond the term (for example, to adjacent whitespace) could prevent
  subsequent matches, and/or redact data needed for troubleshooting.
* Use a name that hints at the purpose of the filter in the <class>
  component of the setting name, and consider an additional explicative
  comment, even for custom local settings. This might skip inquiries from
  support.
```

License manager settings for configuring the license pool(s)

```
[license]

master_uri = [self|<uri>]
* DEPRECATED. Use the 'manager_uri' setting instead.

manager_uri = [self|<uri>]
* The URI of the license manager that a license peer connects to.
* If set to a URI, the instance attempts to connect to the license manager at
  the URI you specify.
* A URI consists of the following: <scheme>://<hostname>:<port>
* For example, if you set "manager_uri = https://example.com:8089", then the
  instance attempts a connection to the instance at "http://example.com:8089"
  to get licensing information.
* No default.

active_group = Enterprise|Trial|Forwarder|Free
* If the instance is a license manager, the license type will be set in 'active_group'.
* Default: <empty>

connection_timeout = <integer>
* Maximum time, in seconds, to wait before sending data to the manager times out.
* This timeout applies only if 'manager_uri' is set.
* Default: 30

send_timeout = <integer>
* Maximum time, in seconds, to wait before sending data to the manager times out.
* This timeout applies only if 'manager_uri' is set.
* Default: 30

receive_timeout = <integer>
* Maximum time, in seconds, to wait before receiving data from the manager times out
* This timeout applies only if 'manager_uri' is set.
* Default: 30
```

```
squash_threshold = <positive integer>
* Periodically the indexer must report to license manager
  the data indexed broken down by source, sourcetype, host, and index. If
  the number of distinct (source, sourcetype, host, index) tuples grows over
  the 'squash_threshold', the (host, source) values are squashed and only a
  breakdown by (sourcetype, index) is reported. This is to prevent explosions in
  memory + license_usage.log lines.
* This is an advanced setting. Set it only after consulting a Splunk
  Support engineer.
* This needs to be set on license peers as well as the license
  manager.
* Default: 2000

report_interval = <nonnegative integer>[s|m|h]
* Selects a time period for reporting in license usage to the license
  manager.
* This value is intended for very large deployments (hundreds of indexers)
  where a large number of indexers may overwhelm the license server.
* The maximum permitted interval is 1 hour.
* The minimum permitted interval is 1 minute.
* Can be expressed as a positive number of seconds, minutes or hours.
* If no time unit is provided, seconds is assumed.
* Default: 1m

license_warnings_update_interval = <nonnegative integer>
* Specifies a time period, in seconds, for license manager to update
  license warnings in Splunk Web bulletin messages.
* License manager checks at every second the last time it updated the
  warnings, and updates if this time period has elapsed.
* Increase this value for very large deployments that contain very
  large number of source types.
* The minimum permitted interval is 10.
* The maximum permitted interval is 3600, equivalent to 1 hour.
* If set to the special value of 0, the license manager automatically tunes this
  setting to accommodate the size of the deployment.
* Default: 0

strict_pool_quota = <boolean>
* Toggles strict pool quota enforcement.
* A value of "true" means members of pools receive warnings for a given day if
  usage exceeds pool size regardless of whether overall stack quota was
  exceeded
* A value of "false" means members of pool only receive warnings if both pool
  usage exceeds pool size AND overall stack usage exceeds stack size
* Default: true

pool_suggestion = <string>
* Suggest a pool to the manager for this peer.
* The manager uses this suggestion if the manager doesn't have an explicit
  rule mapping the peer to a given pool (ie...no peer list for the
  relevant license stack contains this peer explicitly)
* If the pool name doesn't match any existing pool, it is ignored, no
  error is generated
* This setting is intended to give an alternative management option for
  pool/peer mappings. When onboarding an indexer, it may be easier to
  manage the mapping on the indexer itself via this setting rather than
  having to update server.conf on manager for every addition of new indexer
* NOTE: If you have multiple stacks and a peer maps to multiple pools, this
        feature is limited in only allowing a suggestion of a single pool;
        This is not a common scenario however.
* No default. (which means this feature is disabled)
```

```
lm_uri = <comma-separated list>
* A list of the URIs of license managers that this instance is to use when
  in High Availability Redundancy Mode.
* High Availability Redundancy mode lets you use multiple license managers
  behind a load balancer.
* Separate multiple entries with commas.
* If you give this setting a value, that value cannot be empty.
* This setting is valid only when you enable High Availability Redundancy mode,
  which requires a special license and is only available to select customers.
* No default.

lm_ping_interval = <positive integer>
* How often, in seconds, that license managers communicate with each other
  to check if they are all online and have the same license.
* This setting is valid only when you enable High Availability Redundancy mode,
  which requires a special license and is only available to select customers.
* Default: 86400 (once a day)

[lmpool:auto_generated_pool_forwarder]
* This is the auto generated pool for the forwarder stack

description = <textual description of this license pool>
quota = MAX|<maximum amount allowed by this license>
* MAX indicates the total capacity of the license. You may have only 1 pool
  with MAX size in a stack.
* The quota can also be specified as a specific size eg. 20MB, 1GB, etc.

slaves = *|<slave list>
* DEPRECATED. Use the 'peers' setting instead.

peers = *|<peer list>
* An asterisk(*) indicates that any peer can connect to this pool.
* You can also specify a comma separated peer GUID list.

stack_id = forwarder
* The stack to which this pool belongs.

[lmpool:auto_generated_pool_free]
* This is the auto generated pool for the free stack.
* Field descriptions are the same as that for
  the 'lmpool:auto_generated_pool_forwarder' setting.

[lmpool:auto_generated_pool_enterprise]
* This is the auto generated pool for the enterprise stack.
* Field descriptions are the same as that for
  the 'lmpool:auto_generated_pool_forwarder' setting.


[lmpool:auto_generated_pool_fixed-sourcetype_<sha256 hash of srctypes>]
* This is the auto generated pool for the enterprise fixed srctype stack
* Field descriptions are the same as that for
  the 'lmpool:auto_generated_pool_forwarder' setting.

[lmpool:auto_generated_pool_download_trial]
* This is the auto generated pool for the download trial stack.
* Field descriptions are the same as that for
  the "lmpool:auto_generated_pool_forwarder"

[pooling]

state = [enabled|disabled]
* UNSUPPORTED: This setting is no longer supported.
```

```
storage = <path to shared storage>
* UNSUPPORTED: This setting is no longer supported.

app_update_triggers = true|false|silent
* UNSUPPORTED: This setting is no longer supported.

lock.timeout = <time range string>
* UNSUPPORTED: This setting is no longer supported.

lock.logging = <boolean>
* UNSUPPORTED: This setting is no longer supported.

poll.interval.rebuild = <time range string>
* UNSUPPORTED: This setting is no longer supported.

poll.interval.check = <time range string>
* UNSUPPORTED: This setting is no longer supported.

poll.blacklist.<name> = <regex>
* UNSUPPORTED: This setting is no longer supported.


############################################################################
# Amazon Web Services Elastic Compute Cloud Instance Metadata Service
# (AWS EC2 IMDS) configuration
############################################################################

[imds]

imds_version = [v1|v2]
* Sets IMDS version for EC2 instances metadata endpoints.
* This setting is AWS specific.
* Certain features of the Splunk platform use AWS Instance Metadata Service
  (IMDS) when hosted on EC2. IMDS is accessible from the instance via a
  link-local address. It provides metadata about the instance.
* v1 uses request/response method while v2 uses a session-oriented method
  to access IMDS. The version should match the setting used on your EC2
  instance.
* More information about IMDS can be found in the AWS documentation.
* Default: v1
```

High availability clustering configuration

```
[clustering]

mode = [manager|peer|searchhead|disabled]
* Sets operational mode for this cluster node.
* Only one manager may exist per cluster.
* Note: "manager" and "peer" replace the prior 'mode' values of
  "master" and "slave". The prior values are currently still supported,
  but they will be removed from the product in a future release.
* Default: disabled

master_uri = [<uri> | clustermanager:<cm-name1>, clustermanager:<cm-name2>, ...]
* DEPRECATED. Use the 'manager_uri' setting instead.

manager_uri = [<uri> | clustermanager:<cm-name1>, clustermanager:<cm-name2>, ...]
```

```
* There are two uses for this setting, one for 'mode=peer' and 'mode=searchhead',
  and another for 'mode=manager'.
* For 'mode=peer' and 'mode=searchhead':
  * Specify the URI of the cluster manager that the peer or search head
    connects to.
  * An example of <uri>: <scheme>://<hostname>:<port>
  * For 'mode=searchhead' only: If the search head belongs to multiple
    clusters, specify the manager URIs in a comma separated list.
* For 'mode=manager':
  * Only valid if 'manager_switchover_mode=auto|manual'.
  * In this mode, a list of cluster manager stanzas [clustermanager:<cm-nameX>]
    must be provided. Those managers participate in the manager redundancy feature
    for the indexer cluster.
  * This list must be exactly identical in all the participating manager instances.
  * The list of managers serves as a priority list, where a manager earlier
    in the list has higher priority than a manager later in the list.
  * When two managers start up together, they detect each other's presence.
    To determine which manager will be active and which will be standby,
    they use the priority established by the list.
  * Similarly, when 'manager_switchover_mode=auto', and there is one active
    cluster manager and multiple standby managers, if the active manager then
    goes down, the standby manager with highest priority becomes the active
    manager.

manager_switchover_mode = [disabled|auto|manual]
* Set the cluster manager redundancy operation mode.
* Only valid for 'mode=manager'.
* If set to "disabled", the cluster manager does not operate with redundancy.
* The values "auto" and "manual" are valid only when the 'manager_uri' setting
  in server.conf includes multiple cluster manager values.
* If set to "auto", an eligible "standby" cluster manager will try to automatically
  set its redundancy state to "active" upon consecutive loss of heartbeat to the
  manager which is currently in active state.
* If set to "manual", there is no automatic change of redundancy state to "active"
  during heartbeat failure, but there can be other scenarios, where the redundancy
  state may change automatically for some manager instances, as explained below:
  * An eligible "standby" cluster manager will not set its redundancy state to
    "active" upon consecutive loss of heartbeat to the manager which is currently
    in active state. The administrator must manually change the cluster manager
    redundancy state to "active" or "standby".
  * However, when the administrator initiates a redundancy state change request
    to one of the managers, the given manager may communicate with other
    managers internally and can change their redundancy states appropriately
    so that it can move to the correct redundancy state as requested by the
    administrator.
* Cluster manager redundancy solution is closely related to the general
  Splunk Enterprise deployment topology and network environment. In redundancy
  mode, the effectiveness of the chosen 'manager_switchover_mode' can be dependent
  on the actual network architecture that connects the peer nodes with the
  cluster manager, for example, DNS or load-balancer based deployment.
* Contact Splunk Professional Services for guidance on implementing
  cluster manager redundancy.
* Default: disabled

advertised_disk_capacity = <integer>
* Only valid for 'mode=peer'.
* Percentage to use when advertising disk capacity to the cluster manager.
  This is useful for modifying weighted load balancing in indexer discovery.
* For example, if you set this to 50 for an indexer with a
  500GB disk, the indexer advertises its disk size as 250GB, not 500GB.
* Acceptable value range is 10 to 100.
* Default: 100
```

```
pass4SymmKey = <string>
* Secret shared among the nodes in the cluster to prevent any
  arbitrary node from connecting to the cluster. If a peer or
  search head is not configured with the same secret as the manager,
  it is not able to communicate with the manager.
* If 'pass4SymmKey' is not set in the [clustering] stanza, Splunk software
  looks for the key in the [general] stanza.
* Unencrypted passwords must not begin with "$1$", as Splunk software uses
  this substring to determine if the password is already encrypted.
* No default.

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' should be for a
  particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length
  than what you specify with this setting, the platform warns you and advises
  that you change the pass4SymKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* Default: 12

service_interval = <zero or positive integer>
* Only valid when 'mode=manager'.
* How often, in seconds, that the manager runs its service
  loop.
* In its service loop, the manager checks the state of the
  peers and the buckets in the cluster and also schedules
  corrective action, if possible, for buckets that are not in
  compliance with replication policies.
* A special default value of 0 indicates an auto mode where the service
  interval for the next service call is determined by the time taken by
  previous call. It also sets the minimum service interval to be 0.5 second.
* Service interval is bounded by the values 1 and
  the 'max_auto_service_interval' setting.
  If the previous service call takes more than 'max_auto_service_interval'
  seconds, the next service interval is set to
  'max_auto_service_interval' seconds.
* Default: 0

service_execution_threshold_ms = <zero or positive integer>
* Only valid when 'mode=manager'.
* Specifies, in milliseconds, the maximum period for one execution
  of the manager's service loop.
* This setting is useful for large clusters with large numbers of
  buckets, to prevent the service loop from blocking
  other operations for significant amounts of time.
* Default: 1500

deferred_cluster_status_update = <boolean>
* Only valid when 'mode=manager'.
* A value of "true" means that SF/RF met (complete cluster state) checks are
  performed lazily for optimal performance, only when CM is busy with
  cluster maintenance operations (for example, peer addition, fix ups,
  data rebalance).
* A value of "false" means that SF/RF met checks are performed relatively
  more aggressively to improve accuracy, increasing CM overhead and slowing
  down cluster maintenance operations (e.g peer addition, fix ups, data rebalance).
* NOTE: It is recommended to be set as "true" with high number of indexers
  and buckets.
* Default: true
```

```
deferred_rest_api_update = <boolean>
* Only valid when 'mode=manager'.
* A value of "true" means the manager responds to a REST API call from a source
  peer immediately. It might defer part of the actions related to the call until
  it completes already pending work.
* A value of "false" means the manager finishes all work for a received REST API
  call and only then responds to the source peer. The response might be delayed
  if the manager is busy with other work.
* Default: true

max_fixup_time_ms = <zero or positive integer>
* Only valid for 'mode=manager'.
* How long, in milliseconds, each fixup level runs before
  short circuiting to continue to the next fixup level. This
  introduces an upper bound on each service level, and likewise
  introduces an upper bound on the full service() call.
* This setting is useful for larger clusters that have lots of
  buckets, where service() calls can consume a significant amount
  of time blocking other operations.
* 0 denotes that there is no max fixup timer.
* Default: 1000

max_delayed_updates_time_ms = <zero or positive integer>
* Only valid for 'mode=manager'.
* How long, in milliseconds, the cluster manager can continuously
  serve the delayed jobs before quitting to run other jobs.
* This setting is useful for larger clusters that have a large number of
  peer nodes and indexes, where customer manager could occasionally receive
  thousands of REST APIs in a short period.
* Do not change this setting without first consulting with Splunk Support.
* 0 denotes that there is no limit to how long the delayed jobs thread
  can run continuously.
* Default: 1000

primary_src_persist_secs = <zero or positive integer>
* Only valid for 'mode=manager'.
* For a warm bucket, this setting specifies the interval after the bucket's
  latest time that a primary rebalance operation attempts to assign the primary
  to the copy on the source peer node. Once the interval is exceeded,
  the rebalance operation no longer considers the bucket's origin when
  assigning its primary.
* For a hot bucket, a non-zero value causes the primary to always reside with the
  source's hot bucket.
* Do not change this setting without first consulting with Splunk Support.
* If set to 0, the rebalance operation does not consider bucket origin
  when assigning primaries, for both hot and warm buckets.
* Default: 604800 (1 week, 60 * 60 * 24 * 7 seconds)

cm_heartbeat_period = <non-zero positive integer>
* Only valid for 'mode=manager' and 'manager_switchover_mode=auto|manual'.
* Determines the frequency, in seconds, of cluster manager to cluster
  manager heartbeat.
* Default: 1

cm_max_hbmiss_count = <non-zero positive integer>
* Only valid for 'mode=manager' and 'manager_switchover_mode=auto|manual'.
* The maximum number of consecutive heartbeat misses allowed before a
* cluster manager in standby state triggers the switchover sequence.
* Default: 3

cm_com_timeout = <integer>
```

```
* The timeout, in seconds, used in communications between cluster managers
  in redundancy mode.
* Only valid with 'mode=manager' and 'manager_switchover_mode=auto|manual'.
* Depending on the type of message being exchanged, triggering a timeout can
  result in a variety of consequences. For example, if the timeout is triggered
  for a heartbeat message, and the missed heartbeat count exceeds the value of
  'cm_max_hbmiss_count', a manager switchover will be triggered, if
  'manager_switchover_mode=auto'.
* Default: 10

cxn_timeout = <integer>
* The low-level timeout, in seconds, for establishing connection between
  cluster nodes.
* Default: 60

send_timeout = <integer>
* The low-level timeout, in seconds, for sending data between cluster nodes.
* Default: 60

rcv_timeout = <integer>
* The low-level timeout, in seconds, for receiving data between cluster nodes.
* Default: 60

rep_cxn_timeout = <integer>
* Valid only for 'mode=peer'.
* The low-level timeout, in seconds, for establishing connection for replicating
  data.
* Default: 60

rep_send_timeout = <integer>
* Only valid for 'mode=peer'.
* The low-level timeout, in seconds, for sending replication slice data between
  cluster nodes.
* This is a soft timeout. When this timeout is triggered on a source peer,
  it tries to determine if target is still alive. If the target is still alive, it
  resets the timeout for another 'rep_send_timeout' seconds and continues.  If
  target has failed or the cumulative timeout has exceeded the
  'rep_max_send_timeout', replication fails.
* Default: 60

rep_rcv_timeout = <integer>
* Only valid for 'mode=peer'.
* Lowlevel timeout, in seconds, for receiving acknowledgment data from peers.
* This is a soft timeout. When this timeout is triggered on source peer,
  it tries to determine if target is still alive. If it is still alive,
  it reset the timeout for another 'rep_send_timeout' interval and continues.
* If target has failed or cumulative timeout has exceeded
  'rep_max_rcv_timeout', replication fails.
* Default: 60

rep_max_send_timeout = <integer>
* Only valid for 'mode=peer'.
* Maximum send timeout, in seconds, for sending replication slice
  data between cluster nodes.
* On rep_send_timeout source peer determines if total send timeout has
  exceeded 'rep_max_send_timeout'. If so, replication fails.
* If cumulative 'rep_send_timeout' exceeds 'rep_max_send_timeout',
  replication fails.
* For a standalone indexer, changes to this setting are dynamically reloadable
  and do not require a restart.
* For indexer clusters, changes to this setting trigger a rolling restart
  of peer nodes.
```

```
* Default: 180 (3 minutes)

rep_max_rcv_timeout = <integer>
* Only valid for 'mode=peer'.
* Maximum cumulative receive timeout, in seconds, for receiving
  acknowledgment data from peers.
* On 'rep_rcv_timeout', the source peer determines if the total
  receive timeout has exceeded 'rep_max_rcv_timeout'.
  If so, replication fails.
* For a standalone indexer, changes to this setting are dynamically reloadable
  and do not require a restart.
* For indexer clusters, changes to this setting trigger a rolling restart
  of peer nodes.
* Default: 180 (3 minutes)

search_files_retry_timeout = <integer>
* Only valid for 'mode=peer'.
* Timeout, in seconds, after which request for search files from a
  peer is aborted.
* To make a bucket searchable, search specific files are copied from
  another source peer with search files. If search files on source
  peers are undergoing changes, the source peer asks the requesting peer
  to retry after some time. If the cumulative retry period exceeds the
  specified timeout, the requesting peer aborts the request and requests
  search files from another peer in the cluster that might have search files.
* Default: 600 (10 minutes)

re_add_on_bucket_request_error = <boolean>
* Valid only for 'mode=peer'.
* Whether or not a peer re-adds itself to the cluster manager if the manager
  returns an error on any bucket request.
* A value of "true" means the peer re-adds itself to the cluster manager if
  cluster manager returns an error on any bucket request. On re-add,
  peer updates the manager with the latest state of all its buckets.
* A value of "false" means the peer doesn't re-add itself to the cluster manager.
  Instead, it updates the manager with those buckets that manager
  returned an error.
* Default: false

decommission_search_jobs_wait_secs = <unsigned integer>
* Valid only for 'mode=peer'.
* The maximum time, in seconds, that a peer node waits for search
  jobs to finish before it transitions to the 'down' or 'GracefulShutdown'
  state, in response to the 'splunk offline' or
  'splunk offline --enforce-counts' command.
* NOTE: When using this setting, the 'decommission_search_jobs_wait_secs'
  setting in the '[general]' stanza must remain set to its default value.
* You do not need to restart the cluster peer when making changes to
  this setting. This setting reloads automatically.
* Default: 180 (3 minutes)

decommission_node_force_timeout = <seconds>
* Valid only for 'mode=peer' and during node offline operation.
* The maximum time, in seconds, that a peer node waits for searchable copy
  reallocation jobs to finish before it transitions to the 'down' or
  'GracefulShutdown' state.
* This period begins after the peer node receives a 'splunk offline' command
  or its '/cluster/slave/control/control/decommission' REST endpoint
  is accessed.
* This attribute is not applicable to the  "--enforce-counts" version of the
  "splunk offline" command
* Default: 300 seconds
```

```
decommission_force_finish_idle_time = <zero or positive integer>
* Valid only for 'mode=manager'.
* The time, in minutes, that the manager waits before forcibly finishing the
  decommissioning of a peer when there is no progress in the associated
  fixup activity.
* A value of zero (0) means that the manager does not forcibly finish
  decommissioning.
* Default: 0

rolling_restart = restart|shutdown|searchable|searchable_force
* Only valid for 'mode=manager'.
* Determines whether indexer peers restart or shutdown during a rolling
  restart.
* If set to "restart", each peer automatically restarts during a rolling
  restart.
* If set to "shutdown", each peer is stopped during a rolling restart,
  and the customer must manually restart each peer.
* If set to "searchable", the cluster attempts a best effort to maintain
  a searchable state during the rolling restart by reassigning primaries
  from peers that are about to restart to other searchable peers, and
  performing a health check to ensure that a searchable rolling restart is
  possible.
* If set to "searchable_force", the cluster performs a searchable
  rolling restart, but overrides the health check and enforces
  'decommission_force_timeout' and 'restart_inactivity_timeout'.
* If set to "searchable" or "searchable_force", scheduled searches
  are deferred or run during the rolling restart based on the
  'defer_scheduled_searchable_idx' setting in 'savedsearches.conf'.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: restart

searchable_rolling_peer_state_delay_interval = <zero or positive integer>
* Only valid for 'mode=manager'.
* Specifies an extra time interval, in seconds, during which the peer remains
  in the ReassigningPrimaries state.
* Extending the amount of time the peer remains in the ReassigningPrimaries
  state gives the peer more time to complete inflight searches and ingest
  events.
* This also reduces the impact of incomplete searches and bucket corruption,
  which can impede the searchable rolling restart process.
* Default: 60

searchable_rolling_site_down_policy = full|most|half
* Only valid for 'mode=manager' and is only used if 'multisite=true' and
  'site_by_site=true'.
* Sets the policy for calculating the maximum number of peers in a site allowed
  to shutdown at the same time during a searchable rolling restart.
* If set to 'full', the manager will allow an entire site to shutdown at once
  if there are searchable copies of buckets available in at least 3 sites.
* If set to 'most', the manager will maintain a few peers in a site to act as
  hot bucket streaming targets, and shutdown the other peers. At least one
  peer is available as a streaming target, but there can be more depending on
  the cluster's search factor. The 'most' policy attempts to speed up the
  rolling restart more aggressively than 'half', at the expense of a longer
  period fixing up replication and search factors afterwards.
* If set to 'half', the manager will maintain half the peers in a site to act
  as hot bucket streaming targets, and shutdown the other peers.
* The maximum number of peers allowed down at one time is the smallest peer
  count between 'percent_peers_to_restart' and
  'searchable_rolling_site_down_policy'.
```

```
* Default: half

rolling_restart_condition = up|batch_adding|starting
* Only valid for 'mode=manager'.
* Determines the target peer status the manager waits for, when restarting
  a peer during a rolling restart, before it restarts other peers.
* If set to "up", the manager will wait for a restarting peer to reach the
  'Up' status before restarting other peers. A peer reaches 'Up' status
  when it has finished reporting all of its buckets to the manager. This
  option will always respect 'percent_peers_to_restart'.
* If set to "batch_adding", the manager will wait for a restarting peer to
  reach the 'BatchAdding' status before restarting other peers. A peer
  reaches 'BatchAdding' status when it is in the process of reporting its
  buckets to the manager. This option will respect 'percent_peers_to_restart'
  as long as the current restarting peer finishes adding before the next
  restarting peer finishes shutting down, which is extremely likely.
* If set to "starting", the manager will wait for a restarting peer to
  reach the 'Starting' status before restarting other peers. A peer
  reaches 'Starting' status when it first starts up and is in the process
  of scanning its buckets on disk. This option is the fastest, but may
  not always respect 'percent_peers_to_restart'.
* Default: batch_adding

site_by_site = <boolean>
* Only valid for 'mode=manager' and 'multisite=true'.
* Whether or not the manager limits peer restarts to one site at a time during
  a rolling restart.
* A value of "true" means the manager restarts peers from one site at a time,
  waiting for all peers from a site to restart before moving on to another
  site, during a rolling restart.
* A value of "false" means the manager randomly selects peers to restart, from
  across all sites, during a rolling restart.
* Default: true

decommission_force_timeout = <zero or positive integer>
* Only valid for 'mode=manager'.
* Only valid for 'rolling_restart=searchable_force'.
* The amount of time, in seconds, the cluster manager waits for a
  peer in primary decommission status to finish primary reassignment
  and restart, during a searchable rolling restart with timeouts.
* Differs from 'decommission_force_finish_idle_time' in its default value
  and its presence only during a searchable rolling restart with timeouts.
* If you set this parameter to 0, it is automatically reset
  to default value.
* Maximum accepted value is 1800 (30 minutes).
* Default: 180 (3 minutes)

restart_inactivity_timeout = <zero or positive integer>
* Only valid for 'mode=manager'.
* Only valid for 'rolling_restart=searchable_force'.
* The amount of time, in seconds, that the manager waits for a peer to
  restart and rejoin the cluster before it considers the restart a failure
  and proceeds to restart other peers.
* More specifically, the amount of time that the manager waits for a peer in
  the 'Down' status to transition to 'BatchAdding' or 'Up' status.
* A value of zero (0) means that the manager waits indefinitely for a peer
  to restart.
* Default: 600 (10 minutes)

rebalance_pipeline_batch_size = <integer>
* Valid only for 'mode=manager'.
* Valid only for 'searchable_rebalance=true'.
```

```
* The maximum number of buckets for a batch entering the excess bucket removal
  phase of the rebalance pipeline.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 60

rebalance_primary_failover_timeout = <zero or positive integer>
* Valid only for 'mode=manager'.
* Valid only for 'searchable_rebalance=true'.
* The maximum length of time, in seconds, that the manager waits for primacy to
  be reassigned from the batch of excess buckets to other buckets.
* Default: 75

rebalance_newgen_propagation_timeout = <zero or positive integer>
* Valid only for 'mode=manager'.
* Valid only for 'searchable_rebalance=true'.
* The amount of time, in seconds, that the manager waits for the search heads to
  get the newly committed generation after the discarded buckets' primacy has
  been reassigned.
* Default: 60 (1 minute)

rebalance_search_completion_timeout = <integer>
* Valid only for 'mode=manager'.
* Valid only for 'searchable_rebalance=true'.
* The amount of time, in seconds, that the manager waits for older generation
  searches on indexers to complete before removing any excess buckets.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 180 (3 minute)

searchable_rebalance = <boolean>
* Valid only for 'mode=manager'.
* Controls whether searches can continue uninterrupted during data rebalancing.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: false

multisite = <boolean>
* Only valid for 'mode=manager'.
* Whether or not the manager uses multisite mode.
* A value of "true" means that the manager turns on the multisite feature.
* Confirm that you set site parameters on the peers when you set this to "true".
* Default: false

replication_factor = <positive integer>
* Only valid for 'mode=manager'.
* Determines how many copies of rawdata are created in the cluster.
* Use 'site_replication_factor' instead of this in case 'multisite'
  is turned on.
* Must be greater than 0.
* Default: 3

site_replication_factor = <comma-separated string>
* Only valid for 'mode=manager' and is only used if 'multisite=true'.
* This specifies the per-site replication policy for any given
  bucket represented as a comma-separated list of per-site entries.
* Currently specified globally and applies to buckets in all
  indexes.
* Each entry is of the form <site-id>:<positive integer> which
  represents the number of copies to make in the specified site
* Valid site-ids include two mandatory keywords and optionally
  specific site-ids from site1 to site63
```

```
* The mandatory keywords are:
  - origin: Every bucket has a origin site which is the site of
  the peer that originally created this bucket. The notion of
  'origin' makes it possible to specify a policy that spans across
  multiple sites without having to enumerate it per-site.
  - total: The total number of copies needed for each bucket.
* When a site is the origin, it could potentially match both the
  origin and a specific site term. In that case, the max of the
  two is used as the count for that site.
* The total must be greater than or equal to sum of all the other
  counts (including origin).
* The difference between total and the sum of all the other counts
  is distributed across the remaining sites.
* Example 1: site_replication_factor = origin:2, total:3
  Given a cluster of 3 sites, all indexing data, every site has 2
  copies of every bucket ingested in that site and one rawdata
  copy is put in one of the other 2 sites.
* Example 2: site_replication_factor = origin:2, site3:1, total:3
  Given a cluster of 3 sites, 2 of them indexing data, every
  bucket has 2 copies in the origin site and one copy in site3. So
  site3 has one rawdata copy of buckets ingested in both site1 and
  site2 and those two sites have 2 copies of their own buckets.
* Default: origin:2, total:3

search_factor = <positive integer>
* Only valid for 'mode=manager'.
* Determines how many buckets have index structures pre-built.
* Must be less than or equal to the 'replication_factor' setting and
  greater than 0.
* Default: 2

site_search_factor = <comma-separated list>
* Only valid for 'mode=manager' and is only used if 'multisite=true'.
* This specifies the per-site policy for searchable copies for any
  given bucket represented as a comma-separated list of per-site
  entries.
* This is similar to the 'site_replication_factor' setting.
  See that entry for more information on the syntax.
* Default: origin:1, total:2

ack_factor = <positive integer>
* Sets the number of copies of incoming data that must be saved across
  the indexer cluster before an acknowledgement (ACK) is returned from
  the source peer to the forwarder.
* For example, a value of 2 means that two copies of the incoming data
  must be saved on cluster peer nodes - one copy on the source node and
  another copy on one of the target nodes.
* Valid only if 'useACK=true' and 'mode=peer'.
* Supported values range from 0 to the replication factor.
* When configured to 1, the acknowledgement is sent immediately after
  the incoming data is written to the source peer's disk.
* The default value of 0 signifies the replication factor.
* For example, if the cluster has a replication factor of 3, the value
  of 0 requires that 3 copies of the incoming data be saved locally on
  peer nodes before the acknowledgement is returned to the forwarder.
* This setting must be configured to the same value on all peers in the
  cluster.
* Default: 0

available_sites = <comma-separated list>
* Only valid for 'mode=manager' and is only used if 'multisite=true'.
* This is a comma-separated list of all the sites in the cluster.
```

```
* If 'multisite=true' then 'available_sites' must be
  explicitly set.
* Example: available_sites = site1,site2,site3
* Default: an empty string

forwarder_site_failover = <comma-separated list>
* Only valid for 'mode=manager' and is only used if 'multisite=true'.
* This is a comma-separated list of pair of sites, "site1:site2",
  in the cluster.
* If 'multisite' is turned on 'forwarder_site_failover' must be
  explicitly set.
* Default: an empty string

site_mappings = <comma-separated list>
* Only valid for 'mode=manager'.
* When you decommission a site, you must update this attribute so that the
  origin bucket copies on the decommissioned site are mapped to a remaining
  active site. This attribute maps decommissioned sites to active sites.
  The bucket copies for which a decommissioned site is the origin site
  are then replicated to the active site specified by the mapping.
* Used only if multisite is true and sites have been decommissioned.
* Each comma-separated entry is of the form
  <decommissioned_site_id>:<active_site_id>
  or default_mapping:<default_site_id>.
  <decommissioned_site_id> is a decommissioned site and <active_site_id> is
  an existing site,specified in the 'available_sites' setting.
  For example, if available_sites=site1,site2,site3,site4 and you
  decommission site2, you can map site2 to a remaining site such as site4,
  like this: site2:site4 .
* If a site used in a mapping is later decommissioned, its previous mappings
  must be remapped to an available site. For instance, if you have the
  mapping site1:site2 but site2 is later decommissioned, you can remap
  both site1 and site2 to an active site3 using the following replacement
  mappings - site1:site3,site2:site3.
* Optional entry with syntax default_mapping:<default_site_id> represents the
  default mapping, for cases where an explicit mapping site is not specified.
  For example: default_mapping:site3 maps any decommissioned site to site3,
  if they are not otherwise explicitly mapped to a site.
  There can only be one such entry.
* Example 1: site_mappings = site1:site3,default_mapping:site4.
  The cluster must include site3 and site4 in available_sites, and site1
  must be decommissioned.
  The origin bucket copies for decommissioned site1 is mapped to site3.
  Bucket copies for any other decommissioned sites is mapped to site4.
* Example 2: site_mappings = site2:site3
  The cluster must include site3 in available_sites, and site2 must be
  decommissioned. The origin bucket copies for decommissioned site2 is
  mapped to site3. This cluster has no default.
* Example 3: site_mappings = default_mapping:site5
  The specified cluster must include site5 in available_sites.
  The origin bucket copies for any decommissioned sites is mapped onto
  site5.
* Default: an empty string

constrain_singlesite_buckets = <boolean>
* Only valid for 'mode=manager' and is only used if multisite is true.
* Specifies whether the cluster keeps single-site buckets within one site
  in multisite clustering.
* When this setting is "true", buckets in a single site cluster do not
  replicate outside of their site. The buckets follow 'replication_factor'
  'search factor' policies rather than 'site_replication_factor'
  'site_search_factor' policies. This is to mimic the behavior of
```

```
  single-site clustering.
* When this setting is "false", buckets previously created in
  non-multisite clusters can replicate across sites, and must meet the
  specified 'site_replication_factor' and 'site_search_factor' policies.
* Default: true

heartbeat_timeout = <positive integer>
* Only valid for 'mode=manager'.
* Specifies, in seconds, when the manager considers a peer down. After a
  peer is down, the manager initiates fixup steps to replicate
  buckets from the dead peer to its peers.
* Default: 60

access_logging_for_heartbeats = <boolean>
* Only valid for 'mode=manager'.
* Whether or not the manager logs peer heartbeats to the
  splunkd_access.logEnables/disables logging to the splunkd_access.log file
  for peer heartbeats.
* You do not have to restart the manager to set this config parameter.
  Instead, run the cli command on the manager:
    % splunk edit cluster-config -access_logging_for_heartbeats <<boolean>>
* Default: false (logging disabled)


restart_timeout = <positive integer>
* Only valid for 'mode=manager'.
* The amount of time, in seconds, the manager waits for a peer
  to come back when the peer is restarted, to avoid the overhead of
  trying to fixup the buckets that were on the peer.
* More specifically, the amount of time that the manager waits for a
  peer in the 'Restarting' status to transition to the 'Down' status.
* Note that this only works with the offline command or if the peer
  is restarted through the UI.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 60

streaming_replication_wait_secs = <positive integer>
* Only valid for 'mode=manager'.
* The amount of time, in seconds, that a peer node waits to restart after
  receiving a restart request from the manager. During this period, the node
  remains in the eRestartRequested state. This time allows the ongoing
  replications on the peer to complete.
* Default: 60

quiet_period = <positive integer>
* Only valid for 'mode=manager'.
* The amount of time, in seconds, that the manager is
  quiet upon start-up.
* However, if peers are still registering themselves with the manager after
  the initial quiet_period has elapsed, the manager continues to remain
  quiet until all peers finish registering, up to a total quiet time not to
  exceed 3x the specified 'quiet_period', including the initial quiet time.
* During the quiet time, the manager does not initiate any actions. At the end of
  this period, the manager builds its view of the cluster based on the
  registered information. It then starts normal operations.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 60

manager_switchover_quiet_period = <positive integer>
* Only valid for 'mode=manager' and 'manager_switchover_mode=auto|manual'.
```

```
* This setting determines the amount of time, in seconds, that the manager is
  quiet upon switchover from standby to active redundancy mode.
* However, if peers are still registering themselves with the manager after
  the initial manager_switchover_quiet_period has elapsed, the manager continues
  to remain quiet until all peers finish registering, up to a total quiet time
  not to exceed 3x the specified 'manager_switchover_quiet_period', including
  the initial quiet time.
* During the quiet time, the manager does not initiate any actions. At the end of
  this period, the manager builds its view of the cluster based on the
  registered information. It then starts normal operations.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 60

reporting_delay_period = <positive integer>
* Only valid for 'mode=manager'.
* The acceptable amount of delay, in seconds, for reporting both unmet
  search and unmet replication factors for newly created buckets.
* This setting helps provide more reliable cluster status reporting
  by limiting updates to the specified granularity.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 30

generation_poll_interval = <positive integer>
* How often, in seconds, the search head polls the manager for
  generation information.
* This setting is valid only if 'mode=manager' or 'mode=searchhead'.
* This setting reloads automatically and does not require a restart.
* Default: 5

generation_max_staleness = <interval><unit>
* Search heads will ignore search generation information from a cluster
  manager if it looks incomplete, either because it doesn't contain any
  search peers, or because it consists of a subset of the search head's known
  peers and at least one of the missing peers is currently joining
  the cluster.
* This setting specifies how old a search head's own information can be before
  the search head forcefully accepts new search generation information from a
  cluster manager.
* When search performance is compromised by cluster manager restarts, increase
  'generation_max_staleness' to the time it takes for all indexers to join the
  cluster.
* This setting is valid only when 'mode=searchhead'.
* Default: 60s

max_peer_build_load = <integer>
* Only valid for 'mode=manager'.
* This is the maximum number of concurrent tasks to make buckets
  searchable that can be assigned to a peer.
* Default: 2

max_peer_rep_load = <integer>
* Only valid for 'mode=manager'.
* This is the maximum number of concurrent non-streaming
  replications that a peer can take part in as a target.
* Default: 5

max_peer_sum_rep_load = <integer>
* Only valid for 'mode=manager'.
* This is the maximum number of concurrent summary replications
  that a peer can take part in as either a target or source.
```

```
* Default: 5

max_nonhot_rep_kBps = <integer>
* Only valid for 'mode=peer'.
* The maximum throughput, in kilobytes per second, for warm/cold/summary
  replications on a specific source peer.
* Similar to forwarder's 'maxKBps' setting in the limits.conf file.
* This setting throttles total bandwidth consumption for all
  outgoing non-hot replication connections from a given source peer.
  It does not throttle at the per-replication-connection, per-target
  level.
* This setting can be updated without restart on the source peers
  by using the command "splunk edit cluster-config" or by making the
  corresponding REST call.
* If set to 0, signifies unlimited throughput.
* Default: 0

max_replication_errors = <integer>
* Only valid for 'mode=peer'.
* This is the maximum number of consecutive replication errors
  (currently only for hot bucket replication) from a source peer
  to a specific target peer. Until this limit is reached, the
  source continues to roll hot buckets on streaming failures to
  this target. After the limit is reached, the source no
  longer rolls hot buckets if streaming to this specific target
  fails. This is reset if at least one successful (hot bucket)
  replication occurs to this target from this source.
* The special value of 0 turns off this safeguard; so the source
  always rolls hot buckets on streaming error to any target.
* This setting is dynamically reloadable and does not require restart
  of cluster peer.
* Default: 3

searchable_targets = <boolean>
* Only valid for 'mode=manager'.
* Tells the manager to make some replication targets searchable
  even while the replication is going on. This only affects
  hot bucket replication for now.
* Default: true

searchable_target_sync_timeout = <integer>
* Only valid for 'mode=peer'.
* How long, in seconds, that a hot bucket replication connection can be
  inactive before a searchable target flushes out any pending search
  related in-memory files.
* Regular syncing - when the data is flowing through
  regularly and the connection is not inactive - happens at a
  faster rate (default of 5 secs controlled by
  streamingTargetTsidxSyncPeriodMsec in indexes.conf).
* The special value of 0 turns off this timeout behavior.
* Default: 60

target_wait_time = <positive integer>
* Only valid for 'mode=manager'.
* Specifies the time, in seconds, that the manager waits for the
  target of a replication to register itself before it services
  the bucket again and potentially schedules another fixup.
* This setting is dynamically reloadable and does not require restart
  of cluster manager.
* Default: 150 (2 minutes 30 seconds)

summary_wait_time = <positive integer>
```

```
* Only valid when 'mode=manager' and 'summary_replication=true'.
* Specifies the time, in seconds, that the manager waits before
  scheduling fixups for a newly 'done' summary that transitioned
  from 'hot_done'. This allows for other copies of the 'hot_done'
  summary to also make their transition into 'done', avoiding
  unnecessary replications.
* Default: 660 (11 minutes)

commit_retry_time = <positive integer>
* Only valid for 'mode=manager'.
* Specifies the interval, in seconds, after which, if the last
  generation commit failed, the manager forces a retry. A retry is usually
  automatically kicked off after the appropriate events. This is just
  a backup to make sure that the manager does retry no matter what.
* Default: 300 (5 minutes)

percent_peers_to_restart = <integer between 0-100>
* Only valid for 'mode=manager'.
* Suggested percentage of maximum peers to restart for rolling-restart.
* Actual percentage may vary due to lack of granularity for smaller peer
  sets.
* Regardless of setting, a minimum of 1 peer is restarted per round.
* Default: 10

percent_peers_to_reload = <integer between 0-100>
* Only valid for 'mode=manager'.
* Suggested percentage of maximum peers to reload for bundle push.
* Actual percentage may vary due to lack of granularity for smaller peer
  sets.
* If set to 0, a minimum of 1 peer reloads the bundle per round.
* Default: 100

max_peers_to_download_bundle = <positive integer>
* Only valid for 'mode=manager'.
* The maximum number of peers to simultaneously download the configuration bundle
  from the manager, in response to the 'splunk apply cluster-bundle' command.
* When a peer finishes the download, the next waiting peer, if any, begins
  its download.
* If set to 0, all peers try to download at once.
* Default: 5

precompress_cluster_bundle = <boolean>
* Only valid for 'mode=manager'.
* Whether or not the manager compresses the configuration bundle files before
  it pushes them to peers.
* A value of "true" means the manager compresses the configuration bundle, which
  helps reduce network bandwidth consumption during the bundle push.
* Set this option to "true" only when SSL compression is off. Otherwise, the
  files will be compressed twice, which wastes CPU resources and does not save
  network bandwidth. To turn off SSL compression, set
  'allowSslCompression = false' in server.conf on the manager.
* Compressed bundles are denoted by the suffix ".bundle.gz". Uncompressed
  bundles use the suffix ".bundle".
* Default: true

auto_rebalance_primaries = <boolean>
* Only valid for 'mode=manager'.
* Specifies if the manager should automatically rebalance bucket
  primaries on certain triggers. Currently the only defined
  trigger is when a peer registers with the manager. When a peer
  registers, the manager redistributes the bucket primaries so the
  cluster can make use of any copies in the incoming peer.
```

```
* Default: true

rebalance_primaries_execution_limit = <non-negative integer>
* DEPRECATED. Use the 'rebalance_primaries_execution_limit_ms' setting instead.

rebalance_primaries_execution_limit_ms = <non-negative integer>
* Only valid for 'mode=manager'.
* The maximum period, in milliseconds, for one execution
  of the rebalance primary operation.
* This setting is useful for large clusters with large numbers of
  buckets, to prevent the primary rebalance operation from blocking
  other operations for significant amounts of time.
* The default value of 0 signifies auto mode.  In auto mode, the cluster
  manager uses the value of the 'service_interval' setting to determine the
  maximum time for the operation.
* Default: 0

commit_generation_execution_limit_ms = <non-negative integer>
* Only valid for 'mode=manager'.
* Specifies, in milliseconds, the maximum period for one execution
  of the committing pending generation.
* This setting is useful for large clusters with large numbers of
  buckets, to prevent the commit-generation operation from blocking
  other operations for significant amounts of time.
* The default value of 0 signifies auto mode. In auto mode, the cluster
  manager uses the value of the 'service_interval' setting to determine the
  maximum time for the operation.
* If 'service_interval' is auto, the range of this value will be within the
  range of 10ms and 25ms.
* Default: 0

idle_connections_pool_size = <integer>
* Only valid for 'mode=manager'.
* Specifies how many idle http(s) connections that should be kept
  alive to reuse.
* Reusing connections improves the time it takes to send messages to peers
  in the cluster.
* -1 corresponds to "auto", letting the manager determine the
  number of connections to keep around based on the number of peers in the
  cluster.
* Default: -1

use_batch_mask_changes = <boolean>
* Only valid for 'mode=manager'.
* Specifies if the manager should process bucket mask changes in
  batch or individually one by one.
* Set to 'false' when there are version 6.1 peers in the cluster for
  backwards compatibility.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: true

service_jobs_msec = <positive integer>
* Only valid for 'mode=manager'.
* The maximum time, in milliseconds, that the cluster manager spends in servicing
  finished jobs for each service call. Increase this if the 'metrics.log'
  file has very high 'current_size' values.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 100 (0.1 seconds)

summary_replication = true|false|disabled
```

```
* Valid for both 'mode=manager' and 'mode=peer'.
* Cluster Manager:
  If set to "true", summary replication is enabled.
  If set to "false", summary replication is disabled, but can be enabled
  at runtime.
  If set to disabled, summary replication is disabled. Summary replication
  cannot be enabled at runtime.
* Peers:
  If set to "true" or "false", there is no effect. The indexer follows
  whatever setting is on the Cluster Manager.
  If set to "disabled", summary replication is disabled. The indexer does
  no scanning of summaries (increased performance during peers joining
  the cluster for large clusters).
* Default: false (for both Cluster Manager and Peers)

rebalance_threshold = <decimal>
* Only valid for 'mode=manager'.
* During rebalancing buckets amongst the cluster, this threshold is
  used as a percentage to determine when the cluster is balanced.
* Valid values are between 0.10 and 1.00.
* 1.00 is 100% indexers fully balanced.
* Default: 0.90

max_auto_service_interval = <positive integer>
* Only valid for 'mode=manager'.
* Only valid when 'service_interval' is in auto mode.
  For example service_interval=0.
* Indicates the maximum value, in seconds, that service interval is
  bounded by when the 'service_interval' is in auto mode. If the
  previous service call took more than 'max_auto_service_interval'
  seconds, the next service call runs after 'max_auto_service_interval'
  seconds.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: 1

buckets_to_summarize = <primaries|primaries_and_hot|all>
* Only valid for 'mode=manager'.
* Determines which buckets are sent to '| summarize' searches (searches that
  build report acceleration and data models).
* Set to "primaries" to apply only to primary buckets.
* Set to "primaries_and_hot" to also apply it to all hot searchable
  buckets.
* Set to "all" to apply the search to all buckets.
* If "summary_replication' is enabled, then 'buckets_to_summarize' defaults
  to "primaries_and_hot".
* Do not change this setting without first consulting with Splunk Support.
* Default: primaries

maintenance_mode = <boolean>
* Only valid for 'mode=manager'.
* To preserve the maintenance mode setting in case of manager
  restart, the manager automatically updates this setting in the
  etc/system/local/server.conf file whenever you enable or disable
  maintenance mode using the CLI or REST API.
* NOTE: Do not manually update this setting. Instead, use the CLI or REST
  API to enable or disable maintenance mode.

backup_and_restore_primaries_in_maintenance = <boolean>
* Only valid for 'mode=manager'.
* Determines whether the manager performs a backup/restore of bucket
  primary masks during maintenance mode or rolling-restart of cluster peers.
```

```
* A value of "true" means, restoration of primaries occurs automatically when
  the peers rejoin the cluster after a scheduled restart or upgrade.
* Default: false

max_primary_backups_per_service = <zero or positive integer>
* Only valid for 'mode=manager'.
* For use with the 'backup_and_restore_primaries_in_maintenance' setting.
* Determines the number of peers for which the manager backs up primary
  masks for each service call.
* The special value of 0 causes the manager to back up the primary masks for
  all peers in a single service call.
* Default: 10

allow_default_empty_p4symmkey = <boolean>
* Only valid for 'mode=manager'.
* Affects behavior of manager during start-up, if 'pass4SymmKey'resolves
  to the null string or the default password ("changeme").
* A value of "true" means the manager posts a warning but still launches.
* A value of "false" means the manager posts a warning and stops.
* Default: false

register_replication_address = <string>
* Only valid for 'mode=peer'.
* This is the address on which a peer is available for accepting
  replication data. This is useful in the cases where a peer host machine
  has multiple interfaces and only one of them can be reached by another
  splunkd instance.
* This must be either an IP address or fully qualified machine/domain name.
* No default.

register_forwarder_address = <string>
* Only valid for 'mode=peer'.
* This is the address on which a peer is available for accepting
  data from forwarder.This is useful in the cases where a splunk host
  machine has multiple interfaces and only one of them can be reached by
  another splunkd instance.
* This must be either an IP address or fully qualified machine/domain name.
* No default.

register_search_address = <string>
* Only valid for 'mode=peer'.
* This is the address that advertises the peer to search heads.
  This is useful in the cases where a splunk host machine has multiple
  interfaces and only one of them can be reached by another splunkd
  instance.
* This must be either an IP address or fully qualified machine/domain name.
* No default.

executor_workers = <positive integer>
* Only valid if 'mode=manager' or 'mode=peer'.
* Number of threads that can be used by the clustering thread pool.
* A value of 0 defaults to 1.
* Default: 10
* This setting reloads automatically and does not require a restart.

local_executor_workers = <positive integer>
* DEPRECATED.

manual_detention = on|on_ports_enabled|off
* Only valid for 'mode=peer'.
* Puts this peer node in manual detention.
* Default: off
```

```
allowed_hbmiss_count = <positive integer>
* Only valid for 'mode=peer'.
* Sets the count of number of heartbeat failures before the peer node
  disconnects from the manager.
* Default: 3

buckets_per_addpeer = <non-negative integer>
* Only valid for 'mode=peer'.
* Controls the number of buckets for each add peer request.
* When a peer is added or re-added to the cluster, it sends the manager
  information for each of its buckets. Depending on the number of buckets,
  this could take a while. For example, a million buckets could require
  more than a minute of the manager's processing time. To prevent the manager
  from being occupied by this single task too long, you can use this setting to
  split large numbers of buckets into several "batch-add-peer" requests.
* If it is invalid or non-existent, the peer uses the default setting instead.
* If it is set to 0, the peer sends only one request with all buckets
  instead of batches.
* You do not need to restart the cluster peer when making changes to
  this setting. This setting reloads automatically.
* Default: 1000

heartbeat_period = <non-zero positive integer>
* Controls the interval, in seconds, with which the peer attempts
  to send heartbeats to the manager node.
* Only valid for 'mode=peer'.
* Default: 1

auto_fix_corrupt_buckets = <boolean>
* Only valid for 'mode=manager'.
* If set to "true", the manager performs automatic fixup of
  corrupted buckets.
* To fix a corrupted bucket, the manager fetches the current
  searchable events count from all copies of the bucket. It then
  selects a copy with the largest searchable events count as
  the canonical copy. The manager tells all peers holding copies
  with smaller events counts to discard their copies. The cluster
  then replicates the canonical copy as needed until the cluster
  holds the configured replication factor number of copies.
* The manager's peer nodes must be running a version that supports
  this feature.
* This feature is available only for non-SmartStore buckets.
  SmartStore buckets require manual fixup.
* Default: true

bucketsize_mismatch_strategy = smallest|largest
* Only valid for 'mode=manager'.
* This setting determines how the manager decides which target peer's bucket copy
  is retained on the cluster when the source peer is not present at the time
  that a hot bucket is rolled, and there is a bucket size mismatch between
  the target peers
* A value of "largest" means the largest copy of the bucket on any target
  peer gets propagated to the other peers through fixups, overwriting all other
  copies.
* A value of "smallest" means the smallest copy of the bucket on any target
  peer gets propagated to the other peers through fixups, overwriting all other
  copies.
* Do not alter this value without contacting Splunk Support.
* Default: largest

remote_storage_upload_timeout = <non-zero positive integer>
```

```
* Only valid for 'mode=peer'.
* For a remote storage enabled index, this setting specifies the interval
  in seconds, after which target peers assume responsibility for
  uploading a bucket to the remote storage, if they do not hear from
  the source peer.
* This setting is dynamically reloadable and does not require restart
  of cluster peer.
* Default: 60 (1 minute)

report_remote_storage_bucket_upload_to_targets = <boolean>
* Only valid for 'mode=peer' or 'mode=manager'.
* For a remote storage enabled index, this setting specifies whether
  the source peer reports the successful bucket upload to target peers.
  This notification is used by target peers to cancel their upload timers
  and synchronize their bucket state with the uploaded bucket on remote
  storage.
* Do not change the value from the default unless instructed by
  Splunk Support.
* You do not need to restart the cluster manager when making changes to
  this setting. This setting reloads automatically.
* Default: false

remote_storage_retention_period = <non-zero positive integer>
* Only valid for 'mode=manager'.
* The interval, in seconds, after which the manager checks buckets
  in remote storage enabled indexes against the retention policy.
  It then triggers freeze operations on the cluster peers as necessary.
* This setting also determines the time that the manager waits
  following a restart before checking retention policy.
* For details on retention policies, examine the
  'maxGlobalDataSizeMB' and 'frozenTimePeriodInSecs' settings.
* This setting is dynamically reloadable and does not require restart
  of cluster manager.
* Default: 900 (15 minutes)

recreate_bucket_attempts_from_remote_storage = <positive integer>
* Only valid for 'mode=manager'.
* Controls the number of attempts the manager makes to recreate the
  bucket of a remote storage enabled index on a random peer node
  in these scenarios:
    * Manager detects that the bucket is not present on any peers.
    * A peer informs the manager about the bucket as part of the
      re-creation of an index.
      See 'recreate_index_attempts_from_remote_storage' setting.
* Re-creation of the bucket involves the following steps:
    1. Manager provides a random peer with the bucket ID of the bucket that
       needs to be recreated.
    2. Peer fetches the metadata of the bucket corresponding to this
       bucket ID from the remote storage.
    3. Peer creates a bucket with the fetched metadata locally and informs
       the manager that a new bucket has been added.
    4. Manager initiates fix-ups to add the bucket on the necessary number
       of additional peers to match the replication and search factors.
* If set to 0, disables the re-creation of the bucket.
* Default: 10

recreate_bucket_max_per_service = <positive integer>
* Only valid for 'mode=manager'.
* Only applies when using remote storage enabled indexes.
* Controls the maximum number of buckets that the cluster can recreate
  during a service interval.
* Do not change the value from the default unless instructed by
```

```
  Splunk Support.
* If set to 0, recreating buckets will go at full speed.
* Default: 20000

recreate_bucket_fetch_manifest_batch_size = <positive integer>
* Only valid for 'mode=manager'.
* Controls the maximum number of bucket IDs for which a peer
  attempts to initiate a parallel fetch of manifests at a time
  in the process of recreating buckets that have been
  requested by the manager.
* The manager sends this setting to all the peers that are
  involved in the process of recreating the buckets.
* Default: 50

recreate_index_attempts_from_remote_storage = <positive integer>
* Only valid for 'mode=manager'.
* Controls the number of attempts the manager makes to recreate
  a remote storage enabled index on a random peer node when the manager
  is informed about the index by a peer.
* Re-creation of an index involves the following steps:
    1. Manager pushes a bundle either when it is ready for service or
       when requested by the user.
    2. Manager waits for the bundle to be applied successfully on the
       peer nodes.
    3. Manager requests that a random peer node provide it with the list
       of newly added remote storage enabled indexes.
    4. Manager distributes a subset of indexes from this list to
       random peer nodes.
    5. Each of those peer nodes fetches the list of bucket IDs for the
       requested index from the remote storage and provides it
       to the manager.
    6. The manager uses the list of bucket IDs to recreate the buckets.
       See recreate_bucket_attempts_from_remote_storage.
* If set to 0, disables the re-creation of the index.
* Default: 10

recreate_index_fetch_bucket_batch_size = <positive integer>
* Only valid for 'mode=manager'.
* Controls the maximum number of bucket IDs that the manager
  requests a random peer node to fetch from remote storage as part of
  a single transaction for a remote storage enabled index.
  The manager uses the bucket IDs for re-creation of the index.
  See the 'recreate_index_attempts_from_remote_storage' setting.
* Default: 2000

use_batch_remote_rep_changes = <boolean> or <positive integer>
* Only valid for 'mode=manager'.
* Specifies whether the manager processes bucket copy changes (to meet
  replication_factor and search_factor) in batch or individually.
* Also controls the maximum number of bucket replications that are processed in
  one replication batch.
* This is applicable to buckets belonging to
  remote storage enabled indexes only.
* Do not change this setting without consulting with Splunk Support.
* This setting is dynamically reloadable and does not require restart
  of cluster manager.
* If 'false' is specified, batching of buckets would be turned off
* If 'true' is specified, batching of buckets would be turned on, the maximum
  number of buckets processed per batch would be the system default (1000)
* If 0 is specified, batching of buckets would be turned off
* If <any non zero positive integer> is specified, batching of buckets
  would be turned on, and the maximum number of buckets processed per batch
```

```
  would be the value of the integer specified
* Default: 1000

max_peer_batch_rep_load = <positive integer>
* Only valid for 'mode=manager'.
* This setting is applicable to buckets belonging to
  remote storage enabled indexes only.
* Only valid when 'use_batch_remote_rep_changes=true'
* This setting specifies the maximum number of concurrent batch replications
  that a peer node can take part in, as a source.
* Default: 5

enable_primary_fixup_during_maintenance = <boolean>
* Only valid for 'mode=manager'.
* Specifies whether the manager performs primary fixups during
  maintenance mode. This gets overridden by searchable rolling restart.
* This setting is dynamically reloadable and does not require restart
  of cluster manager.
* Default: true

freeze_during_maintenance = <boolean>
* Only valid for 'mode=manager'.
* Specifies whether the manager will tell peers to freeze buckets during
  maintenance mode.
* This setting is dynamically reloadable and does not require restart
  of cluster manager.
* Default: false

assign_primaries_to_all_sites = <boolean>
* Only valid for 'mode=manager' and 'multisite=true'.
* Controls how the manager assigns bucket primary copies on a
  multisite cluster.
* If set to "true", the manager assigns a primary copy to each site
  defined in 'available_sites', as well as site0.
* If set to "false":
  * The manager assigns a primary copy only to sites with a search head.
  * Sites without search heads do not get primary copies.
  * When a new site with a search head joins the cluster, or an existing
    site attains its first search head, the cluster manager gradually
    adds all buckets in the cluster to its fixup list to ensure that the
    site will be populated with primaries.
  * If a site loses its search heads, no action is taken to remove
    existing primaries from the site.
* Setting this parameter to 'false' can significantly reduce the work of primary
  assignments, especially if search heads are only on site0 and
  search affinity is disabled.
* Default: false

log_bucket_during_addpeer = <boolean>
* Only valid for 'mode=manager'.
* Controls the log level for bucket information during add-peer activities.
* If set to "true", the manager logs bucket information to INFO level under
  CMMaster componenet during add-peer.
* If set to "false", the manager logs bucket information to DEBUG level under
  CMMaster component during add-peer.
* Set to 'false' for large clusters with large numbers of buckets.
* Default: false

max_concurrent_peers_joining = <nonzero integer>
* Only valid for 'mode=manager'.
* Limits the number of peers that are allowed to join the cluster at one time.
* The peer reports its buckets to the cluster manager upon first establishing a
```

```
  connection with the manager, and it finishes joining the cluster when all of
  its buckets have been reported.
* Once this limit is hit, any remaining peers check at one second intervals
  for an available slot to join the cluster.
* By limiting the number of peers that can join simultaneously, this setting
  can facilitate faster restart for some peers, thus more quickly restoring
  partial ingest to the cluster.
* Default: 10

enable_parallel_add_peer = <boolean>
* Only valid for 'mode=manager'.
* Enables the cluster manager to accept and process multiple 'add peer' requests
  in parallel.
* The upper limit of concurrent 'add peer' requests that the manager can handle is
  limited by the 'max_concurrent_peers_joining setting'.
* When this feature is enabled, the largest recommended value for
  'max_concurrent_peers_joining' is half the number of CPU cores of
  the indexer. For example, if the indexer has 24 CPU cores, the largest
  recommended value for 'max_concurrent_peers_joining' is 12.
* This setting is useful for clusters with large numbers of buckets
  and large numbers of indexers.  It also improves the responsiveness
  of the cluster manager, helping to prevent unnecessary timeouts.
* Default: true

buckets_status_notification_batch_size = <positive integer>
* Only valid for 'mode=peer'.
* Controls the number of existing buckets IDs that the peer
  reports to the manager every notify_scan_period seconds.
  The manager then initiates fix-ups for these buckets.
* CAUTION: Do not modify this setting without guidance from
  Splunk personnel.
* Default: 1000

notify_scan_period = <non-zero positive integer>
* Only valid for 'mode=peer'.
* Controls the frequency, in seconds, that the indexer handles
  the following options:
  1. summary_update_batch_size
  2. summary_registration_batch_size
* CAUTION: Do not modify this setting without guidance from
  Splunk personnel.
* Default: 10

notify_scan_min_period = <non-zero positive integer>
* Only valid for 'mode=peer'.
* Controls the highest frequency, in milliseconds, that the indexer
  scans summary folders
  for summary updates/registrations. The notify_scan_period temporarily
  becomes notify_scan_min_period when there are more summary
  updates/registration events to be processed but has been limited due to
  either summary_update_batch_size or summary_registration_batch_size.
* CAUTION: Do not modify this setting without guidance from Splunk
  personnel.
* Default: 10

notify_buckets_period = <non-zero positive integer>
* Only valid for 'mode=peer'.
* Controls the frequency, in milliseconds, that the indexer handles
  buckets_status_notification_batch_size
* CAUTION: Do not modify this setting without guidance from
  Splunk personnel.
* Default: 10
```

```
summary_update_batch_size = <non-zero positive integer>
* Only valid for 'mode=peer'.
* Controls the number of summary updates the indexer sends per batch to
  the manager every notify_scan_period.
* CAUTION: Do not modify this setting without guidance from
  Splunk personnel.
* Default: 10

summary_registration_batch_size = <non-zero positive integer>
* Only valid for 'mode=peer'.
* Controls the number of summaries that get asynchronously registered
  on the indexer and sent as a batch to the manager every
  notify_scan_period.
* Caution: Do not modify this setting without guidance from Splunk personnel.
* Default: 1000

enableS2SHeartbeat = <boolean>
* Only valid for 'mode=peer'.
* Splunk software monitors each replication connection for
  presence of a heartbeat, and if the heartbeat is not seen for
  's2sHeartbeatTimeout' seconds, it closes the connection.
* Default: true

s2sHeartbeatTimeout = <integer>
* This specifies the global timeout value, in seconds, for monitoring
  heartbeats on replication connections.
* Splunk software closes a replication connection if heartbeat is not seen
  for 's2sHeartbeatTimeout' seconds.
* Replication source sends heartbeats every 30 seconds.
* Default: 600 (10 minutes)

throwOnBucketBuildReadError = <boolean>
* Valid only for 'mode=peer'.
* A value of "true" means index clustering peer throws an exception if it
  encounters a journal read error while building the bucket for a new
  searchable copy. It also throws all the search & other files generated
  so far in this particular bucket build.
* A value of "false" means index clustering peer just logs the error and preserves
  all the search & other files generated so far & finalizes them as it
  cannot proceed further with this bucket.
* Default: false

cluster_label = <string>
* Only valid for 'mode=manager'.
* This specifies the label of the indexer cluster

warm_bucket_replication_pre_upload = <boolean>
* Valid only for 'mode=peer'.
* This setting applies to remote storage enabled indexes only.
* A value of "true" means the target peers replicate all warm bucket contents
  when necessary for bucket-fixing if the source peer has not yet uploaded
  the bucket to remote storage.
* A value of "false" means the target peers never replicate warm bucket contents.
* In either case the target peers replicate metadata only, once the source peer
  uploads the bucket to remote storage.
* Default: false

bucketsize_upload_preference = largest | smallest
* Valid only for 'mode=peer'.
* This setting applies to remote storage enabled indexes only.
* This setting determines the criteria a target peer uses when deciding whether to
```

```
  overwrite a bucket copy uploaded to remote storage by another target peer. Target
  peers never overwrite copies uploaded by a source peer.
* When "largest" is selected, the largest copy of the bucket on any target
  peer gets uploaded.
* When "smallest" is selected, the smallest copy of the bucket on any target
  peer gets uploaded.
* Note, this and "bucketsize_mismatch_strategy" should follow same scheme.
* Do not alter this value without contacting Splunk Support.
* Default: largest

upload_rectifier_timeout_secs = <unsigned integer>
* Valid only for 'mode=peer'.
* This setting applies to remote storage enabled indexes only.
* When a peer uploads a bucket copy to remote storage, it checks, after a ,
  timeout based on the value of this setting, to determine whether another
  peer overwrote the copy.
* Depending on the value of "bucketsize_upload_preference" it will determine
  if the bucket needs to be re-uploaded.
* This setting controls the timeout that the peer waits before checking.
* Default: 2

localization_based_primary_selection = [disabled|auto]
* Only valid for 'mode=manager'.
* This setting determines the behavior of the cluster manager when
  assigning primacy to SmartStore bucket copies.  If set to 'auto',
  the cluster manager examines each copy's localization flag and
  assigns primacy to a bucket copy, if any, with existing localized
  content. If multiple peers have localized content for the bucket,
  the cluster manager assigns primacy to the copy on the peer with
  the least number of total primary bucket copies.
* If set to 'disabled' the localization flags of the buckets are not
  taken into account during primary assignment.
* Default: disabled

localization_update_batch_size = <non-zero positive integer>
* Only valid for 'mode=peer'.
* Controls the number of bucket localization updates the peer sends
  per batch to the manager every heartbeat_period.
* CAUTION: Do not modify this setting without guidance from
  Splunk personnel.
* Default: 1000

enable_encrypt_bundle = <boolean>
* Whether or not an indexer cluster manager encrypts sensitive fields from the
  'encrypt_fields' setting when it creates an indexer clustering bundle.
* A value of "true" means that indexer clustering bundle encryption is enabled.
* A value of "false" means that indexer clustering bundle encryption is disabled.
* NOTE: If you disable this setting, confirm that all fields in files in the
  configuration bundle on the manager node are not encrypted before you deploy
  the bundle to the peer nodes.
* Default: true

[clustermanager:<cm-nameX>]
* Valid for 'mode=searchhead' when the search head belongs to multiple indexer
  clusters.
* Valid for 'mode=manager' and 'manager_switchover_mode=auto|manual'.

master_uri = <uri>
* DEPRECATED. Use the 'manager_uri' setting instead.

manager_uri = <string>
* There are two uses for this setting, one for 'mode=searchhead',
```

```
  and another for 'mode=manager'.
* For 'mode=searchhead':
  * This represents the URI of the cluster manager that this
    search head should connect to.
* For 'mode=manager':
  * Only valid if 'manager_switchover_mode=auto|manual'
  * This setting is the URI for the manager described by this stanza.
  * Each cluster manager must include a separate copy of this stanza
    for each manager in the cluster, including itself. For example,
    if the cluster has three managers, each manager's configuration
    must include an identical set of three stanzas, one for each manager.

pass4SymmKey = <string>
* Secret shared among the nodes in the cluster to prevent any
  arbitrary node from connecting to the cluster. If a search head
  is not configured with the same secret as the manager,
  it not be able to communicate with the manager.
* If it is not present here, the key in the clustering stanza is used.
  If it is not present in the clustering stanza, the value in the general
  stanza is used.
* Ignored when 'mode=manager' and 'manager_switchover_mode=auto|manual'.
  In this mode, the 'pass4SymmKey' is picked up from the [clustering] stanza
  for connecting to all the managers defined in the [clustermanager] stanzas.
* Unencrypted passwords must not begin with "$1$", as this is used by
  Splunk software to determine if the password is already encrypted.
* No default.

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' should be for a
  particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length than
  what you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* Default: 12

site = <site-id>
* Specifies the site this search head belongs to for this particular manager
  when multisite is enabled (see below).
* Valid values for site-id include site0 to site63.
* The special value "site0" disables site affinity for a search head in a
  multisite cluster. It is only valid for a search head.
* Ignored when 'mode=manager' and 'manager_switchover_mode=auto|manual'.
  In this mode, site id is picked up from the [general] stanza.

multisite = <boolean>
* Turns on the multisite feature for this manager_uri for the search head.
* Make sure the manager has the multisite feature turned on.
* Make sure you specify the site in case this is set to true. If no
  configuration is found in the [clustermanager] stanza, the search head defaults
  to any value for 'site' that might be defined in the [general]
  stanza.
* Ignored when 'mode=manager' and 'manager_switchover_mode=auto|manual'.
  In this mode, the multisite flag is picked up from the [clustering] stanza.
* Default: false

[replication_port://<port>]
# Configure Splunk to listen on a given TCP port for replicated data from
# another cluster member.
# If 'mode=peer' is set in the [clustering] stanza at least one
```

```
# 'replication_port' must be configured and not disabled.

disabled = <boolean>
* Set to true to disable this replication port stanza.
* Default: false

listenOnIPv6 = no|yes|only
* Toggle whether this listening port listens on IPv4, IPv6, or both.
* If not present, the setting in the [general] stanza is used.

acceptFrom = <network_acl> ...
* Lists a set of networks or addresses from which to accept connections.
* Separate multiple rules with commas or spaces.
* Each rule can be in one of the following formats:
    1. A single IPv4 or IPv6 address (examples: "10.1.2.3", "fe80::4a3")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
       (examples: "10/8", "192.168.1/24", "fe80:1234/32")
    3. A DNS name, possibly with a "*" used as a wildcard
       (examples: "myhost.example.com", "*.splunk.com")
    4. "*", which matches anything
* You can also prefix an entry with '!' to cause the rule to reject the
  connection. The input applies rules in order, and uses the first one that
  matches.
  For example, "!10.1/16, *" allows connections from everywhere except
  the 10.1.*.* network.
* Default: "*" (accept from anywhere)

[replication_port-ssl://<port>]
* This configuration is same as the [replication_port] stanza above,
  but uses SSL.

disabled = <boolean>
* Set to true to disable this replication port stanza.
* Default: false

listenOnIPv6 = no|yes|only
* Toggle whether this listening port listens on IPv4, IPv6, or both.
* If not present, the setting in the [general] stanza is used.

acceptFrom = <string> ...
* This setting is the same as the setting in the [replication_port] stanza.

serverCert = <string>
* Full path to file containing private key and server certificate.
* The <path> must refer to a PEM format file.
* No default.

sslPassword = <string>
* Server certificate password, if any.
* No default.

password = <string>
* DEPRECATED; use 'sslPassword' instead.

rootCA = <string>
* DEPRECATED; use '[sslConfig]/sslRootCAPath' instead.
* Full path to the root CA (Certificate Authority) certificate store.
* The <path> must refer to a PEM format file containing one or more root CA
  certificates concatenated together.
* No default.

cipherSuite = <string>
```

```
* If set, uses the specified cipher string for the SSL connection.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Default: The default can vary (See the cipherSuite setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the current default)

sslVersions = <comma-separated list>
* Comma-separated list of SSL versions to support.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list but
  does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Default: The default can vary (See the sslVersions setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the current default)

ecdhCurves = <comma separated list>
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* The server supports only the curves specified in the list.
* Splunk software only supports named curves specified
  by their SHORT names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Default: The default can vary (See the 'ecdhCurves' setting in
  the $SPLUNK_HOME/etc/system/default/server.conf file for the current default)

dhFile = <string>
* PEM format Diffie-Hellman parameter file name.
* DH group size should be no less than 2048bits.
* This file is required in order to enable any Diffie-Hellman ciphers.
* No default.

dhfile = <string>
* DEPRECATED; use 'dhFile' (with a capital F) instead.

supportSSLV3Only = <boolean>
* DEPRECATED. SSLv2 is now always disabled. The exact set of SSL versions
  allowed is now configurable by using the 'sslVersions' setting.

useSSLCompression = <boolean>
* If true, enables SSL compression.
* Default: true

compressed = <boolean>
* DEPRECATED. Use 'useSSLCompression' instead.
* Used only if 'useSSLCompression' is not set.

requireClientCert = <boolean>
* Requires that any peer that connects to replication port has a certificate
  that can be validated by certificate authority specified in rootCA.
* Default: false

allowSslRenegotiation = <boolean>
* In the SSL protocol, a client may request renegotiation of the connection
  settings from time to time.
* Setting this to false causes the server to reject all renegotiation
```

```
  attempts, breaking the connection.  This limits the amount of CPU a
  single TCP connection can use, but it can cause connectivity problems
  especially for long-lived connections.
* Default: true

sslCommonNameToCheck = <comma-separated list>
* Optional.
* Check the common name of the client's certificate against this list of names.
* Separate multiple common names with commas.
* 'requireClientCert' must be set to "true" for this setting to work.
* No default.

sslAltNameToCheck = <comma-separated list>
* Optional.
* Check the alternate name of the client's certificate against this list
  of names.
* If there is no match, assume that Splunk is not authenticated against this
  server.
* Separate multiple alternate names with commas.
* 'requireClientCert' must be set to "true" for this setting to work.
* No default.
```

Introspection settings

```
[introspection:generator:disk_objects]
* For 'introspection_generator_addon', packaged with Splunk; provides the
  data ("i-data") consumed, and reported on, by 'introspection_viewer_app'
  (due to ship with a future release).
* This stanza controls the collection of i-data about: indexes; bucket
  superdirectories (homePath, coldPath, ...); volumes; search dispatch
  artifacts.
* On forwarders the collection of index, volumes and dispatch disk objects
  is disabled.

acquireExtra_i_data = <boolean>
* If true, extra Disk Objects i-data is emitted; you can gain more insight
  into your site, but at the cost of greater resource consumption both
  directly (the collection itself) and indirectly (increased disk and
  bandwidth utilization, to store the produced i-data).
* Consult documentation for the list of regularly emitted Disk Objects
  i-data, and extra Disk Objects i-data, appropriate to your release.
* Default: false

collectionPeriodInSecs = <positive integer>
* Controls frequency of Disk Objects i-data collection; higher frequency
  (hence, smaller period) gives a more accurate picture, but at the cost of
  greater resource consumption both directly (the collection itself) and
  indirectly (increased disk and bandwidth utilization, to store the
  produced i-data).
* Default: 600 (10 minutes)

[introspection:generator:disk_objects__indexes]
  * This stanza controls the collection of i-data about indexes.
  * Inherits the values of 'acquireExtra_i_data' and 'collectionPeriodInSecs'
    attributes from the 'introspection:generator:disk_objects' stanza, but
    may be enabled/disabled independently of it.
  * This stanza should only be used to force collection of i-data about
    indexes on dedicated forwarders.
  * Default: Data collection is disabled on universal forwarders and
```

```
    enabled on all other installations.

[introspection:generator:disk_objects__volumes]
  * This stanza controls the collection of i-data about volumes.
  * Inherits the values of 'acquireExtra_i_data' and 'collectionPeriodInSecs'
    settings from the 'introspection:generator:disk_objects' stanza, but
    may be enabled/disabled independently of it.
  * This stanza should only be used to force collection of i-data about
    volumes on dedicated forwarders.
  * Default: Data collection is disabled on universal forwarders and
    enabled on all other installations.

[introspection:generator:disk_objects__dispatch]
  * This stanza controls the collection of i-data about search dispatch
    artifacts.
  * Inherits the values of 'acquireExtra_i_data' and 'collectionPeriodInSecs'
    settings from the 'introspection:generator:disk_objects' stanza, but
    may be enabled/disabled independently of it.
  * This stanza should only be used to force collection of i-data about
    search dispatch artifacts on dedicated forwarders.
  * Default: Data collection is disabled on universal forwarders and
    enabled on all other installations.

[introspection:generator:disk_objects__fishbucket]
* This stanza controls the collection of i-data about:
  $SPLUNK_DB/fishbucket, where per-input status of file-based
  inputs is persisted.
* Inherits the values of 'acquireExtra_i_data' and 'collectionPeriodInSecs'
  settings from the 'introspection:generator:disk_objects' stanza, but may
  be enabled/disabled independently of it.

[introspection:generator:disk_objects__bundle_replication]
* This stanza controls the collection of i-data about:
  bundle replication metrics of distributed search
* Inherits the values of 'acquireExtra_i_data' and 'collectionPeriodInSecs'
  settings from the 'introspection:generator:disk_objects' stanza, but may
  be enabled/disabled independently of it.

[introspection:generator:disk_objects__partitions]
* This stanza controls the collection of i-data about: disk partition space
  utilization.
* Inherits the values of 'acquireExtra_i_data' and 'collectionPeriodInSecs'
  settings from the 'introspection:generator:disk_objects' stanza, but may
  be enabled/disabled independently of it.

[introspection:generator:disk_objects__summaries]
* Introspection data about summary disk space usage. Summary disk usage
  includes both data model and report summaries. The usage is collected
  for each summaryId, locally at each indexer.

disabled = <boolean>
* If not specified, inherits the value from
  [introspection:generator:disk_objects] stanza.

collectionPeriodInSecs = <positive integer>
* Controls frequency, in seconds, of Disk Objects - summaries
  collection; higher frequency (hence, smaller period) gives a more accurate
  picture, but at the cost of greater resource consumption directly
  (the summaries collection itself);
  it is not recommended for a period less than 15 minutes.
* If you enable summary collection, the first collection happens 5 minutes
  after the Splunk instance is started. For every subsequent collection, this
```

```
  setting is honored.
* If 'collectionPeriodInSecs' is smaller than 5 * 60, it resets to
  30 minutes internally.
* Set to (N*300) seconds. Any remainder is ignored.
* Default: 1800 (30 minutes)

[introspection:generator:resource_usage]
* For 'introspection_generator_addon', packaged with Splunk; provides the
  data ("i-data") consumed, and reported on, by 'introspection_viewer_app'
  (due to ship with a future release).
* "Resource Usage" here refers to: CPU usage; scheduler overhead; main
  (physical) memory; virtual memory; pager overhead; swap; I/O; process
  creation (a.k.a. forking); file descriptors; TCP sockets; receive/transmit
  networking bandwidth.
* Resource Usage i-data is collected at both hostwide and per-process
  levels; the latter, only for processes associated with this SPLUNK_HOME.
* Per-process i-data for Splunk search processes include additional,
  search-specific, information.

acquireExtra_i_data = <boolean>
* A value of "true" means extra Resource Usage i-data is emitted; you can gain
  more insight into your site, but at the cost of greater resource
  consumption both directly (the collection itself) and indirectly
  (increased disk and bandwidth utilization, to store the produced i-data).
* Consult the documentation for list of regularly emitted Resource Usage
  i-data, and extra Resource Usage i-data, appropriate to your release.
* Default: false

collectionPeriodInSecs = <positive integer>
* Controls frequency of Resource Usage i-data collection; higher frequency
  (hence, smaller period) gives a more accurate picture, but at the cost of
  greater resource consumption both directly (the collection itself) and
  indirectly (increased disk and bandwidth utilization, to store the
  produced i-data).
* Default (on universal forwarders): 600 (10 minutes)
* Default (on all other Splunk platform instance types): 10 (1/6th of a minute)

disabled = <boolean>
* Disables Resource Usage data collection.
* Default (on universal forwarders): true
* Default (on all other Splunk platform instance types): false

[introspection:generator:resource_usage__iostats]
* This stanza controls the collection of i-data about: IO Statistics data
* "IO Statistics" here refers to: read/write requests; read/write sizes;
  io service time; cpu usage during service
* IO Statistics i-data is sampled over the collectionPeriodInSecs
* Does not inherit the value of the 'collectionPeriodInSecs' setting from the
  'introspection:generator:resource_usage' stanza, and may be enabled/disabled
  independently of it.

collectionPeriodInSecs = <positive integer>
* Controls interval of IO Statistics i-data collection; higher intervals
  gives a more accurate picture, but at the cost of greater resource consumption
  both directly (the collection itself) and indirectly (increased disk and
  bandwidth utilization, to store the produced i-data).
* Default: 60 (1 minute)

disabled = <boolean>
* Disables IO Statistics data collection.
* Default (on universal forwarders): true
* Default (on all other Splunk platform instance types): false
```

```
[introspection:generator:kvstore]
* For 'introspection_generator_addon', packaged with Splunk Enterprise.
* "KV Store" here refers to: statistics information about KV Store process.

serverStatsCollectionPeriodInSecs = <positive integer>
* The frequency, in seconds, of KV Store server status collection.
* Default: 27

operationStatsCollectionPeriodInSecs = <positive integer>
* The frequency, in seconds, of KV Store operation statistics collection (currentOp).
* Default: 60 seconds

collectionStatsCollectionPeriodInSecs = <positive integer>
* The frequency, in seconds, of KV Store db statistics collection.
* Default: 600 (10 minutes)

profilingStatsCollectionPeriodInSecs = <positive integer>
* The frequency, in seconds, of KV Store profiling data collection.
* Default: 5 seconds

rsStatsCollectionPeriodInSecs = <positive integer>
* The frequency, in seconds, of KV Store replica set stats collection
* Default: 60 seconds

[introspection:distributed-indexes]
* This stanza controls the collection of information for distributed indexes.

disabled = <boolean>
* Whether or not collection of introspection information on distributed
  indexes is disabled.
* A value of "false" means information on distributed indexes is collected.
* This provides additional insight into index usage at the cost of greater
  resource consumption.
* Default: true

collectionPeriodInSecs = <positive integer>
* The frequency, in seconds, of distributed index data collection.
  Shorter intervals provide more accurate results, at the cost of
  greater resource consumption.
* Must be set between 300 (5 minutes) and 86400 (24 hours).
* Default: 3600 (60 minutes)

collectLocalIndexes = <boolean>
* This setting determines whether the search head retrieves index metadata,
  such as current size and event count.
* In single-instance configurations, where the instance serves as both search
  head and indexer, set the value to "true", so that the local index metadata
  is retrieved.
* In distributed search deployments, with separate search heads and indexers,
  set the value to "false" to retrieve metadata only from indexes on the indexers.
* Default: false
```

Settings used to control commands started by Splunk

```
[commands:user_configurable]

prefix = <string>
* All non-internal commands started by splunkd are prefixed with this
```

```
  string, allowing for "jailed" command execution.
* Should be only one word.  In other words, commands are supported, but
  commands and arguments are not.
* Applies to commands such as: search scripts, scripted inputs, SSL
  certificate generation scripts.  (Any commands that are
  user-configurable).
* Does not apply to trusted/non-configurable command executions, such as:
  splunk search, splunk-optimize, gunzip.
* $SPLUNK_HOME is expanded.
* No default.

[app_backup]
backup_path = <string>
* Full path to the directory that contains configuration backups created by
  Splunk Enterprise.
* For search head clusters, this directory resides on the deployer.
* Default: $SPLUNK_HOME/var/backup
```

search head clustering configuration

```
[shclustering]
disabled = <boolean>
* Disables or enables search head clustering on this instance.
* When enabled, the captain needs to be selected via a
  bootstrap mechanism. Once bootstrapped, further captain
  selections are made via a dynamic election mechanism.
* When enabled, you must also specify the cluster member's own server
  address / management URI for identification purpose. This can be
  done in 2 ways: by specifying the 'mgmt_uri' setting individually on
  each member or by specfing pairs of 'GUID, mgmt-uri' strings in the
  servers_list setting.
* Default: true

mgmt_uri = [ mgmt-URI ]
* The management URI is used to identify the cluster member's own address to
  itself.
* Either 'mgmt_uri' or 'servers_list' is necessary.
* The 'mgmt_uri' setting is simpler to author but is unique for each member.
* The 'servers_list' setting is more involved, but can be copied as a
  config string to all members in the cluster.

servers_list = [ <(GUID, mgmt-uri);>+ ]
* A semicolon separated list of instance GUIDs and management URIs.
* Each member uses its GUID to identify its own management URI.

adhoc_searchhead = <boolean>
* This setting configures a member as an ad-hoc search head; i.e., the member
  does not run any scheduled jobs.
* Use the setting 'captain_is_adhoc_searchhead' to reduce compute load on the
  captain.
* Default: false

no_artifact_replications = <boolean>
* Prevent this Search Head Cluster member to be selected as a target for
  replications.
* This is an advanced setting, and not to be changed without proper
  understanding of the implications.
* Default: false
```

```
precompress_artifacts = <boolean>
* Determines whether this search head cluster member compresses the
  search artifacts before replicating them to other members.
* When set to "true", the search head compresses the artifacts
  before replicating them to all other members.
  This helps reduce network bandwidth consumption during artifact replications.
* Set this option to 'true' only when SSL compression is off on
  each search head cluster member. To turn off SSL compression, set
  'allowSslCompression = false' in the [sslconfig] stanza in server.conf
  of each member.
* Default: true

captain_is_adhoc_searchhead = <boolean>
* This setting prohibits the captain from running scheduled jobs.
* The captain is dedicated to controlling the activities of the cluster,
  but can also run adhoc search jobs from clients.
* Default: false

preferred_captain = <boolean>
* The cluster tries to assign captaincy to a member with
 'preferred_captain=true'.
* Note that it is not always possible to assign captaincy to a member with
  preferred_captain=true - for example, if none of the preferred members is
  reachable over the network. In that case, captaincy might remain on a
  member with preferred_captain=false.
* Default: true

prevent_out_of_sync_captain = <boolean>
* This setting prevents a node that could not sync config changes to current
  captain from becoming the cluster captain.
* This setting takes precedence over the preferred_captain setting. For example,
  if there are one or more preferred captain nodes but the nodes cannot
  sync config changes with the current captain, then the current captain
  retains captaincy even if it is not a preferred captain.
* This must be set to the same value on all members.
* Default: true

replication_factor = <positive integer>
* Determines how many copies of search artifacts are created in the cluster.
* This must be set to the same value on all members.
* Default: 3

pass4SymmKey = <string>
* Secret shared among the members in the search head cluster to prevent any
  arbitrary instance from connecting to the cluster.
* All members must use the same value.
* If set in the [shclustering] stanza, it takes precedence over any setting
  in the [general] stanza.
* Unencrypted passwords must not begin with "$1$", as this is used by
  Splunk software to determine if the password is already encrypted.
* Default: The 'changeme' from the [general] stanza in the default the
  server.conf file.

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' should be for a
  particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length than
  what you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
```

```
* Default: 12

async_replicate_on_proxy = <boolean>
* If the jobs/${sid} REST endpoint or its sub-resources (e.g.
  jobs/${sid}/results, jobs/${sid}/summary, etc.) had to be proxied to a
  different member due to missing local replica, this setting automatically
  schedules an async replication to that member when set to true.
* Default: true

master_dump_service_periods = <integer>
* DEPRECATED; use captain_dump_service_periods instead.

captain_dump_service_periods = <integer>
* If SHPMaster info is switched on in log.cfg, then captain statistics
  are dumped in splunkd.log after the specified number of service periods.
  Purely a debugging aid.
* Default: 500

long_running_jobs_poll_period = <integer>
* Long running delegated jobs are polled by the captain every
  "long_running_jobs_poll_period" seconds to ascertain whether they are
  still running, in order to account for potential node/member failure.
* Default: 600 (10 minutes)

scheduling_heuristic = <string>
* This setting configures the job distribution heuristic on the captain.
* There are currently two supported strategies: 'round_robin' or
  'scheduler_load_based'.
* Default: 'scheduler_load_based'

id = <string>
* Unique identifier for this cluster as a whole, shared across all cluster
  members.
* Default: Splunk software arranges for a unique value to be generated and
  shared across all members.

cxn_timeout = <integer>
* Low-level timeout, in seconds, for establishing connection between
  cluster members.
* Default: 60

send_timeout = <integer>
* Low-level timeout, in seconds, for sending data between search head
  cluster members.
* Default: 60

rcv_timeout = <integer>
* Low-level timeout, in seconds, for receiving data between search head
  cluster members.
* Default: 60

cxn_timeout_raft = <integer>
* Low-level timeout, in seconds, for establishing connection between search
  head cluster members for the raft protocol.
* Default: 2

send_timeout_raft = <integer>
* Low-level timeout, in seconds, for sending data between search head
  cluster members for the raft protocol.
* Default: 5

rcv_timeout_raft = <integer>
```

```
* Low-level timeout, in seconds, for receiving data between search head
  cluster members for the raft protocol.
* Default: 5

rep_cxn_timeout = <integer>
* Low-level timeout, in seconds, for establishing connection for replicating
  data.
* Default: 60

rep_send_timeout = <integer>
* Low-level timeout, in seconds, for sending replication slice data
  between cluster members.
* This is a soft timeout. When this timeout is triggered on source peer,
  it tries to determine if target is still alive. If it is still alive,
  it reset the timeout for another rep_send_timeout interval and continues.
  If target has failed or cumulative timeout has exceeded
  rep_max_send_timeout, replication fails.
* Default: 60

rep_rcv_timeout = <integer>
* Low-level timeout, in seconds, for receiving acknowledgement data from
  members.
* This is a soft timeout. When this timeout is triggered on source member,
  it tries to determine if target is still alive. If it is still alive,
  it reset the timeout for another rep_send_timeout interval and continues.
  If target has failed or cumulative timeout has exceeded
  the 'rep_max_rcv_timeout' setting, replication fails.
* Default: 60

rep_max_send_timeout = <integer>
* Maximum send timeout, in seconds, for sending replication slice data
  between cluster members.
* On 'rep_send_timeout' source peer determines if total send timeout has
  exceeded rep_max_send_timeout. If so, replication fails.
* If cumulative rep_send_timeout exceeds 'rep_max_send_timeout', replication
  fails.
* Default: 600 (10 minutes)

rep_max_rcv_timeout = <integer>
* Maximum cumulative receive timeout, in seconds, for receiving acknowledgement
  data from members.
* On 'rep_rcv_timeout' source member determines if total receive timeout has
  exceeded 'rep_max_rcv_timeout'. If so, replication fails.
* Default: 600 (10 minutes)

log_heartbeat_append_entries = <boolean>
* If true, Splunk software logs the the low-level heartbeats between members in
  splunkd_access.log file. These heartbeats are used to maintain the authority
  of the captain authority over other members.
* Default: false

election_timeout_ms = <positive_integer>
* The amount of time, in milliseconds, that a member waits before
  trying to become the captain.
* Note that modifying this value can alter the heartbeat period (See
  election_timeout_2_hb_ratio for further details)
* A very low value of election_timeout_ms can lead to unnecessary captain
  elections.
* Default: 60000 (1 minute)

election_timeout_2_hb_ratio = <positive_integer>
* The ratio between the election timeout, set in 'election_timeout_ms', and
```

```
  the raft heartbeat period.
* The raft heartbeat period is 'election_timeout_ms' / 'election_timeout_2_hb_ratio'.
* This ratio determines the number of heartbeat attempts that would fail
  before a member starts to timeout and tries to become the captain.
* A typical ratio between 5 - 20 is desirable.
* Default: 12 (to keep the raft heartbeat period at 5 seconds)

heartbeat_timeout = <positive integer>
* The amount of time, in seconds, that the captain considers a member down.
  After a member is down, the captain initiates fixup steps to replicate
  artifacts from the dead member to its peers.
* This heartbeat exchanges data between the captain and members, which helps in
  maintaining the in-memory centralized state for all the cluster members.
* Note that this heartbeat is different from the Raft heartbeat described
  in the 'election_timeout_2_hb_ratio' setting.
* Default: 60 (1 minute)

raft_rpc_backoff_time_ms = <positive integer>
* Provides a delay, in milliseconds, should a raft RPC request fail.
* This avoids rapid connection requests being made to unreachable peers.
* This setting should not normally be changed from the default.
* Default: 5000 (5 seconds)

access_logging_for_heartbeats = <boolean>
* Only valid on captain.
* Enables/disables logging to the splunkd_access.log file for member heartbeats
* NOTE: you do not have to restart captain to set this config parameter.
  Simply run the cli command on master:
  % splunk edit shcluster-config -access_logging_for_heartbeats <<boolean>>
* Default: false (logging disabled)

restart_timeout = <positive integer>
* This is the amount of time the captain waits for a member to come
  back when the instance is restarted (to avoid the overhead of
  trying to fixup the artifacts that were on the peer).

quiet_period = <positive integer>
* The amount of time, in seconds, for which a newly
  elected captain waits for members to join.
* During this period the captain does not initiate any fixups
  but instead waits for the members to register themselves.
  Job scheduling and conf replication still happen as usual during
  this time. At the end of this time period, the captain builds its
  view of the cluster based on the registered peers and starts normal
  processing.
* Default: 60

max_peer_rep_load = <integer>
* This is the maximum number of concurrent replications that a
  member can take part in as a target.
* Default: 5

target_wait_time = <positive integer>
* The amount of time, in seconds, that the captain waits for the target
  of a replication to register itself before it services the artifact again
  and potentially schedules another fixup.
* Default: 150

manual_detention = on|off
* This property toggles manual detention on member.
* When a node is in manual detention, it does not accept new search jobs,
  including both scheduled and ad-hoc searches. It also does not receive
```

```
  replicated search artifacts from other nodes.
* Default: off

percent_peers_to_restart = <integer>
* The percentage of members to restart at one time during rolling restarts.
* Actual percentage may vary due to lack of granularity for smaller peer
  sets regardless of setting, a minimum of 1 peer is restarted per
  round.
* Valid values are between 0 and 100.
* CAUTION: Do not set this setting to a value greater than 20%.
  Otherwise, issues can arise during the captain election process.

rolling_restart_with_captaincy_exchange = <boolean>
* Whether or not the captain tries to exchange captaincy with another node
  during a rolling restart.
* A value of "true" means the captain tries to exchange captaincy
  with another node during a rolling restart.
* A value of "false" means the captain restarts and captaincy transfers to some
  other node.
* Default: true

rolling_restart = restart|searchable|searchable_force
* Determines the rolling restart mode for a search head cluster.
* If set to "restart", a rolling restart runs in classic mode.
* If set to "searchable", a rolling restart runs in searchable (minimal
  search disruption) mode.
* If set to "searchable_force", the search head cluster performs a
  searchable rolling restart, but overrides the health check.
* You do not have to restart any search head members to configure this setting.
  Run this CLI command from any member:
  % splunk edit shcluster-config -rolling_restart
    restart|searchable|searchable_force
* Default: restart (runs in classic rolling-restart mode)

decommission_search_jobs_wait_secs = <unsigned integer>
* The amount of time, in seconds, that a search head cluster member waits for
  existing searches to complete before restarting.
* Applies only when rolling restart is triggered in searchable or
  searchable_force mode
  (i.e.'rolling_restart' is set to "searchable" or "searchable_force").
* You do not have to restart search head members to configure this setting.
  Run this CLI command from any member:
  % splunk edit shcluster-config -decommission_search_jobs_wait_secs
    <positive integer>
* NOTE: If you specify 'decommission_search_jobs_wait_secs' in the '[general]'
  stanza, leave it unchanged at its default value.
* Default: 180

register_replication_address = <string>
* This setting is the address on which a member is available for
  accepting replication data. This is useful in the cases where a member
  host machine has multiple interfaces and only one of them can be reached
  by another splunkd instance.
* Can be an IP address, or a fully qualified machine/domain name.

executor_workers = <positive integer>
* Number of threads that can be used by the search head clustering
  threadpool.
* A value of 0 is interpreted as 1.
* Default: 50

heartbeat_period = <non-zero positive integer>
```

```
* The frequency, in seconds, with which the member attempts
  to send heartbeats to the captain.
* This heartbeat exchanges data between the captain and members, which
  helps in maintaining the in-memory centralized state for all the
  cluster members.
* NOTE: This heartbeat period is different from the Raft heartbeat
  period in the 'election_timeout_2_hb_ratio' setting.
* Default: 5

enableS2SHeartbeat = <boolean>
* Whether or not Splunk software monitors each replication connection for
  presence of a heartbeat.
* A value of "true" means that Splunk software monitors the presence of a
  heartbeat. If the heartbeat is not seen for 's2sHeartbeatTimeout' seconds,
  the instance that monitors the heartbeat closes the connection.
* Default: true

s2sHeartbeatTimeout = <integer>
* The global timeout, in seconds, for monitoring heartbeats on replication
  connections.
* Splunk software closes a replication connection if a heartbeat is not seen
  for 's2sHeartbeatTimeout' seconds.
* Replication source sends a heartbeat every 30 seconds.
* Default: 600 (10 minutes)

captain_uri = [ static-captain-URI ]
* The management URI of static captain is used to identify the cluster
  captain for a static captain.


election = <boolean>
* This is used to classify a cluster as static or dynamic (RAFT based).
* If set to "false", a static captain, which is used for DR situation.
* If set to "true", a dynamic captain election enabled through RAFT protocol.

mode = <member>
* Accepted values are captain and member, mode is used to identify
  the function of a node in static search head cluster.
* Setting mode as captain assumes it to function as both captain and a member.

# proxying related
sid_proxying = <boolean>
* Enable or disable search artifact proxying.
* Changing this affects the proxying of search results, and jobs feed
  is not cluster-aware.
* Only for internal/expert use.
* Default: true

ss_proxying = <boolean>
* Enable or disable saved search proxying to captain.
* Changing this affects the behavior of Searches and Reports page
  in Splunk Web.
* Only for internal/expert use.
* Default: true

ra_proxying = <boolean>
* Enable or disable saved report acceleration summaries proxying to captain.
* Changing this affects the behavior of report acceleration summaries
  page.
* Only for internal/expert use.
* Default: true
```

```
alert_proxying = <boolean>
* Enable or disable alerts proxying to captain.
* Changing this impacts the behavior of alerts, and essentially make them
  not cluster-aware.
* Only for internal/expert use.
* Default: true

csv_journal_rows_per_hb = <integer>
* How many rows of CSV from the delta-journal are sent per hb
* Used for both alerts and suppressions
* Do not alter this value without contacting Splunk Support.
* Default: 10000

conf_replication_period = <integer>
* How often, in seconds, a cluster member replicates
  configuration changes.
* A value of 0 disables automatic replication of configuration changes.
* Default: 5

conf_replication_max_pull_count = <integer>
* The maximum number of configuration changes a member
  replicates from the captain at one time.
* A value of 0 disables any size limits.
* Default: 1000

conf_replication_max_push_count = <integer>
* The maximum number of configuration changes a member
  replicates to the captain at one time.
* A value of 0 disables any size limits.
* Default: 100

conf_replication_max_json_value_size = [<integer>|<integer>[KB|MB|GB]]
* The maximum size of a JSON string element at any nested
  level while parsing a configuration change from JSON representation.
* If a knowledge object created on a member has some string element
  that exceeds this limit, the knowledge object is not replicated
  to the rest of the search head cluster, and a warning that mentions
  conf_replication_max_json_value_size is written to splunkd.log.
* If you do not specify a unit for the value, the unit defaults to bytes.
* The lower limit of this setting is 512KB.
* When increasing this setting beyond the default, you must take into
  account the available system memory.
* Default: 15MB

conf_replication_include.<conf_file_name> = <boolean>
* Whether Splunk replicates changes to a particular type of *.conf
  file, along with any associated permissions in *.meta files.
* Default: false

conf_replication_summary.whitelist.<name> = <whitelist_pattern>
* DEPRECATED; use conf_replication_summary.includelist.<name> instead.

conf_replication_summary.includelist.<name> = <includelist_pattern>
* Files to be included in configuration replication summaries.

conf_replication_summary.blacklist.<name> = <blacklist_pattern>
* DEPRECATED; use conf_replication_summary.excludelist.<name> instead.

conf_replication_summary.excludelist.<name> = <excludelist_pattern>
* Files to be excluded from configuration replication summaries.

conf_replication_summary.concerning_file_size = <integer>
```

```
* Any individual file within a configuration replication summary that is
  larger than this value (in MB) triggers a splunkd.log warning message.
* Default: 50

conf_replication_summary.period = <timespan>
* How often configuration replication summaries are created.
* Default: 1m (1 minute)

conf_replication_purge.eligibile_count = <integer>
* How many configuration changes must be present before any become
  eligible for purging.
* In other words: controls the minimum number of configuration changes
  Splunk software remembers for replication purposes.
* Default: 20000

conf_replication_purge.eligibile_age = <timespan>
* How old a configuration change must be before it is eligible for
  purging.
* Default: 1d (1 day).

conf_replication_purge.period = <timespan>
* How often configuration changes are purged.
* Default: 1h (1 hour)

conf_replication_find_baseline.use_bloomfilter_only = <boolean>
* Whether or not a search head cluster only uses bloom filters to
  determine a baseline, when it replicates configurations.
* Set to "true" to only use bloom filters in baseline determination during
  configuration replication.
* Set to "false" to first attempt a standard method, where the search head
  cluster captain interacts with members to determine the baseline, before
  falling back to using bloom filters.
* Default: false

conf_deploy_repository = <path>
* Full path to directory containing configurations to deploy to cluster
  members.

conf_deploy_staging = <path>
* Full path to directory where preprocessed configurations may be written
  before being deployed cluster members.

conf_deploy_concerning_file_size = <integer>
* Any individual file within <conf_deploy_repository> that is larger than
  this value (in MB) triggers a splunkd.log warning message.
* Default: 50

conf_deploy_precompress_bundles = <boolean>
* Determines whether or not the deployer compresses the configuration bundle
  files before pushing them to search heads, which reduces network
  bandwidth consumption.
* Set this option to "true" only when SSL compression is off. Otherwise, the
  files will be compressed twice, which wastes CPU resources and does not save
  network bandwidth. To turn off SSL compression, set
  "allowSslCompression = false" in server.conf on the deployer.
* Default: true

conf_deploy_fetch_url = <URL>
* Specifies the location of the deployer from which members fetch the
  configuration bundle.
* This value must be set to a <URL> in order for the configuration bundle to
  be fetched.
```

```
* No default.

conf_deploy_fetch_mode = auto|replace|none
* Controls configuration bundle fetching behavior when the member starts up.
* When set to "replace", a member checks for a new configuration bundle on
  every startup.
* When set to "none", a member does not fetch the configuration bundle on
  startup.
* Regarding "auto":
  * If no configuration bundle has yet been fetched, "auto" is equivalent
    to "replace".
  * If the configuration bundle has already been fetched, "auto" is
    equivalent to "none".
* Default: replace

artifact_status_fields = <comma-separated list>
* Give a comma separated fields to pick up values from status.csv and
  info.csv for each search artifact.
* These fields are shown in the CLI/REST endpoint splunk list
  shcluster-member-artifacts
* Default: user, eai:acl.app , label

encrypt_fields = <comma-separated list>
* DEPRECATED.
* Use the setting in the '[general]' stanza instead.

enable_jobs_data_lite = <boolean>
* DEPRECATED.
* Use the 'jobs_data_lite.enabled' instead.

jobs_data_lite.enabled = <boolean>
* Enable memory optimizations for sharing search job status within search head
  clustering.
* Default: true

jobs_data_lite.exclude_fields = <comma separated list>
* List of job status fields to be excluded from truncation when
  jobs_data_lite.enabled is true.
* Fields to exclude must be in a comma separated list.
* No default

jobs_data_lite.search_field_len = <non-negative integer>
* Maximum length for any search-based field in the search job status when
  jobs_data_lite.enabled is true.  Fields longer than this value will be
  truncated.
* Any field larger than this size will be truncated unless configured in the
  jobs_data_lite.exclude_fields list.
* Search fields include: remote_search, normalized_search, optimized_search,
  phase_0_search, phase_1_search, report_search, and events_search.
* Default: 100

jobs_data_lite.default_field_len = <non-negative integer>
* Maximum length for any nonsearch-based field in the search job status
  when jobs_data_lite.enabled is true.  Fields longer than this value will be
  truncated.
* Any field larger than this size will be truncated unless configured in the
  jobs_data_lite.exclude_fields list.
* Default: 1000000

jobs_data_lite.max_status_size_per_hb = <non-negative integer>
* The maximum size, in megabyte, of status.csv
* status.csv tracks job statuses and is sent between the captain and each cluster
```

```
  member in each heartbeat.
  Limiting the size of status.csv helps to stabilize the communication between the
  captain and members by preventing the heartbeat from growing overly large.
* Default: 700
* Recommended range: 500-1000
* Absolute range: 100-1500

shcluster_label = <string>
* This specifies the label of the search head cluster.

retry_autosummarize_or_data_model_acceleration_jobs = <boolean>
* Whether or not the captain tries a second time to delegate an
  auto-summarized or data model acceleration job, if the first attempt to
  delegate the job fails.
* Default: true

deployerPushThreads = <positive integer>|auto
* The maximum number of threads to use when performing a deployer bundle push
  to target members.
* If set to "auto", the deployer auto-tunes the number of threads it uses
  for a deployer bundle push. There will be one thread per target member.
* Default: 1

remote_job_retry_attempts = <positive integer>
* Defines the maximum number of re-run attempts for a failing search job
  (this number includes the initial attempt).
* Note that this setting only applies to jobs that either failed to be
  delegated or jobs that returned failure. This means that jobs which have
  'allow_partial_results' set to true will not be re-run.
* The upper limit of the number of job re-run attempts is constrained by
  the total number of nodes in the Search Head Cluster.
* Default: 2


allow_concurrent_dispatch_savedsearch = <boolean>
* The search head cluster captain might dispatch multiple saved searches to a member
  through REST calls.
* This option controls whether the member processes the dispatched REST calls
  concurrently or sequentially.
* If true, the member processes the REST calls concurrently.
* If false, the member processes the REST calls sequentially.
* Default: true

[replication_port://<port>]
############################################################################
# Configures the member to listen on a given TCP port for replicated data
# from another cluster member.
# At least one replication_port must be configured and not disabled.
############################################################################

disabled = <boolean>
* Set to true to disable this replication port stanza.
* Default: false

listenOnIPv6 = no|yes|only
* Toggle whether this listening port listens on IPv4, IPv6, or both.
* If not present, the setting in the [general] stanza is used.

acceptFrom = <network_acl> ...
* Lists a set of networks or addresses from which to accept connections.
* Separate multiple rules with commas or spaces.
* Each rule can be in one of the following formats:
```

```
    1. A single IPv4 or IPv6 address (examples: "10.1.2.3", "fe80::4a3")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
       (examples: "10/8", "192.168.1/24", "fe80:1234/32")
    3. A DNS name, possibly with a "*" used as a wildcard
       (examples: "myhost.example.com", "*.splunk.com")
    4. "*", which matches anything
* You can also prefix an entry with '!' to cause the rule to reject the
  connection. The input applies rules in order, and uses the first one that
  matches.
  For example, "!10.1/16, *" allows connections from everywhere except
  the 10.1.*.* network.
* Default: "*" (accept from anywhere)

[replication_port-ssl://<port>]
* This configuration is the same as the replication_port stanza, but uses SSL.

disabled = <boolean>
* Set to true to disable this replication port stanza.
* Default: false

listenOnIPv6 = no|yes|only
* Toggle whether this listening port listens on IPv4, IPv6, or both.
* Default: The setting in the [general] stanza

acceptFrom = <network_acl> ...
* This setting is the same as the setting in the [replication_port] stanza.

serverCert = <path>
* Full path to file containing private key and server certificate.
* The <path> must refer to a PEM format file.
* No default.

sslPassword = <string>
* Server certificate password, if any.
* No default.

password = <string>
* DEPRECATED; use 'sslPassword' instead.
* Used only if 'sslPassword' is not set.

rootCA = <string>
* DEPRECATED; use '[sslConfig]/sslRootCAPath' instead.
* Used only if '[sslConfig]/sslRootCAPath' is not set.
* Full path to the root CA (Certificate Authority) certificate store.
* The <path> must refer to a PEM format file containing one or more root CA
  certificates concatenated together.
* No default.

cipherSuite = <string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* provided by OpenSSL.  This is used to ensure that the server does not
  accept connections using weak encryption protocols.

supportSSLV3Only = <boolean>
* DEPRECATED.  SSLv2 is now always disabled.  The exact set of SSL versions
  allowed is now configurable via the "sslVersions" setting above.

useSSLCompression = <boolean>
* If true, enables SSL compression.
* Default: false
```

```
compressed = <boolean>
* DEPRECATED; use 'useSSLCompression' instead.
* Used only if 'useSSLCompression' is not set.

requireClientCert = <boolean>
* Requires that any peer that connects to replication port has a certificate
  that can be validated by certificate authority specified in rootCA.
* Default: false

allowSslRenegotiation = <boolean>
* In the SSL protocol, a client may request renegotiation of the connection
  settings from time to time.
* Setting this to false causes the server to reject all renegotiation
  attempts, breaking the connection.  This limits the amount of CPU a
  single TCP connection can use, but it can cause connectivity problems
  especially for long-lived connections.
* Default: true
```

App Key Value Store (KV Store) configuration

```
[kvstore]

disabled = <boolean>
* Set to true to disable the KV Store process on the current server. To
  completely disable KV Store in a deployment with search head clustering or
  search head pooling, you must also disable KV Store on each individual
  server.
* Default: false

port = <integer>
* Port to connect to the KV Store server.
* Default: 8191

replicaset = <string>
* Replica set name.
* Default: splunkrs

distributedLookupTimeout = <integer>
* This setting has been removed, as it is no longer needed.

shutdownTimeout = <integer>
* Time, in seconds, to wait for a clean shutdown of the KV Store. If this time
  is reached after signaling for a shutdown, KV Store is forcibly terminated
* Default: 100

initAttempts = <integer>
* The maximum number of attempts to initialize the KV Store when starting
  splunkd.
* Default: 300

replication_host = <string>
* The host name to access the KV Store.
* This setting has no effect on a single Splunk platform instance.
* When using search head clustering, if the "replication_host" value is not
  set in the [kvstore] stanza, the host you specify for
  "mgmt_uri" in the [shclustering] stanza is used for KV
  Store connection strings and replication.
* In search head pooling, this host value is a requirement for using KV
  Store.
* This is the address on which a kvstore is available for accepting
```

```
  remotely.

verbose = <boolean>
* Whether or not verbose logging for KV Store is enabled.
* Set to "true" to enable verbose logging.
* Default: false

verboseLevel = <nonnegative integer>
* When verbose logging is enabled, specifies the level of verbosity for logging
  from 0 to 5, where 5 is the most verbose.
* Default: 2

dbPath = <string>
* Path where KV Store data is stored.
* Changing this directory after initial startup does not move existing data.
  The contents of the directory should be manually moved to the new
  location.
* Default: $SPLUNK_DB/kvstore

storageEngine = wiredTiger
* The storage engine that KV Store uses to manage its data.
* "mmapv1" is no longer supported for KV Store.
* When you upgrade the Splunk platform, the KV Store storage engine will be
  migrated to "wiredTiger" automatically if "mmapv1" is still being used and
  the Splunk platform instance is not a member of a search head cluster.
* Default: wiredTiger

storageEngineMigration = <boolean>
* DEPRECATED.

oplogSize = <integer>
* The size of the replication operation log, in megabytes, for environments
  with search head clustering or search head pooling.
  In a standalone environment, 20% of this size is used.
* After the KV Store has created the oplog for the first time, changing this
  setting does NOT affect the size of the oplog. A full backup and restart
  of the KV Store is required.
* Do not change this setting without first consulting with Splunk Support.
* Default: 1000 (1GB)

replicationWriteTimeout = <integer>
* The time to wait, in seconds, for replication to complete while saving KV
  store operations. When the value is 0, the process never times out.
* Used for replication environments (search head clustering or search
  head pooling).
* Default: 1800 (30 minutes)

clientConnectionTimeout = <positive integer>
* The time, in seconds, to wait while attempting a connection to the KV Store
  before the attempt times out.
* Default: 10

clientSocketTimeout = <positive integer>
* The time, in seconds, to wait while attempting to send or receive on a
  socket before the attempt times out.
* Default: 300 (5 minutes)

clientConnectionPoolSize = <positive integer>
* The maximum number of active client connections to the KV Store.
* When the number of active connections exceeds this value, KV Store will
  reject new connection attempts until at least one active connection closes.
* Do not change this setting without first consulting with Splunk Support.
```

```
* Default: 500

caCertFile = <string>
* DEPRECATED; use '[sslConfig]/sslRootCAPath' instead.
* Used only if 'sslRootCAPath' is not set.
* Full path to a CA (Certificate Authority) certificate(s) PEM format file.
* If specified, it is used in KV Store SSL connections and
  authentication.
* Only used when Common Criteria is enabled (SPLUNK_COMMON_CRITERIA=1)
  or FIPS is enabled (i.e. SPLUNK_FIPS=1).
* NOTE: Splunk plans to submit Splunk Enterprise for Common Criteria
  evaluation. Splunk does not support using the product in Common
  Criteria mode until it has been certified by NIAP. See the "Securing
  Splunk Enterprise" manual for information on the status of Common
  Criteria certification.
* Default: $SPLUNK_HOME/etc/auth/cacert.pem

caCertPath = <string>
* DEPRECATED; use '[sslConfig]/sslRootCAPath' instead.

serverCert = <string>
* A certificate file signed by the signing authority specified above by
  caCertPath.
* In search head clustering or search head pooling, the certificates at
  different members must share the same 'subject'.
* The Distinguished Name (DN) found in the certificate's subject, must
  specify a non-empty value for at least one of the following settings:
  Organization (O), the Organizational Unit (OU) or the
  Domain Component (DC).
* Only used when Common Criteria is enabled (SPLUNK_COMMON_CRITERIA=1)
  or FIPS is enabled (i.e. SPLUNK_FIPS=1).
* NOTE: Splunk plans to submit Splunk Enterprise for Common Criteria
  evaluation. Splunk does not support using the product in Common
  Criteria mode until it has been certified by NIAP. See the "Securing
  Splunk Enterprise" manual for information on the status of Common
  Criteria certification.

sslVerifyServerCert = <boolean>
* A value of "true" means make sure that the connected server is
  authenticated. Both the common name and the alternate name
  of the server are checked for a match if they are specified
  in this configuration file. A certificate is considered
  verified if either is matched.
* If you have enabled FIPS (by setting SPLUNK_FIPS=1), splunkd always verifies
  the server certificate, and ignores this setting.
* Default (if you have not enabled FIPS): false

sslVerifyServerName = <boolean>
* See the description of 'sslVerifyServerName' under the [sslConfig] stanza
  for details on this setting.
* Default: false

sslKeysPath = <string>
* DEPRECATED; use 'serverCert' instead.
* Used only when 'serverCert' is empty.

sslPassword = <string>
* Password of the private key in the file specified by 'serverCert' above.
* Must be specified if FIPS is enabled (i.e. SPLUNK_FIPS=1), otherwise, KV
  Store is not available.
* Only used when Common Criteria is enabled (SPLUNK_COMMON_CRITERIA=1)
  or FIPS is enabled (i.e. SPLUNK_FIPS=1).
```

```
* NOTE: Splunk plans to submit Splunk Enterprise for Common Criteria
  evaluation. Splunk does not support using the product in Common
  Criteria mode until it has been certified by NIAP. See the "Securing
  Splunk Enterprise" manual for information on the status of Common
  Criteria certification.
* No default.

sslKeysPassword = <string>
* DEPRECATED; use 'sslPassword' instead.
* Used only when 'sslPassword' is empty.

sslCRLPath = <string>
* The path to the Certificate Revocation List (CRL) file.
* A CRL is a list of digital certificates that have been revoked by
  the issuing Certificate Authority (CA) before their scheduled expiratio date
  and can no longer be trusted.
* Splunkd uses the CRL file only in the following cases:
  * When the Splunk platform instance is in Common Criteria mode
    (SPLUNK_COMMON_CRITERIA=1).
  * When the instance is in FIPS mode (SPLUNK_FIPS=1).
  * When you have enabled certificate status validation checks by configuring the
    '[sslConfig]:certificateStatusValidationMethod' setting. See this setting
    to learn how to configure certificate status validation.
* The file that this setting value references must be in privacy-enhanced mail (PEM)
  format.
* NOTE: Splunk does not support using the product in Common Criteria mode until
  it has been certified by the National Information Assurance Partnership (NIAP).
  See the "Securing Splunk Enterprise" and "Securing Splunk Enterprise with
  Common Criteria" manuals for information on the status of Common
  Criteria certification.
* Optional.
* Default: empty string (no revocation list)

modificationsReadIntervalMillisec = <integer>
* How often, in milliseconds, to check for modifications to
  KV Store collections in order to replicate changes for distributed
  searches.
* Default: 1000 (1 second)

modificationsMaxReadSec = <integer>
* Maximum time interval KVStore can spend while checking for modifications
  before it produces collection dumps for distributed searches.
* Default: 30

initialSyncMaxFetcherRestarts = <positive integer>
* Specifies the maximum number of query restarts an oplog fetcher can perform
  before failing the ongoing Initial Sync attempt.
* Increasing this value might help in dynamic deployments with very large
  KV Store databases where Initial Sync might take a long time.
* NOTE: This setting should be changed only if you have been asked to set it by
  a Splunk Support engineer. It might increase KV Store cluster failover time.
* Default: 0



delayShutdownOnBackupRestoreInProgress = <boolean>
* Whether or not splunkd should delay a shutdown if a KV Store backup or restore
  operation is in progress.
* If set to "true", splunkd waits until either the running backup/restore operation
  completes, or 'splunkd_stop_timeout' seconds have elapsed since it received
  the shutdown request.
* NOTE: Setting this to "true" might delay splunkd shutdown for several minutes,
```

```
  depending on the amount of data that KV Store uses and the value of
  'splunkd_stop_timeout'.
* Default: false

percRAMForCache = <positive integer>
* The percentage of total system memory that KV store can use.
* Value can range from 5 to 50, inclusive.
* If less than 1 GB of system memory is present, only 256 MB of cache will be used.
* If you have less than 256 MB of system memory, you cannot use KVStore with wiredTiger.
* Changing this value can affect performance on KV store,
  Splunk Enterprise apps that use KV store, and KV store lookups.
  For more information, search the Splunk documentation for "KV store
  troubleshooting tools".
* If you are not using the WiredTiger storage engine, Splunk Enterprise ignores
  this setting.
* Default: 15
```

Indexer Discovery configuration

```
[indexer_discovery]
pass4SymmKey = <string>
* Security key shared between manager node and forwarders.
* If specified here, the same value must also be specified on all forwarders
  connecting to this manager.
* Unencrypted passwords must not begin with "$1$", as this is used by
  Splunk software to determine if the password is already encrypted.

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' should be for a
  particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length than
  what you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* Default: 12

polling_rate = <integer>
* A value between 1 to 10. This value affects the forwarder polling
  frequency to achieve the desired polling rate. The number of connected
  forwarders is also taken into consideration.
* The formula used to determine effective polling interval,
  in Milliseconds, is:
  (number_of_forwarders/polling_rate + 30 seconds) * 1000
* Default: 10

indexerWeightByDiskCapacity = <boolean>
* A value of "true" means it instructs the forwarders to use weighted load
  balancing. In weighted load balancing, load balancing is based on the
  total disk capacity  of the target indexers, with the forwarder streaming
  more data to indexers with larger disks.
* The traffic sent to each indexer is based on the ratio of:
  indexer_disk_capacity/total_disk_capacity_of_indexers_combined
* Default: false
```

Cascading Replication Configuration

```
[cascading_replication]
pass4SymmKey = <string>
* Security key shared between indexers participating in cascading replication.
* The same value must be specified on all indexers participating in cascading
  replication.
* Unencrypted passwords must not begin with "$1$", as this is used by
  Splunk software to determine if the password is already encrypted.
* Empty passwords will not be accepted.
* Default: None

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' should be for a
  particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length than
  what you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* Default: 12

max_replication_threads = <integer>
* Maximum threads used for replicating metadata and payload to search peers.
* If set to "auto", the peer auto-tunes the number of threads it uses for
  cascading replication.
    * If the peer has 3 or fewer CPUs, it allocates 2 threads.
    * If the peer has 4-7 CPUs, it allocates up to '# of CPUs - 2' threads.
    * If the peer has 8-15 CPUs, it allocates up to '# of CPUs - 3' threads.
    * If the peer has 16 or more CPUs, it allocates up to
      '# of CPUs - 4' threads.
* Maximum accepted value for this setting is 16.
* Default: auto

max_replication_jobs = <integer>
* Maximum jobs used for replicating metadata and payload to search peers.
* Default: 5

cascade_replication_plan_reap_interval = <interval>
* The interval at which the cascade replication plans are reaped.
* The interval can be specified as a string for minutes, seconds, hours, days.
  For example: 60s, 1m, 1h, 1d etc.
* Maximum accepted value is 5h
* Default: 1h

cascade_replication_plan_age = <interval>
* The age of the cascade replication plan when it gets reaped.
* The interval can be specified as a string for minutes, seconds, hours, or days.
  For example: 60s, 1m, 1h, 1d etc.
* Maximum accepted value is 24h
* Default: 8h

cascade_replication_plan_fanout = auto|<positive integer>
* Number of receivers that each sender replicates to at a time.
* If set to auto, Splunk automatically calculates an optimal fanout, based on
  the maximum number of replication threads, as determined by the
  'max_replication_threads' setting under [cascading_replication] in server.conf.
* If set to an integer, the integer must be no greater than the number of cluster
  peers, or, in the case of multisite clustering, no greater than the least number
  of peers on any one site.
```

```
* Default: auto

cascade_replication_plan_topology = size_balanced
* Topology used for building a cascading plan.
* When set to size_balanced, receivers are evenly distributed among senders.
  Senders on the same layer have same or similar number of receivers.
* Default: size_balanced

cascade_replication_plan_select_policy = random
* Policy for deciding which receivers the senders pick.
* When set to random, receivers are randomly picked.
* Default: random
```

Node level authentication

```
[node_auth]
signatureVersion = <comma-separated list>
* A list of authentication protocol versions that nodes of a Splunk
  deployment use to authenticate to other nodes.
* Each version of node authentication protocol implements an algorithm
  that specifies cryptographic parameters to generate authentication data.
* Nodes may only communicate using the same authentication protocol version.
* For example, if you set "signatureVersion = v1,v2" on one node, that
  node sends and accepts authentication data using versions "v1" and "v2"
  of the protocol, and you must also set "signatureVersion" to one of
  "v1", "v2", or "v1,v2" on other nodes for those nodes to mutually
  authenticate.
* For higher levels of security, set 'signatureVersion' to "v2".
* Default: v1,v2
```

Cache Manager Configuration

```
[cachemanager]
max_concurrent_downloads = <unsigned integer>
* The maximum number of buckets that can be downloaded simultaneously from
  external storage
* Default: 8

max_concurrent_uploads = <unsigned integer>
* The maximum number of buckets that can be uploaded simultaneously to external
  storage.
* Default: 8


eviction_policy = <string>
* The name of the eviction policy to use.
* Current options: lru, clock, random, lrlt, noevict, lruk
* Do not change the value from the default unless instructed by
  Splunk Support.
* Default: lru

enable_eviction_priorities = <boolean>
* When requesting buckets, search peers can give hints to the Cache Manager
  about the relative importance of buckets.
* When enabled, the Cache Manager takes the hints into consideration; when
  disabled, hints are ignored.
* Default: true
```

```
eviction_padding = <positive integer>
* Specifies the additional space, in megabytes, beyond 'minFreeSpace' that the
  cache manager uses as the threshold to start evicting data.
* If free space on a partition falls below
  ('minFreeSpace' + 'eviction_padding'), then the cache manager tries to evict
  data from remote storage enabled indexes.
* Default: 5120 (~5GB)

max_cache_size = <positive integer>
* Specifies the maximum space, in megabytes, per partition, that the cache can
  occupy on disk. If this value is exceeded, the cache manager starts
  evicting buckets.
* A value of 0 means this setting is not used to control cache eviction.
  Eviction will instead be based on the sum of 'minFreeSpace' and 'eviction_padding'
  settings, which limits the size of the partition that the cache resides on.
* Default: 0

persist_pending_upload_from_external = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Specifies whether the information of the buckets that have been uploaded
  to remote storage can be serialized to disk or not.
* When set to true, this information is serialized to disk and
  the bucket is deemed to be on remote storage.
* Otherwise, the bucket is deemed to be not on remote storage and
  bucket is then uploaded to remote storage.
* Default: true

persistent_id_set_remove_min_sync_secs = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Cache manager persists the set of objects that are
  no longer pending upload to the remote storage based
  on when the previous set of changes were persisted to disk.
* This setting controls the interval from the last persist time that
  cache manager waits to persist the current set of changes to disk.
* Default: 5

local_delete_summary_metadata_ttl = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* The local copy of a bucket needs to be synced with the copy in remote
  storage only when the bucket switches primaries.
* However in certain experimental modes of operation the delete journals
  in the remote storage could be mutated without an update to the local copy.
* Similarly, accelerated summaries in remote storage could be updated without
  an update to the local copy.
* This setting is meant for use in such modes. The Cache manager will make
  a best effort to invalidate the local delete journals and summary
  metadata files periodically.
* The period will be controlled by this ttl. A value of 0 will disable
  this behavior
* Default: 0

hotlist_recency_secs = <unsigned integer>
* When a bucket is older than this value, it becomes eligible for eviction.
  Buckets younger than this value are evicted only if there are no older
  buckets eligible for eviction.
* For the purpose of determining recency, the age of a bucket is calculated by
  subtracting the time of the bucket's most recent event data from the current time.
* For example, if the current time (expressed in UTC epoch time) is 1567891234 and
```

```
  the bucket is named db_1567809123_1557891234_10_8A21BEE9-60D4-436B-AA6D-21B68F631A8B,
  thus indicating that the time of the most recent event in the bucket is 1567809123,
  then the bucket's age, in seconds, is 82111 (~23 hours).
* Ensure that the cache is of sufficient size to handle the value of this setting.
  Otherwise, cache eviction cannot function optimally.  In other words, do not
  configure this setting to a size that will cause the cache to retain a quantity of
  buckets that approach or exceed the size of the cache based on this setting
  alone.
* Also, consider the amount of data you're ingesting and the needs of
  the types of searches you run. As a best practice, start with a fairly low value
  for this setting and adjust over time.
* For example, if the cache size is 100 GB and you typically add 10 GB of new buckets to
  the indexer in a 24 hour period, setting this to 172800 (48 hours) would mean that
  the cache manager will try to keep those 20 GB of recent buckets in the cache
  all the time.
* This setting can be overridden on a per-index basis in indexes.conf.
* Default: 86400 (24 hours)

hotlist_bloom_filter_recency_hours = <unsigned integer>
* When a bucket's non-journal and non-tsidx files (such as bloomfilter files)
  are older than this value, those files become eligible for eviction. Bloomfilter
  and associated files younger than this value are evicted only if there are
  no older files eligible for eviction.
* The recency of a bloomfilter file is based on its bucket's recency and is
  calculated in the same manner described for 'hotlist_recency_secs'.
* This setting works in concert with 'hotlist_recency_secs' which is designed to be
  configured for a shorter age. If 'hotlist_recency_secs' leads to the eviction of a
  bucket, the bloomfilter and associated files will continue to remain in the cache
  until they reach the age configured by this setting. Thus, the bucket will remain
  in cache, but without its journal and tsidx files.
* This setting can be overridden on a per-index basis in indexes.conf.
* Default: 360 (15 days)

evict_on_stable = <boolean>
* When the source peer completes upload of a bucket to remote storage, it
  notifies the target peers so that they can evict any local copies of the bucket.
* When set to true, each target peer evicts its local copy, if any, upon such
  notification.
* When set to false, each target peer continues to store its local copy, if any,
  until its cache manager eventually evicts the bucket according to its cache
  eviction policy.
* Default: false

max_file_exists_retry_count = <unsigned integer>
* The cache manager retries its check on whether the file exists on
  remote storage when the check fails due to network errors until
  the retry count exceeds this setting.
* Default: 5

access_logging = <boolean>
* Enables/disables logging to the splunkd_access.log file for cachemanager requests.
* Default: false

cache_usage_collection_interval_minutes = <positive integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Interval at which cache usage information is reported to metrics.log.
* The cache usage logging reports cache usage, in bytes, broken down by
  cache type (bid, dma, ra, metrics), index and by time range. The time
  bins are defined by the setting 'cache_usage_collection_time_bins'.
* A value of 0 will disable this feature.
* Do not use a value less than 10 (minutes). Doing so can
```

```
  affect performance.
* Hot buckets are not managed by the cache manager and not reflected
  in the log messages.
* Default: 10

cache_usage_collection_time_bins = <comma-separated list>
* Currently not supported. This setting is related to a feature that is
  still under development.
* This setting is used when 'cache_usage_collection_interval_minutes' is
  non-zero. See the 'cache_usage_collection_interval_minutes' section for
  more information.
* This comma-separated list of integers, representing days, are boundaries
  to the time ranges to which the cache usage is broken down and reported.
  There is an implicit bin, 0, that represents all data more recent than the
  first non-zero value. The highest value specified will represent all data
  older than that value.
* For example, using the default "1, 3, 7, 14, 30, 60, 90", cache usage will
  collect the size of buckets whose latest-time (endEpoch) into the following
  bins: 0 (future-1d), 1 (1d-3d), 3 (3d-7d), 7 (7d-15d), 15 (15d-30d),
  30 (30d-60d), 60 (60d-90d), 90 (90d and older).
* Default: 1, 3, 7, 15, 30, 60, 90

cache_usage_collection_per_index = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Enables the reporting cache usage information by index.
* This setting is used when 'cache_usage_collection_interval_minutes' is
  non-zero. See the 'cache_usage_collection_interval_minutes' section for
  more information.
* Default: false

batch_registration = <boolean>
* This setting enables/disables batch registration of buckets upon indexer startup.
* If this setting is disabled, then when an indexer starts up, its cache manager
  registers each index bucket individually. This can slow the startup process.
  If an indexer is experiencing long startup durations, enable this setting to
  register buckets in batches.
* The size of each batch of buckets is set with 'batch_registration_size'.
* Default: true

batch_registration_size = <unsigned integer>
* This setting specifies the size of each batch of buckets that are
  registered.
* This setting is used when 'batch_registration' is enabled.
* Use the default value unless instructed otherwise by Splunk Support.
* Default: 5000

cache_upload_backoff_sleep_secs = <unsigned_integer>
* This setting specifies the interval, in seconds, that the cache manager waits to
  retry an upload to the remote store after encountering a 4xx HTTP error.
* A value of 0 causes the cache manager to continue retrying the upload without
  performing a backoff.
* Default: 60

max_known_remote_absent_summaries = <unsigned_integer>
* This setting specifies the maximum number of frozen (absent) summaries that the
  cache manager maintains in a list.
* The list of frozen summaries helps the cache manager to avoid making calls to the
  remote store that could result in an HTTP 404 "not found" error. By increasing the
  limit, you decrease the likelihood of such calls, while potentially using more
  memory in the process.
* When this value is reached, the cache manager deletes the oldest frozen
```

```
  summaries from the list.
* Default: 200000 (200K)
```

Raft Statemachine configuration

```
[raft_statemachine]

disabled = <boolean>
* Set to true to disable the raft statemachine.
* This feature requires search head clustering to be enabled.
* Any consensus replication among search heads uses this feature.
* Default: true

replicate_search_peers = <boolean>
* Add/remove search-server request is applied on all members
  of a search head cluster, when this value to set to true.
* Requires a healthy search head cluster with a captain.

[watchdog]
disabled = <boolean>
* Disables thread monitoring functionality.
* Any thread that has been blocked for more than 'responseTimeout' milliseconds
  is logged to $SPLUNK_HOME/var/log/watchdog/watchdog.log
* Default: false.

responseTimeout = <decimal>
* Maximum time, in seconds, that a thread can take to respond before the
  watchdog logs a 'thread blocked' incident.
* The minimum value for 'responseTimeout' is 0.1.
* If you set 'responseTimeout' to lower than 0.1, the setting uses the minimum
  value instead.
* Default: 8

actions = <comma-separated list>
* A comma-separated list of actions that execute sequentially when a blocked
  thread is encountered.
* The following actions can be included in the list: 'pstacks', 'script', and
  'bulletin'.
  * 'pstacks' enables call stack generation for a blocked thread.
    * Call stack generation gives the user immediate information on the
      potential thread bottleneck or deadlock.
    * The watchdog saves each call stack in a separate file in
      $SPLUNK_HOME/var/log/watchdog with the following file name format:
      wd_stack_<pid>_<thread_name>_%Y_%m_%d_%H_%M_%S.%f_<uid>.log.
  * 'script' executes the script configured by the
    [watchdogaction:script] stanza.
  * 'bulletin' causes a message to be displayed on the web interface.
* NOTE: Use this setting only under the guidance of a Splunk Support engineer.
  It might degrade performance by increasing CPU and disk usage.
* Default: empty list (no action executed)

actionsInterval = <decimal>
* The interval, in seconds, that the watchdog uses while tracing a blocked
  thread. The watchdog executes each action every 'actionsInterval' seconds.
* The minimum value for 'actionsInterval' is 0.01.
* If you set 'actionsInterval' to lower than 0.01, the setting uses the minimum
  value instead.
* NOTE: A very brief interval might reduce performance by increasing CPU usage.
  Frequently-executed actions can also slow down performance.
* Default: 1
```

```
pstacksEndpoint = <boolean>
* Enables pstacks endpoint at /services/server/pstacks
* Endpoint allows ad-hoc pstacks generation of all running threads.
* This setting is ignored if 'watchdog' is not enabled.
* NOTE: This setting should be used only during troubleshooting and only if you
  have been explicitly asked to set it by a Splunk Support engineer.
* Default: true

usePreloadedPstacks = <boolean>
* Use preloaded wrapper to enable pstacks.
* NOTE: This setting should be changed only during troubleshooting and only if you
  have been explicitly asked to disable it by a Splunk Support engineer.
* Default: true

[watchdog:<threadname>]
* Settings under this stanza apply only to the specified "<threadname>".
* When these per-thread settings are defined, they take precedence over the
  default settings in the [watchdog] stanza.
* NOTE: Use this feature only under the guidance of Splunk Support. A Splunk
  engineer will provide the <threadname>.

disabled = <boolean>
* Disables thread monitoring for the specified thread.
* If the thread has been blocked for more than 'responseTimeout' milliseconds
  the Splunk platform logs it to $SPLUNK_HOME/var/log/watchdog/watchdog.log
* Default: false.

responseTimeout = <decimal>
* Maximum time, in seconds, that this thread can take to respond before the
  watchdog logs a 'thread blocked' incident.
* The minimum value for 'responseTimeout' is 0.1.
* If you set 'responseTimeout' to lower than 0.1, the setting uses the minimum
  value instead.
* Default: 8

actions = <comma-separated list>
* The actions that are to execute sequentially when this
  thread is blocked.
* The following actions can be included in the list: 'pstacks', 'script', and
  'bulletin'.
  * 'pstacks' enables call stack generation for the blocked thread.
    * Call stack generation gives the user immediate information on the
      potential thread bottleneck or deadlock.
    * The watchdog saves each call stack in a separate file in
      $SPLUNK_HOME/var/log/watchdog with the following file name format:
      wd_stack_<pid>_<thread_name>_%Y_%m_%d_%H_%M_%S.%f_<uid>.log.
  * 'script' executes the script configured by the
    [watchdogaction:script] stanza.
  * 'bulletin' causes a message to be displayed on the web interface.
* NOTE: Use this setting only under the guidance of a Splunk Support engineer.
  It might degrade performance by increasing CPU and disk usage.
* Default: empty list (no action executed)

actionsInterval = <decimal>
* The interval, in seconds, that the watchdog uses while tracing this blocked
  thread. The watchdog executes each action every 'actionsInterval' seconds.
* The minimum value for 'actionsInterval' is 0.01.
* If you set 'actionsInterval' to lower than 0.01, the setting uses the minimum
  value instead.
* NOTE: A very brief interval might reduce performance by increasing CPU usage.
  Frequently-executed actions can also slow down performance.
```

```
* Default: 1

[watchdogaction:pstacks]
* Setting under this stanza are ignored if 'pstacks' is not enabled in the
  'actions' list.
* NOTE: Change these settings only during troubleshooting, and if you have
  been asked to set it by a Splunk Support engineer. It can affect performance
  by increasing CPU and disk usage.

dumpAllThreads = <boolean>
* Determines whether or not the watchdog saves stacks of all monitored threads
  when it encounters a blocked thread.
* If you set 'dumpAllThreads' to true, the watchdog generates call stacks for
  all threads, regardless of thread state.
* Default: true

stacksBufferSizeOrder = <unsigned integer>
* The maximum number of call stacks an internal queue can hold.
* The watchdog uses the internal queue to temporarily store a call stack between
  the time the watchdog generates the call stack and the time it saves the call
  stack to a file.
* Increase the value of this setting if you see gaps in stack files due to high
  frequency of call stack generation. This might occur when, for example, you
  set 'stacksBufferSizeOrder' to a very low value, or if the number of threads
  is high.
* This number must be in the range 1 to 16.
* The watchdog uses this value to calculate the real size of the buffer, whose
  value must be a power of 2. For example, if 'stackBufferSizeOrder' is 4, the
  size of the buffer is 4 ^ 2, or 16.
* CAUTION: Setting to too low a value can cause dropped call stacks, and too
  high a value can cause increased memory consumption.
* Default: 14

maxStacksPerBlock = <unsigned integer>
* Maximum number of stacks that the watchdog can generate for a blocked thread.
* If you set 'dumpAllThreads' to true, the watchdog generates call stacks for
  all threads.
* If the blocked thread starts responding again, the count of stacks that the
  watchdog has generated resets to zero.
* If another thread blockage occurs, the watchdog begins generating stacks
  again, up to 'maxStacksPerBlock' stacks.
* When set to 0, an unlimited number of stacks will be generated.
  list.
* Default: 60

batchStacksThreshold = <unsigned integer>|auto
* The timeout, in milliseconds, after which the watchdog generates a new call
  stack file.
* This setting controls the batching up of call stacks when saving them to files,
  and can decrease the number of files the watchdog creates.
* When set to 0, batching is disabled.
* When set to 'auto', Splunk Enterprise determines the best frequency to create
  new call stack files.
* Default: auto

[watchdogaction:script]
* Setting under this stanza are ignored if 'script' is not enabled in the
  'actions' list.
* NOTE: Change these settings only during troubleshooting, and if you have
  been asked to set it by a Splunk Support engineer. It can affect performance
  by increasing CPU and disk usage.
```

```
path = <string>
* The path to the script to execute when the watchdog triggers the action.
* If you do not set 'path', the watchdog ignores the action.
* No default.

useShell = <boolean>
* A value of "true" means the script runs from the OS shell
  ("/bin/sh -c" on UNIX, "cmd.exe /c" on Windows)
* A value of "false" means the program will be run directly without attempting to
  expand shell metacharacters.
* Default: false

forceStop = <boolean>
* Whether or not the watchdog forcefully stops an active watchdog action script
  when a blocked thread starts to respond.
* Use this setting when, for example, the watchdog script has internal logic
  that controls its lifetime and must run without interruption.
* Default: false

forceStopOnShutdown = <boolean>
* If you set this setting to "true", the watchdog forcefully stops active
  watchdog scripts upon receipt of a shutdown request.
* Default: true
```

Parallel Reduce Configuration

```
[parallelreduce]
pass4SymmKey = <string>
* DEPRECATED. The setting is no longer required.

pass4SymmKey_minLength = <integer>
* The minimum length, in characters, that a 'pass4SymmKey' should be for a
  particular stanza.
* When you start the Splunk platform, if the 'pass4SymmKey' is shorter in length
  than what you specify with this setting, the platform warns you and advises
  that you change the pass4SymKey.
* If you use the CLI to modify 'pass4SymmKey' to a value that is shorter than what
  you specify with this setting, the platform warns you and advises that you
  change the pass4SymKey.
* Default: 12
```

Remote Storage of Search Artifacts Configuration

```
[search_artifact_remote_storage]
disabled = <boolean>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Optional.
* Specifies whether or not search artifacts should be stored remotely.
* Splunkd does not clean up artifacts from remote storage. Set up cleanup
  separately with the remote storage provider.
* Default: true

path = <path on server>
* The path setting points to the remote storage location where
```

```
  artifacts reside.
* The format for this setting is: <scheme>://<remote-location-specifier>
  * The "scheme" identifies a supported external storage system type.
  * The "remote-location-specifier" is an external system-specific string for
     identifying a location inside the storage system.
* These external systems are supported:
  * Object stores that support AWS's S3 protocol. These use the scheme "s3".
    For example, "path=s3://mybucket/some/path".
* This is a required setting. If you do not set the path, the search artifact
  remote storage feature is disabled.
* No default.

upload_archive_format = [none|tar.lz4]
* Creates a tarball so that the entire artifact can be stored as a single object
  on the remote storage.
* This can reduce time to upload and artifact when the remote storage has a high
  seek penalty and the search artifact contains more than 100 individual files
* Default : none
```

S3 specific settings

```
remote.s3.header.<http-method-name>.<header-field-name> = <String>
* Enable server-specific features, such as reduced redundancy, encryption,
  and so on, by passing extra HTTP headers with the REST requests.
* The <http-method-name> can be any valid HTTP method. For example, GET,
  PUT, or ALL, for setting the header field for all HTTP methods.
* Optional.
* Example: remote.s3.header.PUT.x-amz-storage-class = REDUCED_REDUNDANCY

remote.s3.access_key = <String>
* Specifies the access key to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, the indexer looks for these environment variables:
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the access key from the IAM role.
* Optional.
* No default.

remote.s3.secret_key = <String>
* Specifies the secret key to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, the indexer looks for these environment variables:
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the secret key from the IAM role.
* Optional.
* No default.

remote.s3.list_objects_version = v1|v2
* The AWS S3 Get Bucket (List Objects) Version to use.
* See AWS S3 documentation "GET Bucket (List Objects) Version 2" for details.
* Default: v1

remote.s3.signature_version = v2|v4
* The signature version to use when authenticating with the remote storage
  system supporting the S3 API.
* For 'sse-kms' server-side encryption scheme, you must use
```

```
  signature_version=v4.
* Optional.
* Default: v4

remote.s3.auth_region = <String>
* The authentication region to use for signing requests when interacting with
  the remote storage system supporting the S3 API.
* Used with v4 signatures only.
* If unset and the endpoint (either automatically constructed or explicitly
  set with remote.s3.endpoint setting) uses an AWS URL
  (for example, https://s3-us-west-1.amazonaws.com), the instance attempts
  to extract the value from the endpoint URL (for example, "us-west-1").  See
  the description for the remote.s3.endpoint setting.
* If unset and an authentication region cannot be determined, the request
  will be signed with an empty region value.
* Optional.
* No default.

remote.s3.use_delimiter = <boolean>
* Specifies whether a delimiter (currently "guidSplunk") should be
  used to list the objects that are present on the remote storage.
* A delimiter groups objects that have the same delimiter value
  so that the listing process can be more efficient as it
  does not need to report similar objects.
* Optional.
* Default: true

remote.s3.supports_versioning = <boolean>
* Specifies whether the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage.
* This setting determines how splunkd removes data from remote storage.
  A value of "true" means splunkd will delete all versions of objects at
  time of data removal. Otherwise, A value of "false" means splunkd will
  use a simple DELETE
  (See https://docs.aws.amazon.com/AmazonS3/latest/dev/DeletingObjectVersions.html).
* Optional.
* Default: true

remote.s3.endpoint = <URL>
* The URL of the remote storage system supporting the S3 API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified and the indexer is running on EC2, the endpoint is
  constructed automatically based on the EC2 region of the instance where the
  indexer is running, as follows: https://s3-<region>.amazonaws.com
* Optional.
* Example: https://s3-us-west-2.amazonaws.com

remote.s3.multipart_download.part_size = <unsigned integer>
* Sets the download size of parts during a multipart download.
* This setting uses HTTP/1.1 Range Requests (RFC 7233) to improve throughput
  overall and for retransmission of failed transfers.
* A value of 0 disables downloading in multiple parts, i.e., files are always
  downloaded as a single (large) part.
* Do not change this value unless that value has been proven to improve
  throughput.
* Minimum value: 5242880 (5 MB)
* Optional.
* Default: 134217728 (128 MB)

remote.s3.multipart_upload.part_size = <unsigned integer>
```

```
* Sets the upload size of parts during a multipart upload.
* Minimum value: 5242880 (5 MB)
* Optional.
* Default: 134217728 (128 MB)

remote.s3.multipart_max_connections = <unsigned integer>
* Specifies the maximum number of HTTP connections to have in progress for
  either multipart download or upload.
* A value of 0 means unlimited.
* Default: 8

remote.s3.retry_policy = max_count
* Sets the retry policy to use for remote file operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a file operation is
    retried upon intermittent failure both for individual parts of a multipart
    download or upload and for files as a whole.
* Optional.
* Default: max_count

remote.s3.max_count.max_retries_per_part = <unsigned integer>
* When the remote.s3.retry_policy setting is max_count, sets the maximum number
  of times a file operation is retried upon intermittent failure.
* The count is maintained separately for each file part in a multipart download
  or upload.
* Optional.
* Default: 1

remote.s3.max_count.max_retries_in_total = <unsigned integer>
* When the remote.s3.retry_policy setting is max_count, sets the maximum number
  of times a file operation is retried upon intermittent failure.
* The count is maintained for each file as a whole.
* Optional.
* Default: 1

remote.s3.timeout.connect = <unsigned integer>
* Set the connection timeout, in milliseconds, to use when interacting with
  S3 for this volume.
* Optional.
* Default: 5000 (5 seconds)

remote.s3.timeout.read = <unsigned integer>
* Set the read timeout, in milliseconds, to use when interacting with S3
  for this volume.
* Optional.
* Default: 60000 (60 seconds)

remote.s3.timeout.write = <unsigned integer>
* Set the write timeout, in milliseconds, to use when interacting with S3
  for this volume.
* Optional.
* Default: 60000 (60 seconds)

remote.s3.sslVerifyServerCert = <boolean>
* If this is set to true, Splunk verifies certificate presented by S3
  server and checks that the common name/alternate name matches the
  ones specified in 'remote.s3.sslCommonNameToCheck'
  and 'remote.s3.sslAltNameToCheck'.
* Optional.
* Default: false
```

```
remote.s3.sslVersions = <comma-separated list>
* Comma-separated list of SSL versions to connect to 'remote.s3.endpoint'.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list
  but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Optional.
* Default: tls1.2

remote.s3.sslCommonNameToCheck = <commonName1>, <commonName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to
  true, splunkd checks the common name of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this
  list of common names.
* No default.

remote.s3.sslAltNameToCheck = <alternateName1>, <alternateName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to true,
  splunkd checks the alternate name(s) of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this list
  of subject alternate names.
* No default.

remote.s3.sslRootCAPath = <path>
* Full path to the Certificate Authority (CA) certificate PEM format file
  containing one or more certificates concatenated together. S3 certificate
  is validated against the CAs present in this file.
* Optional.
* Default: [sslConfig/caCertFile] in the server.conf file

remote.s3.cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Optional.
* Default: TLSv1+HIGH:TLSv1.2+HIGH:@STRENGTH

remote.s3.ecdhCurves = <comma separated list>
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* Splunk software only supports named curves specified
  by their SHORT names.
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Optional.
* No default.

remote.s3.dhFile = <path>
* PEM format Diffie-Hellman parameter file name.
* DH group size should be no less than 2048bits.
* This file is required in order to enable any Diffie-Hellman ciphers.
* Optional.
* No default.
```

```
remote.s3.encryption = sse-s3 | sse-kms | sse-c | none
* Specifies the scheme to use for Server-side Encryption (SSE) for
  data-at-rest.
* sse-s3: Check http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html
* sse-kms: Check http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingKMSEncryption.html
* sse-c: Check http://docs.aws.amazon.com/AmazonS3/latest/dev/ServerSideEncryptionCustomerKeys.html
* none: no Server-side encryption enabled. Data is stored unencrypted on
  the remote storage.
* Optional.
* Default: none

remote.s3.encryption.sse-c.key_type = kms
* Optional
* Determines the mechanism Splunk uses to generate the key for sending
  over to S3 for SSE-C.
* The only valid value is 'kms', indicating AWS KMS service.
* One must specify required KMS settings: e.g. remote.s3.kms.key_id
  for Splunk to start up while using SSE-C.
* Optional.
* Default: kms

remote.s3.encryption.sse-c.key_refresh_interval = <unsigned integer>
* Specifies the period, in seconds, at which a new key is generated and used
  for encrypting any new data being uploaded to S3.
* Optional.
* Default: 86400 (24 hours)

remote.s3.kms.key_id = <string>
* Required if remote.s3.encryption = sse-c | sse-kms
* Specifies the identifier for Customer Master Key (CMK) on KMS. It can be the
  unique key ID or the Amazon Resource Name (ARN) of the CMK or the alias
  name or ARN of an alias that refers to the CMK.
* Examples:
  Unique key ID: 1234abcd-12ab-34cd-56ef-1234567890ab
  CMK ARN: arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
  Alias name: alias/ExampleAlias
  Alias ARN: arn:aws:kms:us-east-2:111122223333:alias/ExampleAlias
* No default.

remote.s3.kms.access_key = <string>
* Similar to 'remote.s3.access_key'.
* If not specified, KMS access uses 'remote.s3.access_key'.
* Optional.
* No default.

remote.s3.kms.secret_key = <string>
* Optional.
* Similar to 'remote.s3.secret_key'.
* If not specified, KMS access uses 'remote.s3.secret_key'.
* Optional.
* No default.

remote.s3.kms.auth_region = <string>
* Required if 'remote.s3.auth_region' is not set and Splunk can not
  automatically extract this information.
* Similar to 'remote.s3.auth_region'.
* If not specified, KMS access uses 'remote.s3.auth_region'.
* No default.

remote.s3.kms.max_concurrent_requests = <unsigned integer>
* Limits maximum concurrent requests to KMS from this Splunk instance.
* NOTE: Can severely affect search performance if set to very low value.
```

```
* Optional.
* Default: 10

remote.s3.kms.<ssl_settings> = <...>
* Check the descriptions of the SSL settings for remote.s3.<ssl_settings>
  above. e.g. remote.s3.sslVerifyServerCert.
* Valid ssl_settings are sslVerifyServerCert, sslVersions, sslRootCAPath,
  sslAltNameToCheck, sslCommonNameToCheck, cipherSuite, ecdhCurves and dhFile.
* All of these are optional and fall back to same defaults as
  the 'remote.s3.<ssl_settings>'.


[hot_bucket_streaming]

slices_list_executor_workers = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Number of workers that do list operations to discover slices during bucket recovery.
* Must be greater than 0.
* Default: 4

slices_download_executor_workers = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Number of workers that download slices during bucket recovery.
* Must be greater than 0.
* Default: 10

slices_build_executor_workers = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Maximum number of parallel bucket rebuilds during bucket recovery.
* Must be greater than 0.
* Default: 4

slices_removal_executor_workers = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Number of workers that remove slices after a bucket rolls to warm or is rebuilt.
* Must be greater than 0.
* Default: 2

slices_upload_executor_workers = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Number of workers that upload slices from hot buckets.
* Must be greater than 0.
* Default: 10

slices_upload_executor_capacity = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Maximum number of queued slices to be uploaded. This affects the same thread
  pool that uses the 'slices_upload_executor_workers' setting.
* A value of 0 means that the queue capacity is unlimited.
* Default: 10

slices_upload_send_interval = <interval><unit>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Periodic send interval, in seconds, for the slices to be uploaded.
* Examples: 10s, 1m
```

```
* Must not be greater than 300s or 5m
* Default: 5s

slices_upload_size_threshold = <unsigned integer>[B|KB|MB]
* Currently not supported. This setting is related to a feature that is
  still under development.
* Slice size threshold.
* When this threshold is reached, slice coalescing ends and the accumulated
  slice is uploaded.
* Must be a positive number followed by a size suffix.
  * Valid suffixes: b: bytes, kb: kilobytes, mb: megabytes
  * Suffixes are case insensitive.
* Must not be greater than 10MB
* Default: 1MB

slices_upload_retry_pending = <unsigned integer>
* Currently not supported. This setting is related to a feature that is
  still under development.
* Maximum number of upload slices that could be pending retry due to failure to enqueue
  or failure to upload.
* Must not be greater than 1000
* Default: 25


[federated_search]
# This section contains settings for the data federation feature.

disabled = <boolean>
* Set this flag to 'false' to enable the data federation functionality on this
  instance.
* Default: false

transparent_mode = <boolean>
* A setting of 'true' means federated search transparent mode is enabled on this
  Splunk platform instance.
* Default: true

whole_search_execution_optimization = <boolean>
* A setting of 'true' means federated searches that involve only a single
  provider run phases 0 and 1 entirely on the remote search head.
  * When set to 'true', this setting can improve federated search performance
    and reduce the network bandwidth used by federated searches.
  * This setting is dynamically set to 'true' for federated searches involving
    Splunk Cloud Platform federated providers.
* A setting of 'false' means that federated searches that involve only a single
  provider might do some search processing on the local search head.
* Default: false

sendsDeltaBundle = <boolean>
* Set this flag to 'false' on a federated search head to disable it from sending
  delta knowledge object bundles to its remote providers. Full knowledge object bundles will be continued to
be sent to the remote providers.
* Set this flag to 'false' on a federated remote provider to disable it from sending delta knowledge object
bundles
  to it's indexers. Full knowledge object bundles will be continued to be sent to the remote provider
indexers.
* Default: true

receivesDeltaBundle = <boolean>
* Specifies whether federated providers can receive delta knowledge object
  bundles from federated search heads.
* When set to 'false' for a federated provider, the federated provider can't
```

```
  receive delta knowledge bundles from federated search heads.
  * The federated provider continues to receive full knowledge bundles from
    federated search heads even when this setting is set to 'false'.
* Default: true

syncProxyBundleToClusterMembers = <boolean>
* If you set up a load balancer in front of a search head cluster as a
  federated provider, this setting specifies whether the provider syncs
  proxy bundles among the cluster members.
* A setting of 'false' means that the federated provider will not sync the
  proxy bundle with the cluster members and federated searches using this
  provider will fail to run properly.
* Change this setting from its default only when instructed to do so by Splunk
  Support.
* Default: true


[distributed_leases]
sslVerifyServerCert = <boolean>
* A value of "true" means the instance authenticates the remote server endpoint that
  it is attempting to connect to.
* Default: false

sslVerifyServerName = <boolean>
* See the description of 'sslVerifyServerName' under the [sslConfig] stanza
  for details on this setting.
* Default: false

disabled = <boolean>
* Determines whether or not the distributed lease manager is enabled.
* Default: true


[search_state]
alert_store = local
* Specifies location of alert state store
* Default: local


suppression_store = local
* Specifies location of suppression state store
* Default: local




[manager_pages]
sanitize_uri_param = <boolean>
* Determines whether the URI parameter received in the manager pages will be
  sanitized.
* A value of 'true' means the URI parameter will be sanitized.
* A value of 'false' means the URI parameter will not be sanitized.
* Use this flag to opt out of URI parameter sanitization in case it causes any
  breaking changes.
* Note: It is critical to sanitize the URI parameter if possible as it can be
  abused.
* Default: true

[localProxy]
max_concurrent_requests = <decimal>
* Currently not supported. This setting relates to a feature that is
  still under development.
```

```
* The maximum number of concurrent requests to proxy by using the 'local-proxy'
  REST endpoint.
* Maximum accepted value for this setting is "100".
* Minimum accepted value for this setting is "1".
* Default: 10

response_timeout_ms = <decimal>
* Currently not supported. This setting relates to a feature that is
  still under development.
* The maximum time, in milliseconds, to wait for the proxy destination to complete a
  response.
* Maximum accepted value for this setting is "3600000" milliseconds (1 hour).
* Minimum accepted value for this setting is "100" milliseconds.
* CAUTION: Setting this to a value close to the lower bound might result in
  timeouts due to insufficient time for the operation to complete, causing
  error or interruption.
* CAUTION: Setting this to a value close to the upper bound might delay
  cleaning up unresponsive sessions.
* Default: 600000 (10 minutes)
```

server.conf.example

```
#   Version 9.2.2
#
# This file contains an example server.conf.  Use this file to configure SSL
# and HTTP server options.
#
# To use one or more of these configurations, copy the configuration block
# into server.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# Allow users 8 hours before they time out
[general]
sessionTimeout=8h
pass4SymmKey = changeme

# Listen on IPv6 in addition to IPv4...
listenOnIPv6 = yes
# ...but make all outgoing TCP connections on IPv4 exclusively
connectUsingIpVersion = 4-only

# Turn on SSL:

[sslConfig]
enableSplunkdSSL = true
useClientSSLCompression = true
serverCert = $SPLUNK_HOME/etc/auth/server.pem
sslPassword = password
sslRootCAPath = $SPLUNK_HOME/etc/auth/cacert.pem
certCreateScript = genMyServerCert.sh

[proxyConfig]
http_proxy = http://proxy:80
https_proxy = http://proxy:80
proxy_rules = *
```

```
no_proxy = localhost, 127.0.0.1, ::1

######## SSO Example ########
# This example trusts all logins from the splunk web server and localhost
# Note that a proxy to the splunk web server should exist to enforce
# authentication
[general]
trustedIP = 127.0.0.1

####### Cascading Replication Example ######
[cascading_replication]
pass4SymmKey = someSecret
max_replication_threads = auto
max_replication_jobs = 5
cascade_replication_plan_reap_interval = 1h
cascade_replication_plan_age = 8h
cascade_replication_plan_fanout = auto
cascade_replication_plan_topology = size_balanced
cascade_replication_plan_select_policy = random


############################################################################
# Set this node to be a cluster manager.
############################################################################


[clustering]
mode = manager
replication_factor = 3
pass4SymmKey = someSecret
search_factor = 2


############################################################################
# Set this node to be a peer to cluster manager "SplunkManager01" on port
# 8089.
############################################################################

[clustering]
mode = peer
manager_uri = https://SplunkManager01.example.com:8089
pass4SymmKey = someSecret

############################################################################
# Set this node to be a searchhead to cluster manager "SplunkManager01" on
# port 8089.
############################################################################
[clustering]
mode = searchhead
manager_uri = https://SplunkManager01.example.com:8089
pass4SymmKey = someSecret

############################################################################
# Set this node to be a searchhead to multiple cluster managers -
# "SplunkManager01" with pass4SymmKey set to 'someSecret and "SplunkManager02"
# with no pass4SymmKey set here.
############################################################################
[clustering]
mode = searchhead
manager_uri = clustermanager:east, clustermanager:west

[clustermanager:east]
```

```
manager_uri = https://SplunkManager01.example.com:8089
pass4SymmKey=someSecret

[clustermanager:west]
manager_uri = https://SplunkManager02.example.com:8089

############################################################################
#     Configuration file change tracker
#  To enable the feature, set 'disabled=false'.
#  Set 'mode=auto' to include all available features.
############################################################################
[config_change_tracker]
disabled = false
mode = auto
denylist=peer-apps|savedsearches\.conf$
exclude_fields = server.conf:general:pass4SymmKey, authentication.conf:authentication:*

############################################################################
# Open an additional non-SSL HTTP REST port, bound to the localhost
# interface (and therefore not accessible from outside the machine)  Local
# REST clients like the CLI can use this to avoid SSL overhead when not
# sending data across the network.
############################################################################
[httpServerListener:127.0.0.1:8090]
ssl = false


############################################################################
# Set modinput facing exec queue to 16MB.
############################################################################
[queue=execProcessorInternalQ]
maxSize = 16384KB
```

serverclass.conf

The following are the spec and example files for `serverclass.conf`.

serverclass.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attributes and values for defining server
# classes to which deployment clients can belong. These attributes and
# values specify what content a given server class member will receive from
# the deployment server.
#
# For examples, see serverclass.conf.example. You must reload the deployment
# server configuration ("splunk reload deploy-server"), or restart splunkd,
# for changes to this file to take effect.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles



#*******************************************************************************
# Configure the server classes used by a deployment server instance.
#
```

```
# Server classes are essentially categories. They use filters to control
# what clients they apply to, contain a set of applications, and might define
# deployment server behavior for the management of those applications. The
# filters can be based on DNS name, IP address, build number of client
# machines, platform, and the clientName. If a target machine
# matches the filter, then the deployment server deploys the apps and configuration
# content that make up the server class to that machine.

# Property Inheritance
#
# Stanzas in serverclass.conf go from general to more specific, in the
# following order:
# [global] -> [serverClass:<name>] -> [serverClass:<scname>:app:<appname>]
#
# Some properties defined in the [global] stanza can be
# overridden by a more specific stanza as it applies to them. If a global
# setting can be overridden, the description says so.
```

FIRST LEVEL: global ##########

```
# Global stanza that defines properties for all server classes.
[global]

disabled = <boolean>
* Toggles the deployment server off and on.
* Set to true to disable.
* Default: false

crossServerChecksum = <boolean>
* Ensures that each app has the same checksum across different deployment
  servers.
* Useful if you have multiple deployment servers behind a load-balancer.
* Default: false

excludeFromUpdate = <comma-separated list>
* Specifies paths to one or more top-level files or directories (and their
  contents) to exclude from being touched during app update.  Note that
  each comma-separated entry MUST be prefixed by "$app_root$/"
  to avoid warning messages.
* Can be overridden at the serverClass level.
* Can be overridden at the app level.
* Requires version 6.2.x or higher for both the deployment server and client.

repositoryLocation = <path>
* The repository of applications on the server machine.
* Can be overridden at the serverClass level.
* Default: $SPLUNK_HOME/etc/deployment-apps

syncMode = [none | sharedDir]
* Specifies whether deployment apps are shared across multiple deployment servers.
* A value of "none" means the set of deployment apps are specific to
  this deployment server only and are not shared with any other
  deployment servers.
* A value of "sharedDir" means multiple deployment servers share the same
  deployment app directory and will sync app bundles and serverclass.conf.
* Each deployment server specifies its app directory with the
  the 'repositoryLocation' setting.
```

```
* When the deployment server reloads, either through manual intervention via
  the CLI or the REST endpoint or automatically in response to the forwarder
  management interface, the deployment server updates the
  "_splunk_ds_info/_metadata" file in the shared deployment server app
  directory. The other deployment servers sharing the directory periodically
  check that file to determine whether they need to run a reload.
* Default: "none"

maxConcurrentDownloads = <positive integer>
* The maximum number of deployment clients that can simultaneously download
  the bundle from the deployment server.
* If a deployment client fails to download the bundle because of this setting,
  it retries the bundle download on the next phonehome until it successfully
  downloads the bundle.
* A value of "0" means there is no limit to the number of deployment clients
  that can simultaneously download.
* Default: 0

reloadCheckInterval = <integer>
* The interval, in seconds, between reload checks, where a deployment server
  determines if it must run a reload to sync its configurations.
* This setting only applies in the case where 'syncMode' has
  a value of "sharedDir".
* Default: 60

targetRepositoryLocation = <path>
* The location on the deployment client where the deployment server
  should install the apps.
* If this value is unset, or set to empty, the repositoryLocation path is used.
* Can be overridden at the [serverClass:<name>] level.
* Useful only with complex (for example, tiered) deployment strategies.
* Default: $SPLUNK_HOME/etc/apps, the live
  configuration directory for a Splunk Enterprise instance.

tmpFolder = <path>
* Working folder used by deployment server.
* Default: $SPLUNK_HOME/var/run/tmp

continueMatching = <boolean>
* Controls how configuration is layered across classes and server-specific
  settings.
* If true, configuration lookups continue matching server classes, beyond
  the first match.
* If false, only the first match is used.
* Matching is done in the order in which server classes are defined.
* A serverClass can override this property and stop the matching.
* Can be overridden at the serverClass level.
* Default: true

endpoint = <URL template string>
* The endpoint from which content a deployment client can download content.
  The deployment client knows how to substitute values for variables in the
  URL.
* You can supply any custom URL here, as long as it uses the specified
  variables.
* Need not be specified unless you have a very specific need, for example:
  To acquire deployment application files from a third-party Web server, for
  extremely large environments.
* Can be overridden at the serverClass level.
* Default: $deploymentServerUri$/services/streams/deployment?name=$tenantName$:$serverClassName$:$appName$

filterType = whitelist | blacklist
```

```
* The whitelist setting indicates a filtering strategy that pulls in a
  subset:
    * Items are considered to not match the stanza by default.
    * Items that match any whitelist entry, and do not match any blacklist
      entry, are considered to match the stanza.
    * Items that match any blacklist entry are not considered to match the
      stanza, regardless of whitelist.
* The blacklist setting indicates a filtering strategy that rules out a subset:
    * Items are considered to match the stanza by default.
    * Items that match any blacklist entry, and do not match any whitelist
      entry, are considered to not match the stanza.
    * Items that match any whitelist entry are considered to match the stanza.
* More briefly:
    * whitelist: default no-match
    * blacklist: default match
* Can be overridden at the serverClass level, and the serverClass:app level.
* Default: whitelist

whitelist.<n> = <clientName> | <IP address> | <hostname> | <instanceId>
blacklist.<n> = <clientName> | <IP address> | <hostname> | <instanceId>
* 'n' is an unsigned integer. The sequence may start at any value and may be
  non-consecutive.
* The value of this attribute is matched against several things in order:
    * Any clientName specified by the client in its deploymentclient.conf file
    * The IP address of the connected client
    * The hostname of the connected client, as provided by reverse DNS lookup
    * The hostname of the client, as provided by the client
    * For Splunk Enterprise version > 6.4, the instanceId of the client. This is
      a GUID string, for example: 'ffe9fe01-a4fb-425e-9f63-56cc274d7f8b'.
* All of these can be used with wildcards.  The asterisk character (*) matches
  any sequence of characters. For example:
    * Match a network range: 10.1.1.*
    * Match a domain: *.splunk.com
* Can be overridden at the serverClass level, and the serverClass:app level.
* There are no whitelist or blacklist entries by default.
* These patterns are PCRE regular expressions, with the following aids for
  easier entry:
    * You can specify '.' to mean '\.'
    * You can specify '*' to mean '.*'
* Matches are always case-insensitive; you do not need to specify the '(?i)' prefix.

# Note: Overriding one type of filter (whitelist/blacklist) causes the other to
# be overridden (and hence not inherited from parent) too.

# Example with filterType=whitelist:
#     whitelist.0=*.splunk.com
#     blacklist.0=printer.splunk.com
#     blacklist.1=scanner.splunk.com
# This causes all hosts in splunk.com, except 'printer' and 'scanner', to
# match this server class.

# Example with filterType=blacklist:
#     blacklist.0=*
#     whitelist.0=*.web.splunk.com
#     whitelist.1=*.linux.splunk.com
# This causes only the 'web' and 'linux' hosts to match the server class.
# No other hosts match.

# You can also use deployment client machine types (hardware type of host
# machines) to match deployment clients.
# This filter is used only if match of a client could not be decided using
# the whitelist/blacklist filters. The value of each machine type is
```

```
# designated by the hardware platform itself; a few common ones are:
#   linux-x86_64, windows-intel, linux-i686, freebsd-i386,
    darwin-i386, sunos-sun4u.
# The method for finding it varies by platform; once a deployment client is
# connected to the deployment server, however, you can determine the value of a
# deployment client's machine type with this Splunk CLI command on the
# deployment server:
#       <code>./splunk list deploy-clients</code>
# The <code>utsname</code> values in the output are the respective deployment
# clients' machine types.

whitelist.from_pathname = <pathname>
blacklist.from_pathname = <pathname>
* As as alternative to a series of (whitelist|blacklist).<n>, the <clientName>,
  <IP address>, and <hostname> list can be imported from <pathname> that is
  either a plain text file or a comma-separated values (CSV) file.
* May be used in conjunction with (whitelist|blacklist).select_field,
  (whitelist|blacklist).where_field, and (whitelist|blacklist).where_equals.
* If used by itself, then <pathname> specifies a plain text file where one
  <clientName>, <IP address>, or <hostname> is given per line.
* If used in conjunction with select_field, where_field, and where_equals, then
  <pathname> specifies a CSV file.
* The <pathname> is relative to $SPLUNK_HOME.
* May also be used in conjunction with (whitelist|blacklist).<n> to specify
  additional values, but there is no direct relation between them.
* At most one from_pathname may be given per stanza.

whitelist.select_field = <field name> | <positive integer>
blacklist.select_field = <field name> | <positive integer>
* Specifies which field of the CSV file contains the <clientName>, <IP address>,
  or <hostname> either by field name or number.
* If <field name> is given, then the first line of the CSV file MUST be a
  header line containing the name(s) of all the field(s) and the <field name>
  must specify which field contains the value(s) to be used. Note that field
  names are case-sensitive.
* If <positive integer> is given, then it specifies the column number (starting
  at 1) of the field that contains the value(s) to be used. In this case, the
  first line of the CSV file MUST NOT be a header line.
* MUST be used in conjunction with (whitelist|blacklist).from_pathname.
* May be used in conjunction with (whitelist|blacklist).where_field and
  (whitelist|blacklist).where_equals.
* At most one select_field may be given per stanza.

whitelist.where_field = <field name> | <positive integer>
blacklist.where_field = <field name> | <positive integer>
* Specifies that only a subset of values are to be selected from
  (whitelist|blacklist).select_field.
* Specifies which field of the CSV file contains values to be compared against
  for equality with the (whitelist|blacklist).where_equals values.
* Like (whitelist|blacklist).select_field, the field may be specified by either
  name or number.  However, select_field and where_field MUST be specified the
  same way, either BOTH by name or BOTH by number.
* MUST be used in conjunction with (whitelist|blacklist).select_field and
  (whitelist|blacklist).where_equals.
* At most one where_field may be given per stanza.

whitelist.where_equals = <comma-separated list>
blacklist.where_equals = <comma-separated list>
* Specifies the value(s) that the value of (whitelist|blacklist).where_field
  must equal in order to be selected via (whitelist|blacklist).select_field.
* If more than one value is specified (separated by commas), then the value
  of (whitelist|blacklist).where_field may equal ANY ONE of the values.
```

```
* Each value is a PCRE regular expression with the following aids for easier
  entry:
    * You can specify '.' to mean '\.'
    * You can specify '*' to mean '.*'
* Matches are always case-insensitive; you do not need to specify the '(?i)'
  prefix.
* MUST be used in conjunction with (whitelist|blacklist).select_field and
  (whitelist|blacklist).where_field.
* At most one where_equals may be given per stanza.

machineTypesFilter = <comma-separated list>
* Optional.
* Boolean OR logic is employed: a match against any element in the list
  constitutes a match.
* This filter is used in boolean AND logic with whitelist/blacklist filters.
  Only clients which match the whitelist/blacklist AND which match this
  machineTypesFilter are included.
  * In other words, the match is an intersection of the matches for the
    whitelist/blacklist and the matches for MachineTypesFilter.
* This filter can be overridden at the serverClass and serverClass:app
  levels.
* These patterns are PCRE regular expressions, with the following aids for
  easier entry:
    * You can specify '.' to mean '\.'
    * You can specify '*' to mean '.*'
* Matches are always case-insensitive; you do not need to specify the '(?i)'
  prefix.
* Unset by default.

packageTypesFilter = <comma-separated list>
* Optional.
* Boolean OR logic is employed: a match against any element in the list
  constitutes a match.
* This filter is used in boolean AND logic with 'whitelist'/'blacklist'
  filters. Only clients which match the 'whitelist'/'blacklist' AND which
  match this packageTypesFilter are included.
  * In other words, the match is an intersection of the matches for the
    'whitelist'/'blacklist' and the matches for 'packageTypesFilter'.
* You can override this filter at the serverClass and serverClass:app
  levels.
* These patterns are PCRE(Perl Compatible Regular Expressions) regular
  expressions, with the following aids for easier entry:
    * You can specify '.' to mean '\.'
    * You can specify '*' to mean '.*'
* Matches are always case-insensitive; you do not need to specify the '(?i)'
  prefix.
* Default: Not set

updaterRunningFilter = <boolean>
* This filter is used in boolean AND logic with 'whitelist'/'blacklist' filters.
  Only clients which match the 'whitelist'/'blacklist' AND which match this
  updaterRunningFilter are included.
  * In other words, the match is an intersection of the matches for the
    'whitelist'/'blacklist' and the matches for 'updaterRunningFilter'.
* The self-updater is a process that must be installed separately to
  upgrade the deployment client. This setting is applicable only if the
  self-updater is installed.
* A value of "true" means only the clients with self-updater running
  on the host are included.
* You can override this filter at the serverClass level and the
  serverClass:app level.
* Unset by default.
```

```
restartSplunkWeb = <boolean>
* If true, restarts SplunkWeb on the client when a member app or a directly
  configured app is updated.
* Can be overridden at the serverClass level and the serverClass:app level.
* Default: false

restartSplunkd = <boolean>
* If true, restarts splunkd on the client when a member app or a directly
  configured app is updated.
* Can be overridden at the serverClass level and the serverClass:app level.
* Default: false

issueReload = <boolean>
* If true, triggers a reload of internal processors at the client when a
  member app or a directly configured app is updated.
* If you don't want to immediately start using an app that is pushed to a
  client, you should set this to false.
* Default: false

restartIfNeeded = <boolean>
* This is only valid on forwarders that are newer than 6.4.
* If true and issueReload is also true, then when an updated app is deployed
  to the client, that client tries to reload that app. If it fails, it restarts.
* Default: false

stateOnClient = enabled | disabled | noop
* If set to "enabled", sets the application state to enabled on the client,
  regardless of state on the deployment server.
* If set to "disabled", set the application state to disabled on the client,
  regardless of state on the deployment server.
* If set to "noop", the state on the client is the same as on the
  deployment server.
* Can be overridden at the serverClass level and the serverClass:app level.
* Default: enabled

precompressBundles = <boolean>
* Controls whether the deployment server generates both .bundle and
  .bundle.gz files. The pre-compressed files offer improved performance as
  the deployment server is not required to compress the bundles on the fly
  for each client that it has to send the bundle to. However, this setting
  is only beneficial if there is no SSL compression in use and the client has
  support for HTTP compression.

* Deployment Server / server.conf
*   allowSslCompression = false
*   useHTTPServerCompression = true
*
* Deployment Client / server.conf
*   useHTTPClientCompression = true
*
* This option is inherited and available up to the serverclass level (not
  app). Apps belonging to server classes that required precompression are
  compressed, even if they belong to a server class which does not
  require precompression.
* Default: true

cronSchedule = <string>
* The cron schedule that is used to reload this serverclass in following format:
* "<minute> <hour> <day of month> <month> <day of week>"
* Special characters are acceptable. You can use combinations of "*",
  ",", "/", and "-" to specify wildcards, separate values, ranges
```

```
  of values, and step values. For example:
    * Run reload at midnight from Monday to Friday: 0 0 * * 1-5
    * Run reload at midnight on Dec 1: 0 0 1 12 *
    * Run reload every hour at hh:03, hh:23, hh:43: 03,23,43 * * * *
* This option is available only at the serverclass level.
* You must set 'cronSchedule' in order to run reload jobs automatically
  rather than manually.
* No default.
```

SECOND LEVEL: serverClass ###########

```
[serverClass:<serverClassName>]
* This stanza defines a server class. A server class is a collection of
  applications; an application may belong to multiple server classes.
* serverClassName is a unique name that is assigned to this server class.
* A server class can override all inheritable properties in the [global] stanza.
* A server class name may only contain: letters, numbers, spaces, underscores,
  dashes, dots, tildes, and the '@' symbol.  It is case-sensitive.

# NOTE:
# The following settings are all described in detail in the
# previous [global] section. They can be used in the serverClass stanza to
# override the global setting.
continueMatching = <boolean>
endpoint = <URL template string>
excludeFromUpdate = <comma-separated list>
filterType = whitelist | blacklist
whitelist.<n> = <clientName> | <IP address> | <hostname>
blacklist.<n> = <clientName> | <IP address> | <hostname>
machineTypesFilter = <comma-separated list>
packageTypesFilter = <comma-separated list>
updaterRunningFilter = <boolean>
restartSplunkWeb = <boolean>
restartSplunkd = <boolean>
issueReload = <boolean>
restartIfNeeded = <boolean>
stateOnClient = enabled | disabled | noop
repositoryLocation = <path>
targetRepositoryLocation = <path>
cronSchedule = <string>
```

THIRD LEVEL: app ###########

```
[serverClass:<server class name>:app:<app name>]
* This stanza maps an application (which must already exist in
  repositoryLocation) to the specified server class.
* server class name is the server class to which this content should be
  added.
* app name can be '*' or the name of an app:
    * The value '*' refers to all content in the repositoryLocation, adding
      it to this serverClass. '*' stanza cannot be mixed with named stanzas
      for a given server class.
    * The name of an app explicitly adds the app to a server class.
      Typically apps are named by the folders that contain them.
    * An application name, if it is not the special '*' sign explained
```

```
      directly above, may only contain: letters, numbers, spaces, underscores,
      dashes, dots, tildes, and the '@' symbol.  It is case-sensitive.

appFile=<file name>
* In cases where the app name is different from the file or directory name,
  you can use this parameter to specify the file name. Supported formats
  are: directories, .tar files, and .tgz files.

# NOTE: The following settings may override settings at the global or
  serverClass levels.
issueReload = <boolean>
restartIfNeeded = <boolean>
excludeFromUpdate = <comma-separated list>
filterType = whitelist | blacklist
whitelist.<n> = <clientName> | <IP address> | <hostname>
blacklist.<n> = <clientName> | <IP address> | <hostname>
machineTypesFilter = <comma-separated list>
packageTypesFilter = <comma-separated list>
updaterRunningFilter = <boolean>
stateOnClient = enabled | disabled | noop
```

serverclass.conf.example

```
#   Version 9.2.2
#
# Example 1
# Matches all clients and includes all apps in the server class

[global]
whitelist.0=*
# whitelist matches all clients.
[serverClass:AllApps]
[serverClass:AllApps:app:*]
# a server class that encapsulates all apps in the repositoryLocation


# Example 2
# Assign server classes based on dns names.

[global]

[serverClass:AppsForOps]
whitelist.0=*.ops.yourcompany.com
[serverClass:AppsForOps:app:unix]
[serverClass:AppsForOps:app:SplunkLightForwarder]

[serverClass:AppsForDesktops]
filterType=blacklist
# exclude everybody except the Windows desktop machines.
blacklist.0=*
whitelist.0=*.desktops.yourcompany.com
[serverClass:AppsForDesktops:app:SplunkDesktop]


# Example 3
# Deploy server class based on machine types

[global]
```

```
[serverClass:AppsByMachineType]
# Ensure this server class is matched by all clients. It is IMPORTANT to
# have a general filter here, and a more specific filter at the app level.
# An app is matched _only_ if the server class it is contained in was
# successfully matched!
whitelist.0=*

[serverClass:AppsByMachineType:app:SplunkDesktop]
# Deploy this app only to Windows boxes.
machineTypesFilter=windows-*

[serverClass:AppsByMachineType:app:unix]
# Deploy this app only to unix boxes - 32/64 bit.
machineTypesFilter=linux-i686, linux-x86_64

# Example 4
# Specify app update exclusion list.

[global]

# The local/ subdirectory within every app will not be touched upon update.
excludeFromUpdate=$app_root$/local

[serverClass:MyApps]

[serverClass:MyApps:app:SpecialCaseApp]
# For the SpecialCaseApp, both the local/ and lookups/ subdirectories will
# not be touched upon update.
excludeFromUpdate=$app_root$/local,$app_root$/lookups

# Example 5
# Control client reloads/restarts

[global]
restartSplunkd=false
restartSplunkWeb=true

# For this serverclass, we attempt to only reload the configuration files
# within the app, if we fail to reload ie if there's a conf in the app that
# requires a restart, the admin must restart the instance themselves
[serverClass:ReloadOnly]
issueReload=true

# This is an example of a best effort reloadable serverClass. ie we try to
# reload the app, but if there are files that require a restart, only then
# do we restart
[serverClass:tryReloadThenRestart]
issueReload=true
restartIfNeeded=true

# Example 6a
# Use (allow list|deny list) text file import.
[serverClass:MyApps]
whitelist.from_pathname = etc/system/local/clients.txt

# Example 6b
# Use (allow list|deny list) CSV file import to read all values from the Client
# field (ignoring all other fields).
[serverClass:MyApps]
whitelist.select_field = Client
whitelist.from_pathname = etc/system/local/clients.csv
```

```
# Example 6c
# Use (allow list|deny list) CSV file import to read some values from the Client
# field (ignoring all other fields) where ServerType is one of T1, T2, or
# starts with dc.
[serverClass:MyApps]
whitelist.select_field = Client
whitelist.from_pathname = etc/system/local/server_list.csv
whitelist.where_field = ServerType
whitelist.where_equals = T1, T2, dc*

# Example 6d
# Use (allow list|deny list) CSV file import to read some values from field 2
# (ignoring all other fields) where field 1 is one of T1, T2, or starts with
# dc.
[serverClass:MyApps]
whitelist.select_field = 2
whitelist.from_pathname = etc/system/local/server_list.csv
whitelist.where_field = 1
whitelist.where_equals = T1, T2, dc*
```

serverclass.seed.xml.conf

The following are the spec and example files for `serverclass.seed.xml.conf`.

serverclass.seed.xml.conf.spec

```
#   Version 9.2.2

<!--
# This configuration is used by deploymentClient to seed a Splunk installation with applications, at startup
time.
# This file should be located in the workingDir folder defined by deploymentclient.conf.
#
# An interesting fact - the DS -> DC communication on the wire also uses this XML format.
-->
<?xml version="1.0"?>
<deployment name="somename">

    <!--
    # The endpoint from which all apps can be downloaded.  This value can be overridden by serviceClass or
ap declarations below.
    # In addition, deploymentclient.conf can control how this property is used by deploymentClient - see
deploymentclient.conf.spec.
    -->
    <endpoint>$deploymentServerUri$/services/streams/deployment?name=$serviceClassName$:$appName$<
/endpoint>

    <!--
    # The location on the deploymentClient where all applications will be installed. This value can be
overridden by serviceClass or
    # app declarations below.
    # In addition, deploymentclient.conf can control how this property is used by deploymentClient - see
deploymentclient.conf.spec.
    -->
    <repositoryLocation>$SPLUNK_HOME/etc/apps</repositoryLocation>

    <serviceClass name="serviceClassName">
        <!--
        # The order in which this service class is processed.
```

```
        -->
        <order>N</order>

        <!--
        # DeploymentClients can also override these values using serverRepositoryLocationPolicy and
serverEndpointPolicy.
        -->
        <repositoryLocation>$SPLUNK_HOME/etc/myapps</repositoryLocation>
        <endpoint>splunk.com/spacecake/$serviceClassName$/$appName$.tgz</endpoint>

        <!--
        # Please See serverclass.conf.spec for how these properties are used.
        -->
        <continueMatching>true</continueMatching>
        <restartSplunkWeb>false</restartSplunkWeb>
        <restartSplunkd>false</restartSplunkd>
        <stateOnClient>enabled</stateOnClient>

        <app name="appName1">
            <!--
            # Applications can override the endpoint property.
            -->
            <endpoint>splunk.com/spacecake/$appName$</endpoint>
        </app>
        <app name="appName2"/>

    </serviceClass>
</deployment>
```

serverclass.seed.xml.conf.example

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment name="root">
  <serverClass name="spacecake_apps">
    <app name="app_0">
      <repositoryLocation>$SPLUNK_HOME/etc/myapps</repositoryLocation>
      <!-- Download app_0 from the given location -->
      <endpoint>splunk.com/spacecake/apps/app_0.tgz</endpoint>
    </app>
    <app name="app_1">
      <repositoryLocation>$SPLUNK_HOME/etc/myapps</repositoryLocation>
      <!-- Download app_1 from the given location -->
      <endpoint>splunk.com/spacecake/apps/app_1.tgz</endpoint>
    </app>
  </serverClass>
  <serverClass name="foobar_apps">
    <!-- construct url for each location based on the scheme below and download each app -->
    <endpoint>foobar.com:5556/services/streams/deployment?name=$serverClassName$_$appName$.bundle<
/endpoint>
    <app name="app_0"/>
    <app name="app_1"/>
    <app name="app_2"/>
  </serverClass>
  <serverClass name="local_apps">
    <endpoint>foo</endpoint>
    <app name="app_0">
      <!-- app present in local filesystem -->
      <endpoint>file:/home/johndoe/splunk/ds/service_class_2_app_0.bundle</endpoint>
```

```
    </app>
    <app name="app_1">
      <!-- app present in local filesystem -->
      <endpoint>file:/home/johndoe/splunk/ds/service_class_2_app_1.bundle</endpoint>
    </app>
    <app name="app_2">
      <!-- app present in local filesystem -->
      <endpoint>file:/home/johndoe/splunk/ds/service_class_2_app_2.bundle</endpoint>
    </app>
  </serverClass>
</deployment>
```

setup.xml.conf

The following are the spec and example files for `setup.xml.conf`.

setup.xml.conf.spec

```
#   Version 9.2.2
#
#

<!--
This file describes the setup XML config and provides some examples.

Note that setup XML is not supported in Splunk Cloud or on deployments with search head clustering.

setup.xml provides a Setup Screen that you provide to users to specify configurations
for an app. The Setup Screen is available when the user first runs the app or from the
Splunk Manager: Splunk > Manager > Apps > Actions > Set up

Place setup.xml in the app's default directory:

  $SPLUNK_HOME/etc/apps/<app>/default/setup.xml

The basic unit of work is an <input>, which is targeted to a triplet
(endpoint, entity, field) and other information used to model the data. For example
data type, validation information, name/label, etc.

The (endpoint, entity, field attributes) identifies an object where the input is
read/written to, for example:

   endpoint=saved/searches
   entity=MySavedSearch
   field=cron_schedule

The endpoint/entities addressing is relative to the app being configured. Endpoint/entity can
be inherited from the outer blocks (see below how blocks work).

Inputs are grouped together within a <block> element:

(1) blocks provide an iteration concept when the referenced REST entity is a regex

(2) blocks allow you to group similar configuration items

(3) blocks can contain <text> elements to provide descriptive text to the user.

(4) blocks can be used to create a new entry rather than edit an already existing one, set the
```

```
    entity name to "_new". NOTE: make sure to add the required field 'name' as
    an input.

(5) blocks cannot be nested

See examples below.


Block Node attributes:

endpoint - The REST endpoint relative to "https://hostname:port/servicesNS/nobody/<app-name>/"
           of entities/object the block/input addresses. Generally, an endpoint maps to a
           Splunk configuration file.

entity   - An object at the endpoint. Generally, this maps to a stanza name in a configuration file.
          NOTE: entity names should be URI encoded.

mode     - (bulk | iter) used if the entity attribute is a regular expression:

           o iter - (default value for mode) Iterate over all matching entities and provide a
                    separate input field for each.
           o bulk - Update all matching entities with the same value.

           NOTE: splunk interprets '*' as the regex '.*'

eai_search - a search to filter entities returned by an endpoint. If not specified, the following
             search is used: eai:acl.app="" OR eai:acl.app="<current-app>" This search matches
             only objects defined in the app which the setup page is being used for.

             NOTE: if objects from another app are allowed to be configured, any changes to those
                   objects will be stored in the current app.

enabled   - (true | false | in-windows | in-unix) whether this block is enabled or not
            o true          - (default) this block is enabled
            o false         - block disabled
            o in-windows    - block is enabled only in windows installations
            o in-unix       - block is enabled in non-windows installations

Input Node Attributes:

endpoint          - see description above (inherited from block)

entity            - see description above (inherited from block)

field             - <string> the field which is being configured

old_style_disable - <bool> whether to perform entity disabling by submitting the edited entity with the
following
                     field set: disabled=1. (This is only relevant for inputs whose
field=disabled|enabled).
                     Defaults to false.

Nodes within an <input> element can display the name of the entity and field values within the entity
on the setup screen. Specify $name$ to display the name of the entity. Use $<field_name>$ to specify
the value of a specified field.

 -->

<setup>
  <block title="Basic stuff" endpoint="saved/searches/" entity="foobar">
    <text> some description here </text>
```

```
    <input field="is_scheduled">
      <label>Enable Schedule for $name$</label>   <!-- this will be rendered as "Enable Schedule for foobar"
-->
      <type>bool</type>
    </input>

    <input field="cron_scheduled">
      <label>Cron Schedule</label>
      <type>text</type>
    </input>
    <input field="actions">
      <label>Select Active Actions</label>
      <type>list</type>
    </input>

    <!-- bulk update  -->
    <input entity="*" field="is_scheduled" mode="bulk">
      <label>Enable Schedule For All</label>
      <type>bool</type>
    </input>
  </block>

  <!-- iterative update in this block -->
  <block title="Configure search" endpoint="saved/eventtypes/" entity="*" mode="iter">
    <input field="search">
      <label>$name$ search</label>
      <type>string</type>
    </input>
    <input field="disabled">
      <label>disable $name$</label>
      <type>bool</type>
    </input>
  </block>

  <block title="Create a new eventtype" endpoint="saved/eventtypes/" entity="_new">
    <input target="name">
      <label>Name</label>
      <type>text</type>
    </input>
    <input target="search">
      <label>Search</label>
      <type>text</type>
    </input>
  </block>

  <block title="Add Account Info" endpoint="storage/passwords" entity="_new">
    <input field="name">
      <label>Username</label>
      <type>text</type>
    </input>
    <input field="password">
      <label>Password</label>
      <type>password</type>
    </input>
  </block>

  <!--  example config for "Windows setup" -->
  <block title="Collect local event logs" endpoint="admin/win-eventlogs/" eai_search="" >
    <text>
      Splunk for Windows needs at least your local event logs to demonstrate how to search them.
      You can always add more event logs after the initial setup in Splunk Manager.
    </text>
```

```
    <input entity="System" field="enabled" old_style_disable="true">
      <label>Enable $name$</label>
      <type>bool</type>
    </input>
    <input entity="Security" field="enabled"  old_style_disable="true">
      <label>Enable $name$</label>
      <type>bool</type>
    </input>
    <input entity="Application" field="enabled"  old_style_disable="true">
      <label>Enable $name$</label>
      <type>bool</type>
    </input>
  </block>

  <block title="Monitor Windows update logs" endpoint="data/inputs/monitor">
    <text>
      If you monitor the Windows update flat-file log, Splunk for Windows can show your patch history.
      You can also monitor other logs if you have them, such as IIS or DHCP logs, from Data Inputs in Splunk
Manager
    </text>
    <input entity="%24WINDIR%5CWindowsUpdate.log" field="enabled">
      <label>Enable $name$</label>
      <type>bool</type>
    </input>
  </block>
</setup>
```

setup.xml.conf.example

```
No example
```

source-classifier.conf

The following are the spec and example files for `source-classifier.conf`.

source-classifier.conf.spec

```
#   Version 9.2.2
#
# This file contains all possible options for configuring settings for the
# file classifier in source-classifier.conf.
#
# There is a source-classifier.conf in $SPLUNK_HOME/etc/system/default/ To
# set custom configurations, place a source-classifier.conf in
# $SPLUNK_HOME/etc/system/local/.  For examples, see
# source-classifier.conf.example. You must restart Splunk to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

```
ignored_model_keywords = <space-separated list of terms>
* Terms to ignore when generating a sourcetype model.
* To prevent sourcetype "bundles/learned/*-model.xml" files from containing
  sensitive terms (e.g. "bobslaptop") that occur very frequently in your
  data files, add those terms to ignored_model_keywords.

ignored_filename_keywords = <space-separated list of terms>
* Terms to ignore when comparing a new sourcename against a known
  sourcename, for the purpose of classifying a source.
```

source-classifier.conf.example

```
#   Version 9.2.2
#
# This file contains an example source-classifier.conf.  Use this file to
# configure classification
# of sources into sourcetypes.
#
# To use one or more of these configurations, copy the configuration block
# into source-classifier.conf in $SPLUNK_HOME/etc/system/local/. You must
# restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# terms to ignore when generating sourcetype model to prevent model from
# containing servernames
ignored_model_keywords = sun mon tue tues wed thurs fri sat sunday monday tuesday wednesday thursday friday
saturday jan feb mar apr may jun jul aug sep oct nov dec january february march april may june july august
september october november december 2003 2004 2005 2006 2007 2008 2009 am pm ut utc gmt cet cest cetdst met
mest metdst mez mesz eet eest eetdst wet west wetdst msk msd ist jst kst hkt ast adt est edt cst cdt mst mdt
pst pdt cast cadt east eadt wast wadt

# terms to ignore when comparing a sourcename against a known sourcename
ignored_filename_keywords = log logs com common event events little main message messages queue server
splunk
```

sourcetypes.conf

The following are the spec and example files for `sourcetypes.conf`.

sourcetypes.conf.spec

```
#  Version 9.2.2
#
# NOTE: sourcetypes.conf is a machine-generated file that stores the document
# models used by the file classifier for creating source types.

# Generally, you should not edit sourcetypes.conf, as most attributes are
# machine generated.  However, there are two attributes which you can change.
#
# There is a sourcetypes.conf in $SPLUNK_HOME/etc/system/default/ To set custom
# configurations, place a sourcetypes..conf in $SPLUNK_HOME/etc/system/local/.
```

```
# For examples, see sourcetypes.conf.example. You must restart Splunk to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
#     file wins.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.


_sourcetype = <value>
* Specifies the sourcetype for the model.
* Change this to change the model's sourcetype.
* Future sources that match the model will receive a sourcetype of this new
  name.

_source = <value>
* Specifies the source (filename) for the model.
```

sourcetypes.conf.example

```
#   Version 9.2.2
#
# This file contains an example sourcetypes.conf.  Use this file to configure
# sourcetype models.
#
# NOTE: sourcetypes.conf is a machine-generated file that stores the document
# models used by the file classifier for creating source types.
#
# Generally, you should not edit sourcetypes.conf, as most attributes are
# machine generated.  However, there are two attributes which you can change.
#
# To use one or more of these configurations, copy the configuration block into
# sourcetypes.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

#
# This is an example of a machine-generated sourcetype models for a fictitious
# sourcetype cadcamlog.
#

[/Users/bob/logs/bnf.x5_Thu_Dec_13_15:59:06_2007_171714722]
```

```
_source = /Users/bob/logs/bnf.x5
_sourcetype = cadcamlog
L---------- = 0.096899
L-t<_EQ> = 0.016473
```

splunk-launch.conf

The following are the spec and example files for `splunk-launch.conf`.

splunk-launch.conf.spec

```
#   Version 9.2.2

# splunk-launch.conf contains values used at startup time, by the Splunk
# command and by Windows services.
#

# Note: this conf file is different from most splunk conf files.  There is
# only one in the whole system, located at
# $SPLUNK_HOME/etc/splunk-launch.conf; further, there are no stanzas,
# explicit or implicit.  Finally, any splunk-launch.conf files in
# etc/apps/... or etc/users/... will be ignored.


# Lines beginning with a # are comments and are ignored.

#*******
# Environment variables
#
# Primarily, this file simply sets environment variables to be used by
# Splunk programs.
#
# These environment variables are the same type of system environment
# variables that can be set, on unix, using:
#   bourne shells:
#       $ export ENV_VAR=value
#   c-shells:
#       % setenv ENV_VAR value
#
# or at a windows command prompt:
#   C:\> SET ENV_VAR=value
#*******

<environment_variable>=<value>

* Any desired environment variable can be set to any value.
  Whitespace is trimmed from around both the key and value.
  Variable substitution (VAR=$OTHER_VAL) is not supported.
* Environment variables set here will be available to all Splunk
  platform processes, barring operating system limitations.


#*******
# Specific Splunk environment settings
#
# These settings are primarily treated as environment variables, though some
# have some additional logic (defaulting).
#
# There is no need to explicitly set any of these values in typical
```

```
# environments.
#*******

SPLUNK_HOME = <string>
* The fully qualified path to the Splunk platform instance installation directory.
* The comment in the auto-generated splunk-launch.conf is informational, not
  a live setting, and does not need to be uncommented.
* If not set, the Splunk platform automatically determines the location of SPLUNK_HOME
  based on the location of the splunk CLI executable.
    * Specifically, the parent of the directory containing splunk or splunk.exe
* Must be set if Common Criteria mode is enabled.
* NOTE: Splunk plans to submit Splunk Enterprise for Common Criteria
  evaluation. Splunk does not support using the product in Common
  Criteria mode until it has been certified by the National Information Assurance
  Partnership (NIAP). See the "Securing Splunk Enterprise" manual for information on
  the status of Common Criteria certification.
* Default: not set

SPLUNK_DB = <string>
* The comment in the auto-generated splunk-launch.conf is informational, not
  a live setting, and does not need to be uncommented.
* The fully qualified path to the directory containing the index
  directories for the Splunk platform instance.
* Primarily used by paths expressed in indexes.conf
* The comment in the autogenerated splunk-launch.conf is informational, not
  a live setting, and does not need to be uncommented.
* If unset, the path becomes $SPLUNK_HOME/var/lib/splunk (unix) or
     %SPLUNK_HOME%\var\lib\splunk (windows>)
* Default: not set

SPLUNK_BINDIP = <ip address>
* The network IP address that splunkd and splunkweb should bind to, as
  opposed to binding to the default for the local operating system.
* If not set, the Splunk platform makes no specific request to the operating
  system when binding to ports or opening a listening socket. This means it
  effectively binds to '*', meaning an unspecified bind. Operating system
  behavior and configuration controls the exact result in this case.
* NOTE: When using this setting you must update 'mgmtHostPort' in web.conf to
  match. Otherwise, the command line and splunkweb cannot reach splunkd.
* For splunkd, this sets both the management port and the ports that receive
  from forwarders.
* This setting is useful for a host with multiple IP addresses, either to enable
  or restrict access. But using a firewall is typically a superior
  method of restriction.
* Does not override web.conf/[settings]/server.socket_host for SplunkWeb
  if set; the latter is preferred when SplunkWeb behavior is the focus.
* Default: not set

SPLUNK_OS_USER = <string> | <nonnegative integer>
* The OS user whose privileges splunkd adopts when running.
* Example: SPLUNK_OS_USER=fnietzsche. Splunkd starts with a root login.
  Immediately upon starting, splunkd abandons the root user's privileges,
  and acquires fnietzsche's privileges. User fnietzsche owns any files
  that splunkd creates (index data, logs, etc.) When fnietzsche starts splunkd
  the next time, the files are readable.
* When 'splunk enable boot-start -user <user>' is invoked, SPLUNK_OS_USER
  is set to <user> as a side effect.
* On UNIX, username or apposite numeric UID are both acceptable;
  on Windows, only usernames are acceptable.
* Default: not set

SPLUNK_FIPS = [0|1]
```

```
* Whether or not the Splunk platform instance operates in Federal Information
  Processing Standards (FIPS) mode, and uses the algorithms and restrictions
  that apply to the FIPS Publication 140-2 standard.
* If the machine on which the Splunk platform instance operates runs a kernel
  that operates in FIPS mode, this setting is "true" by default.
* Configure this setting to ensure that your Splunk platform instance operates
  fully within US federal guidelines set by the FIPS publication.
* NOTE: This setting is one-time only.
  * If you need for the instance to be fully FIPS-compliant, configure it to
    "true" before you start it for the first time. If you do not do this,
    the Splunk secret key that the instance generates on first-time startup
    might not meet FIPS guidance.
  * If you configure it to "true" and then start the Splunk platform instance,
    you cannot later configure it to "false". You must reinstall the software.
* Running the Splunk platform in FIPS mode can result in the platform operating
  more slowly than if you ran it in normal mode.
* Default: 0

PYTHONHTTPSVERIFY = [0|1]
* Whether or not the Splunk platform instance sets up TLS validation for the httplib
  module in the Python interpreter embedded with the Splunk package.
* Default: 0

PYTHONUTF8 = [0|1]
* Determines whether the Splunk platform instance enables the UTF-8 mode
  in the Python interpreter embedded with the Splunk package.
* A value of 1 means UTF-8 mode is enabled.
* This setting applies regardless of the system locale encoding.
* Default: 1

#*******
# Service/server names.
#
# These settings are considered internal, and altering them is not
# supported.
#
# On Windows, they influence the expected name of the service;
# on UNIX they influence the reported name of the appropriate
# server or daemon process.
#
# On Linux distributions that run systemd, this is the name of the
# unit file for the service that Splunk Enterprise runs as.
# For example, if you set 'SPLUNK_SERVER_NAME' to 'splunk'
# then the corresponding unit file should be named 'splunk.service'.
#
# If you want to run multiple instances of Splunk as *services* on
# Windows, you must change the names for instances after the first.
# This is because the first instance takes up the service names
# 'Splunkd' and 'Splunkweb', and you may not have multiple services with
# same name.
#*******

SPLUNK_SERVER_NAME = <string>
* Names the splunkd server/service.
* Defaults to splunkd (UNIX), or Splunkd (Windows).

SPLUNK_WEB_NAME = <string>
* No longer used.

#*******
# File system check enable/disable
#
```

```
# CAUTION!
# USE OF THIS ADVANCED SETTING IS NOT SUPPORTED. IRREVOCABLE DATA LOSS
# CAN OCCUR. YOU USE THE SETTING SOLELY AT YOUR OWN RISK.
# CAUTION!
#
# When the Splunk software encounters a file system that it does not recognize,
# it runs a utility called 'locktest' to confirm that it can write to the
# file system correctly. If 'locktest' fails for any reason, splunkd
# cannot start.
#
# The following setting lets you temporarily bypass the 'locktest'
# check (for example, when a software vendor introduces a new default
# file system on a popular operating system). When it is active, splunkd
# starts regardless of its ability to interact with the file system.
#
# Use this setting if and only if:
#
# * You are a skilled Splunk administrator and know what you are doing.
# * You use Splunk software in a development environment.
# * You want to recover from a situation where the default
#   filesystem has changed outside your control, such as
#   during an operating system upgrade.
# * You want to recover from a situation where a Splunk bug
#   has invalidated a previously functional file system after an upgrade.
# * You want to evaluate the performance of a file system for which
#   Splunk has not yet offered support.
# * You have been given explicit instruction from Splunk Support to use
#   the setting to solve a problem where the Splunk software does not start
#   because of a failed file system check.
# * You understand and accept all the risks of using the setting,
#   up to and including LOSING ALL YOUR DATA WITH NO CHANCE OF RECOVERY
#   while the setting is active.
#
# If none of these scenarios applies to you, then DO NOT USE THE SETTING.
#
# CAUTION!
# USE OF THIS ADVANCED SETTING IS NOT SUPPORTED. IRREVOCABLE DATA LOSS
# CAN OCCUR. YOU USE THE SETTING SOLELY AT YOUR OWN RISK.
# CAUTION!
#*******

OPTIMISTIC_ABOUT_FILE_LOCKING = [0|1]
* Whether or not Splunk software skips the file system lock check on
  unrecognized file systems.
* CAUTION: USE THIS SETTING AT YOUR OWN RISK. YOU CAN LOSE ANY DATA
  THAT HAS BEEN INDEXED WHILE THE SETTING IS ACTIVE.
* When set to 1, Splunk software skips the file system check, and
  splunkd starts whether or not it can recognize the file system.
* Defaults to 0 (Run the file system check.)

SPLUNK_PYTHON_DONT_ESCAPE_PRINTABLE = 0|1
* Determines whether the Splunk Python interpreter escapes non-printable
  characters such as ASCII 0-32,127, when logging with the Python
  logging module.
* Exceptions: \t (chr(9)/0x09), \n (chr(10)/0x0a), \r (chr(13)/0x0d)
* When set to 1, scripts that log when using Splunk's Python interpreter
  will NOT escape non-printable characters and may be in log files
* Defaults to 0 (Non-printable characters WILL be escaped)

ENABLE_CPUSHARES = <boolean>
* Whether or not Splunk software adds 'CPUShares=1024' to the systemd service
  unit file named Splunkd.service by default, in /etc/systemd/system.
```

```
* Supported for only Linux.
* Defaults: true
```

splunk-launch.conf.example

```
No example
```

tags.conf

The following are the spec and example files for `tags.conf`.

tags.conf.spec

```
# Version 9.2.2
#
# This file contains possible attribute/value pairs for configuring tags. Set
# any number of tags for indexed or extracted fields.
#
# There is no tags.conf in $SPLUNK_HOME/etc/system/default/. To set custom
# configurations, place a tags.conf in $SPLUNK_HOME/etc/system/local/. For
# examples, see tags.conf.example. You must restart Splunk software to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[<fieldname>=<value>]

```
* The field name and value to which the tags in the stanza
  apply. For example, host=localhost.
* A tags.conf file can contain multiple stanzas. It is recommended that the
  value be URL encoded to avoid configuration file parsing errors, especially
  if the field value contains the following characters: \n, =, []
* Each stanza can refer to only one field/value pair.

<tag1> = <enabled|disabled>
<tag2> = <enabled|disabled>
<tag3> = <enabled|disabled>
* Enable or disable each <tag> for this specific field/value pair.
* While you can have multiple tags in a stanza (meaning that multiple tags are
  assigned to the same field/value combination), only one tag is allowed per
  stanza line. In other words, you can't have a list of tags on one line of the
  stanza.
* CAUTION: Do not put the <tag> value in quotes. For example,
  use foo=enabled, not "foo"=enabled.
```

tags.conf.example

```
#   Version 9.2.2
#
# This is an example tags.conf.  Use this file to define tags for fields.
#
```

```
# To use one or more of these configurations, copy the configuration block into
# tags.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# This first example presents a situation where the field is "host" and the
# three hostnames for which tags are being defined are "hostswitch,"
# "emailbox," and "devmachine." Each hostname has two tags applied to it, one
# per line. Note also that the "building1" tag has been applied to two hostname
# values (emailbox and devmachine).

[host=hostswitch]
pci = enabled
cardholder-dest = enabled

[host=emailbox]
email = enabled
building1 = enabled

[host=devmachine]
development = enabled
building1 = enabled

[src_ip=192.168.1.1]
firewall = enabled

[seekPtr=1cb58000]
EOF = enabled
NOT_EOF = disabled
```

telemetry.conf

The following are the spec and example files for `telemetry.conf`.

telemetry.conf.spec

```
# This file contains possible attributes and values for configuring global
# telemetry settings. Please note that enabling these settings would enable
# apps to collect telemetry data about app usage and other properties.
#
# There is no global, default telemetry.conf. Instead, a telemetry.conf may
# exist in each app in Splunk Enterprise.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top
#    of the file.
#  * Each conf file should have at most one default stanza. If there are
```

```
#    multiple default stanzas, attributes are combined. In the case of
#    multiple definitions of the same attribute, the last definition in the
#    file wins.
#  * If an attribute is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.
```

[general]

```
optInVersion = <number>
* An integer that identifies the set of telemetry data to be collected
* Incremented upon installation if the data set collected by Splunk has changed
* This field was introduced for version 2 of the telemetry data set. So,
  when this field is missing, version 1 is assumed.
* Should not be changed manually

optInVersionAcknowledged = <number>
* The latest optInVersion acknowledged by a user on this deployment
* While this value is less than the current optInVersion, a prompt for
  data collection opt-in will be shown to users with the
  edit_telemetry_settings capability at login
* Once a user confirms interaction with this login - regardless of
  opt-in choice - this number will be set to the value of optInVersion
* This gets set regardless of whether the user opts in using the opt-in
  dialog or the Settings > Instrumentation page
* If manually decreased or deleted, then a user that previously acknowledged
  the opt-in dialog will not be shown the dialog the next time they log in
  unless the related settings (dismissedInstrumentationOptInVersion and
  hideInstrumentationOptInModal) in their user-prefs.conf are also changed.
* Unset by default

sendLicenseUsage = true|false
* Send the licensing usage information of splunk/app to the app owner
* Defaults to true

sendAnonymizedUsage = true|false
* Send the anonymized usage information about various categories like
  infrastructure, utilization etc of splunk/app to Splunk, Inc
* Defaults to true

sendSupportUsage = true|false
* Send the support usage information about various categories like
  infrastructure, utilization etc of splunk/app to Splunk, Inc
* Defaults to false

sendAnonymizedWebAnalytics = true|false
* Send the anonymized usage information about user interaction with
  splunk performed through the web UI
* Defaults to true

precheckSendLicenseUsage = true|false
* Default value for sending license usage in opt in modal
* Defaults to true

precheckSendAnonymizedUsage = true|false
* Default value for sending anonymized usage in opt in modal
* Defaults to true

precheckSendSupportUsage = true|false
* Default value for sending support usage in opt in modal
* Defaults to true
```

```
showOptInModal = true|false
* DEPRECATED - see optInVersion and optInVersionAcknowledged settings
* Shows the opt in modal. DO NOT SET! When a user opts in, it will
  automatically be set to false to not show the modal again.
* Defaults to true

deploymentID = <string>
* A uuid used to correlate telemetry data for a single splunk
  deployment over time. The value is generated the first time
  a user opts in to sharing telemetry data.

deprecatedConfig = true|false
* Setting to determine whether the splunk deployment is following
  best practices for the platform as well as the app
* Defaults to false

retryTransaction = <string>
* Setting that is created if the telemetry conf updates cannot be delivered to
  the cluster master for the splunk_instrumentation app.
* Defaults to an empty string

swaEndpoint = <string>
* The URL to which swajs will forward UI analytics events
* If blank, swajs sends events to the Splunk MINT CDS endpoint.
* Blank by default

telemetrySalt = <string>
* A salt used to hash certain fields before transmission
* Autogenerated as a random UUID when splunk starts

scheduledHour = <number>
* Time of day, on a 24 hour clock, that the scripted input responsible for collecting telemetry data starts.
* The script begins at the top of the hour and completes, including running searches on the primary instance
in your deployment, after a few minutes.
* Defaults to 3

scheduledDay = <string>
* Number representing the weekday on which telemetry data collection is executed
* 0 represents Monday
* Defaults to every day (*)

reportStartDate = <string>
* Start date for the next telemetry data collection
* Uses format YYYY-MM-DD
* Defaults to empty string

bufferFlushTimeout = <number>
* Timeout for buffer flush, number in seconds
* Defaults to 600s

onCloudInstance = true|false
* Whether the instance is on cloud or on prem
* Defaults to false
```

telemetry.conf.example

```
# This file contains possible attributes and values for configuring global
# telemetry settings. Please note that enabling these settings would enable
# apps to collect telemetry data about app usage and other properties.
```

```
#
# There is no global, default telemetry.conf. Instead, a telemetry.conf may
# exist in each app in Splunk Enterprise.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[general]
sendLicenseUsage = false
sendAnonymizedUsage = false
sendAnonymizedWebAnalytics = false
precheckSendAnonymizedUsage = false
precheckSendLicenseUsage = true
showOptInModal = true
deprecatedConfig = false
scheduledHour = 16
reportStartDate = 2017-10-27
scheduledDay = 4
bufferFlushTimeout = 600
onCloudInstance = false
```

times.conf

The following are the spec and example files for `times.conf`.

times.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains possible attribute/value pairs for creating custom time
# ranges.
#
# Each stanza controls different search commands settings.
#
# There is a times.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name times.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see times.conf.example.
# You must restart the Splunk instance to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top
#     of the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
#     file wins.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
```

[<timerange_name>]

```
* The token to use when accessing time ranges through the API or command line.
* A times.conf file can contain multiple stanzas.

label = <string>
* The textual description used by the UI to reference this time range.
* Required

header_label = <string>
* The textual description used by the UI when displaying search results in
  this time range.
* Optional.
* Default: The <timerange_name>

earliest_time = <string>
* The string that represents the time of the earliest event to return,
  inclusive.
* The time can be expressed with a relative time identifier or in UNIX time.
* Optional.
* No default (No earliest time bound is used)

latest_time = <string>
* The string that represents the time of the earliest event to return,
  inclusive.
* The time can be expressed with a relative time identifier or in UNIX
  time.
* Optional.
* NOTE: events that occur in the future (relative to the server timezone)
       might be returned.
* No default (No latest time bound is used)

order = <integer>
* The key on which all custom time ranges are sorted, ascending.
* The default time range selector in the UI will merge and sort all time
  ranges according to the 'order' key, and then alphabetically.
* Optional.
* Default: 0

disabled = <integer>
* Specifies if the menu item is shown. Set to 1 to hide menu item.
* Optional.
* Default: 0

sub_menu = <submenu name>
* REMOVED.  This setting is no longer used.
```

```
is_sub_menu = <boolean>
* REMOVED.  This setting is no longer used.
```

[settings]

```
* List of flags that modify the panels that are displayed in the time range picker.

show_advanced = <boolean>
* Specifies if the 'Advanced' panel should be displayed in the time range picker.
* Optional.
* Default: true

show_date_range = <boolean>
* Specifies if the 'Date Range' panel should be displayed in the time range picker.
* Optional.
* Default: true

show_datetime_range = <boolean>
* Specifies if the 'Date & Time Range' panel should be displayed in the time range picker.
* Optional.
* Default: true

show_presets = <boolean>
* Specifies if the 'Presets' panel should be displayed in the time range picker.
* Optional.
* Default: true

show_realtime = <boolean>
* Specifies if the 'Realtime' panel should be displayed in the time range picker.
* Optional.
* Default: true

show_relative = <boolean>
* Specifies if the 'Relative' panel should be displayed in the time range picker.
* Optional.
* Default: true
```

times.conf.example

```
#   Version 9.2.2
#
# This is an example times.conf.  Use this file to create custom time ranges
# that can be used while interacting with the search system.
#
# To use one or more of these configurations, copy the configuration block
# into times.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# Note: These are examples.  Replace the values with your own customizations.


# The stanza name is an alphanumeric string (no spaces) that uniquely
# identifies a time range.
```

```
[this_business_week]

# Define the label used in the time range control
label = This business week

# Define the label to be used in display headers. If omitted the 'label' key
# will be used with the first letter lowercased.
header_label = during this business week
earliest_time = +1d@w1
latest_time = +6d@w6

# Define the ordering sequence of this time range.  All time ranges are
# sorted numerically, ascending. If the time range is in a sub menu and not
# in the main menu, this will determine the position within the sub menu.
order = 110


# a time range that only has a bound on the earliest time
#
[last_3_hours]
label = Last 3 hours
header_label = in the last 3 hours
earliest_time = -3h
order = 30


# Use epoch time notation to define the time bounds for the Fall Semester
# 2013, where earliest_time is 9/4/13 00:00:00 and latest_time is 12/13/13
# 00:00:00.
#
[Fall_2013]
label = Fall Semester 2013
earliest_time = 1378278000
latest_time = 1386921600

#
# Disable the realtime panel in the time range picker
[settings]
show_realtime = false
```

transactiontypes.conf

The following are the spec and example files for `transactiontypes.conf`.

transactiontypes.conf.spec

```
#   Version 9.2.2
#
# This file contains all possible attributes and value pairs for a
# transactiontypes.conf file.  Use this file to configure transaction searches
# and their properties.
#
# There is a transactiontypes.conf in $SPLUNK_HOME/etc/system/default/.  To set
# custom configurations, place a transactiontypes.conf in
# $SPLUNK_HOME/etc/system/local/. You must restart Splunk to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
```

```
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each conf file should have at most one default stanza. If there are
#     multiple default stanzas, attributes are combined. In the case of
#     multiple definitions of the same attribute, the last definition in the
#     file wins.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.


[<TRANSACTIONTYPE>]
* Create any number of transaction types, each represented by a stanza name and
  any number of the following attribute/value pairs.
* Use the stanza name, [<TRANSACTIONTYPE>], to search for the transaction in
  Splunk Web.
* If you do not specify a value for an attribute, the Splunk platform
  uses the default value.

maxspan = [<integer> s|m|h|d|-1]
* Set the maximum pause between the events in a transaction.
* Can be in seconds, minutes, hours, or days, or -1 for an unlimited timespan.
  * Example:  5s, 6m, 12h or 30d.
* This setting is accurately documented even though its name is inconsistent
  with its description.
* Default: maxspan=-1

maxpause = [<integer> s|m|h|d|-1]
* Set the maximum time span for the transaction.
* Can be in seconds, minutes, hours, or days, or -1 for an unlimited pause.
  * Example:  5s, 6m, 12h or 30d.
* This setting is accurately documented even though its name is inconsistent
  with its description.
* Default: maxpause=-1

maxevents = <integer>
* The maximum number of events in a transaction. This constraint is disabled if
  the value is a negative integer.
* Default: maxevents=1000

fields = <comma-separated list of fields>
* If set, each event must have the same field(s) to be considered part of the
  same transaction.
  * Example: fields=host,cookie
* Default: ""

connected =< boolean>
* Relevant only if 'fields' (see above) is not empty. Controls whether an event
  that is not inconsistent and not consistent with the fields of a transaction
  opens a new transaction (connected=true) or is added to the transaction.
* An event can be not inconsistent and not field-consistent if it contains
  fields required by the transaction but none of these fields has been
  instantiated in the transaction (by a previous event addition).
* Default: true
```

```
startswith=<transam-filter-string>
* A search or eval filtering expression which, if satisfied by an event, marks
  the beginning of a new transaction.
* Examples:
  * startswith="login"
  * startswith=(username=foobar)
  * startswith=eval(speed_field < max_speed_field)
  * startswith=eval(speed_field < max_speed_field/12)
* Default: empty string

endswith=<transam-filter-string>
* A search or eval filtering expression which, if satisfied by an event, marks
  the end of a transaction.
* Examples:
  * endswith="logout"
  * endswith=(username=foobar)
  * endswith=eval(speed_field > max_speed_field)
  * endswith=eval(speed_field > max_speed_field/12)
* Default: empty string

* For 'startswith' and 'endswith' <transam-filter-string> has the following syntax:
* syntax:   "<search-expression>" | (<quoted-search-expression>) | eval(<eval-expression>)
* Where:
  * <search-expression>        is a valid search expression that does not contain quotes
  * <quoted-search-expression> is a valid search expression that contains quotes
  * <eval-expression>          is a valid eval expression that evaluates to a boolean.
                               For example, startswith=eval(foo<bar*2) matches events
                               where "foo" is less than 2 x "bar".
* Examples:
  * "<search expression>":       startswith="foo bar"
  * <quoted-search-expression>:  startswith=(name="mildred")
  * <quoted-search-expression>:  startswith=("search literal")
  * eval(<eval-expression>):     startswith=eval(distance/time < max_speed)

### memory constraint options ###

maxopentxn=<int>
* Specifies the maximum number of not yet closed transactions to keep in the
  open pool. When this limit is exceeded, the Splunk platform begins to evict
  transactions using LRU (least-recently-used memory cache algorithm) policy.
* The default value of this attribute is read from the transactions stanza in
  limits.conf.

maxopenevents=<int>
* Specifies the maximum number of events that can be part of open transactions.
  When this limit is exceeded, the Splunk platform begins to evict transactions
  using LRU (least-recently-used memory cache algorithm) policy.
* The default value of this attribute is read from the transactions stanza in
  limits.conf.

keepevicted=<bool>
* Specifies whether to output evicted transactions. Evicted transactions can be
  distinguished from non-evicted transactions by checking the value of the
  'evicted' field, which is set to "1" for evicted transactions.
* Default: keepevicted=false

### multivalue rendering options ###

mvlist=<bool>|<field-list>
* Specifies whether the multivalued fields of the transaction are (1) a
  list of the original events ordered in arrival order or (2) a set of unique
  field values ordered lexicographically.
```

```
* If a comma or space delimited list of fields is provided, only those fields
  are rendered as lists.
* Default: mvlist=f

delim=<string>
* A string used to delimit the original event values in the transaction event
  fields.
* Default: " " (a single space)

nullstr=<string>
* The string value to use when rendering missing field values as part of mv
  fields in a transaction.
* This option applies only to fields that are rendered as lists.
* Default: NULL

### values used only by the searchtxn search command ###

search=<string>
* A search string used to more efficiently seed transactions of this type.
* Make the value as specific as possible, to limit the number of events
  that must be retrieved to find transactions.
* Example: sourcetype="sendmaill_sendmail"
* Default: "*" (all events)
```

transactiontypes.conf.example

```
#   Version 9.2.2
#
# This is an example transactiontypes.conf.  Use this file as a template to
# configure transactions types.
#
# To use one or more of these configurations, copy the configuration block into
# transactiontypes.conf in $SPLUNK_HOME/etc/system/local/.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[default]
maxspan = 5m
maxpause = 2s
match = closest

[purchase]
maxspan  = 10m
maxpause = 5m
fields  = userid
```

transforms.conf

The following are the spec and example files for `transforms.conf`.

transforms.conf.spec

```
#   Version 9.2.2
```

```
#
# This file contains settings and values that you can use to configure
# data transformations.
#
# Transforms.conf is commonly used for:
# * Configuring host and source type overrides that are based on regular
#   expressions.
# * Anonymizing certain types of sensitive incoming data, such as credit
#   card or social security numbers.
# * Routing specific events to a particular index, when you have multiple
#   indexes.
# * Creating new index-time field extractions. NOTE: We do not recommend
#   adding to the set of fields that are extracted at index time unless it
#   is absolutely necessary because there are negative performance
#   implications.
# * Creating advanced search-time field extractions that involve one or more
#   of the following:
#   * Reuse of the same field-extracting regular expression across multiple
#     sources, source types, or hosts.
#   * Application of more than one regular expression to the same source,
#     source type, or host.
#   * Using a regular expression to extract one or more values from the values
#     of another field.
#   * Delimiter-based field extractions, such as extractions where the
#     field-value pairs are separated by commas, colons, semicolons, bars, or
#     something similar.
#   * Extraction of multiple values for the same field.
#   * Extraction of fields with names that begin with numbers or
#     underscores.
#   * NOTE: Less complex search-time field extractions can be set up
#           entirely in props.conf.
# * Setting up lookup tables that look up fields from external sources.
#
# All of the above actions require corresponding settings in props.conf.
#
# You can find more information on these topics by searching the Splunk
# documentation (http://docs.splunk.com/Documentation).
#
# There is a transforms.conf file in $SPLUNK_HOME/etc/system/default/. To
# set custom configurations, place a transforms.conf file in
# $SPLUNK_HOME/etc/system/local/.
#
# For examples of transforms.conf configurations, see the
# transforms.conf.example file.
#
# You can enable configuration changes made to transforms.conf by running this
# search in Splunk Web:
#
# | extract reload=t
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top
#    of the file.
```

```
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, settings are combined. In the case of
#    multiple definitions of the same setting, the last definition in the
#    file wins.
#  * If a setting is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.


[<unique_transform_stanza_name>]
* Name your stanza. Use this name when you configure field extractions,
  lookup tables, and event routing in props.conf. For example, if you are
  setting up an advanced search-time field extraction, in props.conf you
  would add REPORT-<class> = <unique_transform_stanza_name> under the
  [<spec>] stanza that corresponds with a stanza you've created in
  transforms.conf.
* Follow this stanza name with any number of the following setting/value
  pairs, as appropriate for what you intend to do with the transform.
* If you do not specify an entry for each setting, Splunk software uses
  the default value.

REGEX = <regular expression>
* Enter a regular expression to operate on your data.
* NOTE: This setting is valid for index-time and search-time field extraction.
* REGEX is required for all search-time transforms unless you are setting up
  an ASCII-only delimiter-based field extraction, in which case you can use
  DELIMS (see the DELIMS setting description, below).
* REGEX is required for all index-time transforms.
* REGEX and the FORMAT setting:
  * FORMAT must be used in conjunction with REGEX for index-time transforms.
    Use of FORMAT in conjunction with REGEX is optional for search-time
    transforms.
  * Name-capturing groups in the REGEX are extracted directly to fields.
    This means that you do not need to specify the FORMAT setting for
    simple search-time field extraction cases (see the description of FORMAT,
    below).
  * If the REGEX for a field extraction configuration does not have the
    capturing groups referenced in the FORMAT, searches that use that
    configuration will not return events.
  * The REGEX must have at least one capturing group, even if the FORMAT does
    not reference any capturing groups.
  * If the REGEX extracts both the field name and its corresponding field
    value, you can use the following special capturing groups if you want to
    skip specifying the mapping in FORMAT for search-time field extractions:
      _KEY_<string>, _VAL_<string>.
  * For example, the following are equivalent for search-time field extractions:
    * Using FORMAT:
      * REGEX  = ([a-z]+)=([a-z]+)
      * FORMAT = $1::$2
    * Without using FORMAT
      * REGEX  = (?<_KEY_1>[a-z]+)=(?<_VAL_1>[a-z]+)
    * When using either of the above formats, in a search-time extraction,
      the regular expression attempts to match against the source text,
      extracting as many fields as can be identified in the source text.
* Default: empty string

FORMAT = <string>
* NOTE: This option is valid for both index-time and search-time field
  extraction. Index-time field extraction configurations require the FORMAT
  setting. The FORMAT setting is optional for search-time field extraction
  configurations.
* This setting specifies the format of the event, including any field names or
  values you want to add.
```

```
* FORMAT is required for index-time extractions:
  * Use $n (for example $1, $2, etc) to specify the output of each REGEX
    match.
  * If REGEX does not have n groups, the matching fails.
  * The special identifier $0 represents what was in the DEST_KEY before the
    REGEX was performed.
  * At index time only, you can use FORMAT to create concatenated fields:
    * Example: FORMAT = ipaddress::$1.$2.$3.$4
  * When you create concatenated fields with FORMAT, "$" is the only special
    character. It is treated as a prefix for regular expression capturing
    groups only if it is followed by a number and only if the number applies to
    an existing capturing group. So if REGEX has only one capturing group and
    its value is "bar", then:
      * "FORMAT = foo$1" yields "foobar"
      * "FORMAT = foo$bar" yields "foo$bar"
      * "FORMAT = foo$1234" yields "foo$1234"
      * "FORMAT = foo$1\$2" yields "foobar\$2"
  * At index-time, FORMAT defaults to <stanza-name>::$1
* FORMAT for search-time extractions:
  * The format of this field as used during search time extractions is as
    follows:
    * FORMAT = <field-name>::<field-value>( <field-name>::<field-value>)*
      where:
      * field-name  = [<string>|$<capturing-group-number>]
      * field-value = [<string>|$<capturing-group-number>]
  * Search-time extraction examples:
      * 1. FORMAT = first::$1 second::$2 third::other-value
      * 2. FORMAT = $1::$2
  * If the REGEX for a field extraction configuration does not have the
    capturing groups specified in the FORMAT, searches that use that
    configuration will not return events.
  * If you configure FORMAT with a variable <field-name>, such as in the second
    example above, the regular expression is repeatedly applied to the source
    key to match and extract all field/value pairs in the event.
  * When you use FORMAT to set both the field and the value (such as FORMAT =
    third::other-value), and the value is not an indexed token, you must set the
    field to INDEXED_VALUE = false in fields.conf. Not doing so can cause
    inconsistent search results.
  * NOTE: You cannot create concatenated fields with FORMAT at search time.
    That functionality is only available at index time.
  * At search-time, FORMAT defaults to an empty string.

MATCH_LIMIT = <integer>
* Only set in transforms.conf for REPORT and TRANSFORMS field extractions.
  For EXTRACT type field extractions, set this in props.conf.
* Optional. Limits the amount of resources that are spent by PCRE
  when running patterns that do not match.
* Use this to set an upper bound on how many times PCRE calls an internal
  function, match(). If set too low, PCRE may fail to correctly match a pattern.
* Default: 100000

DEPTH_LIMIT = <integer>
* Only set in transforms.conf for REPORT and TRANSFORMS field extractions.
   For EXTRACT type field extractions, set this in props.conf.
* Optional. Limits the amount of resources that are spent by PCRE
  when running patterns that do not match.
* Use this to limit the depth of nested backtracking in an internal PCRE
  function, match(). If set too low, PCRE might fail to correctly match a
  pattern.
* Default: 1000

CLONE_SOURCETYPE = <string>
```

* This name is wrong; a transform with this setting actually clones and
  modifies events, and assigns the new events the specified source type.
* If CLONE_SOURCETYPE is used as part of a transform, the transform creates a
  modified duplicate event for all events that the transform is applied to via
  normal props.conf rules.
* Use this setting when you need to store both the original and a modified
  form of the data in your system, or when you need to to send the original and
  a modified form to different outbound systems.
  * A typical example would be to retain sensitive information according to
    one policy and a version with the sensitive information removed
    according to another policy. For example, some events may have data
    that you must retain for 30 days (such as personally identifying
    information) and only 30 days with restricted access, but you need that
    event retained without the sensitive data for a longer time with wider
    access.
* Specifically, for each event handled by this transform, a near-exact copy
  is made of the original event, and the transformation is applied to the
  copy. The original event continues along normal data processing unchanged.
* The <string> used for CLONE_SOURCETYPE selects the source type that is used
  for the duplicated events.
* The new source type MUST differ from the the original source type. If the
  original source type is the same as the target of the CLONE_SOURCETYPE,
  Splunk software makes a best effort to log warnings to splunkd.log, but this
  setting is silently ignored at runtime for such cases, causing the transform
  to be applied to the original event without cloning.
* The duplicated events receive index-time transformations & sed
  commands for all transforms that match its new host, source, or source type.
  * This means that props.conf matching on host or source will incorrectly be
    applied a second time.
* Can only be used as part of of an otherwise-valid index-time transform. For
  example REGEX is required, there must be a valid target (DEST_KEY or
  WRITE_META), etc as above.

LOOKAHEAD = <integer>
* NOTE: This option is valid for all index time transforms, such as
  index-time field creation, or DEST_KEY modifications.
* Optional. Specifies how many characters to search into an event.
* Default: 4096
  * You may want to increase this value if you have event line lengths that
    exceed 4096 characters (before linebreaking).

WRITE_META = <boolean>
* NOTE: This setting is only valid for index-time field extractions.
* Automatically writes REGEX to metadata.
* Required for all index-time field extractions except for those where
  DEST_KEY = _meta (see the description of the DEST_KEY setting, below)
* Use instead of DEST_KEY = _meta.
* Default: false

DEST_KEY = <KEY>
* NOTE: This setting is only valid for index-time field extractions.
* Specifies where Splunk software stores the expanded FORMAT results in
  accordance with the REGEX match.
* Required for index-time field extractions where WRITE_META = false or is
  not set.
* For index-time extractions, DEST_KEY can be set to a number of values
  mentioned in the KEYS section at the bottom of this file.
  * If DEST_KEY = _meta (not recommended) you should also add $0 to the
    start of your FORMAT setting. $0 represents the DEST_KEY value before
    Splunk software performs the REGEX (in other words, _meta).
    * The $0 value is in no way derived *from* the REGEX match. (It
      does not represent a captured group.)

```
* KEY names are case-sensitive, and should be used exactly as they appear in
  the KEYs list at the bottom of this file. (For example, you would say
  DEST_KEY = MetaData:Host, *not* DEST_KEY = metadata:host .)

DEFAULT_VALUE = <string>
* NOTE: This setting is only valid for index-time field extractions.
* Optional. The Splunk software writes the DEFAULT_VALUE to DEST_KEY if the
  REGEX fails.
* Default: empty string

SOURCE_KEY = <string>
* NOTE: This setting is valid for both index-time and search-time field
  extractions.
* Optional. Defines the KEY that Splunk software applies the REGEX to.
* For search time extractions, you can use this setting to extract one or
  more values from the values of another field. You can use any field that
  is available at the time of the execution of this field extraction
* For index-time extractions use the KEYs described at the bottom of this
  file.
  * KEYs are case-sensitive, and should be used exactly as they appear in
    the KEYs list at the bottom of this file. (For example, you would say
    SOURCE_KEY = MetaData:Host, *not* SOURCE_KEY = metadata:host .)
* If <string> starts with "field:" or "fields:" the meaning is changed.
  Instead of looking up a KEY, it instead looks up an already indexed field.
  For example, if a CSV field name "price" was indexed then
  "SOURCE_KEY = field:price" causes the REGEX to match against the contents
  of that field.  It's also possible to list multiple fields here with
  "SOURCE_KEY = fields:name1,name2,name3" which causes MATCH to be run
  against a string comprising of all three values, separated by space
  characters.
* SOURCE_KEY is typically used in conjunction with REPEAT_MATCH in
  index-time field transforms.
* Default: _raw
  * This means it is applied to the raw, unprocessed text of all events.

REPEAT_MATCH = <boolean>
* NOTE: This setting is only valid for index-time field extractions.
  This setting is ignored if DEST_KEY is _raw.
* Optional. When set to true, Splunk software runs 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 REGEX matches are expected per event.
* Default: false

INGEST_EVAL = <comma-separated list of evaluator expressions>
* NOTE: This setting is only valid for index-time field extractions.
* When you set INGEST_EVAL, this setting overrides all but one of other
  index-time settings (such as REGEX, DEST_KEY, etc) and declares
  the index-time extraction to be evaluator-based. The exception is
  STOP_PROCESSING_IF, which is applied after INGEST_EVAL setting.
* The expression takes a similar format to the search-time "|eval" command.
  For example "a=b+c*d" Just like the search-time operator, you can
  string multiple expressions together, separated by commas like
  "len=length(_raw), length_category=floor(log(len,2))".
* Keys which are commonly used with DEST_KEY or SOURCE_KEY (like
  "_raw", "queue", etc) can be used directly in the expression.
  Also available are values which would be populated by default when
  this event is searched ("source", "sourcetype", "host", "splunk_server",
  "linecount", "index"). Search-time calculated fields (the "EVAL-" settings
  in props.conf) are NOT available.
* When INGEST_EVAL accesses the "_time" variable, subsecond information is
```

  included. This is unlike regular-expression-based index-time extractions,
  where "_time" values are limited to whole seconds.
* By default, other variable names refer to index-time fields which are
  populated in "_meta" So an expression 'event_category=if(_raw LIKE "WARN %",
  "warning", "normal")' would append a new indexed field to _meta like
  "event_category::warning".
* You can force a variable to be treated as a direct KEY name by
  prefixing it with "pd:". You can force a variable to be always
  treated as a "_meta" field by prefixing it with "field:" Therefore
  the above expression could also be written as
  '$field:event_category$=if($pd:_raw$ LIKE "WARN %", "warning", "normal")'
* When writing to a _meta field, the default behavior is to add a new
  index-time field even if one exists with the same name, the same way
  WRITE_META works for regular-expression-based extractions. For example, "a=5,
  a=a+2" adds two index-time fields to _meta: "a::5 a::7". You can change this
  by using ":=" after the variable name. For example, setting "a=5, a:=a+2"
  causes Splunk software to add a single "a::7" field.
* NOTE: Replacing index-time fields is slower than adding them. It is best to
  only use ":=" when you need this behavior.
* The ":=" operator can also be used to remove existing fields in _meta
  by assigning the expression null() to them.
* When reading from an index-time field that occurs multiple times inside the
  _meta key, normally the first value is used. You can override this by
  prefixing the name with "mv:" which returns all of the values into a
  "multival" object. For example, if _meta contains the keys "v::a v::b" then
  'mvjoin(v,",")' returns "a" while 'mvjoin($mv:v$,",")' returns "a,b".
* Note that this "mv:" prefix does not change behavior when it writes to a
  _meta field. If the value returned by an expression is a multivalue, it
  always creates multiple index-time fields. For example,
  'x=mvappend("a","b","c")' causes the string "x::a x::b x::c" to be appended
  to the _meta key.
* Internally, the _meta key can hold values with various numeric types.
  Splunk software normally picks a type appropriate for the value that the
  expression returned. However, you can override this this choice by specifying
  a type in square brackets after the destination field name. For example,
  'my_len[int]=length(source)' creates a new field named "my_len" and forces it
  to be stored as a 64-bit integer inside _meta. You can force Splunk software
  to store a number as floating point by using the type "[float]". You can
  request a smaller, less-precise encoding by using "[float32]". If you want to
  store the value as floating point but also ensure that the Splunk software
  remembers the significant-figures information that the evaluation expression
  deduced, use "[float-sf]" or "[float32-sf]". Finally, you can force the
  result to be treated as a string by specifying "[string]".
* The capability of the search-time |eval operator to name the destination
  field based on the value of another field (like "| eval {destname}=1")
  is NOT available for index-time evaluations.
* Optional.
* Default: empty

DELIMS = <quoted string list>
* NOTE: This setting is only valid for search-time field extractions.
* IMPORTANT: If a value may contain an embedded unescaped double quote
  character, such as "foo"bar", use REGEX, not DELIMS. An escaped double
  quote (\") is ok. Non-ASCII delimiters also require the use of REGEX.
* Optional. Use DELIMS in place of REGEX when you are working with ASCII-only
  delimiter-based field extractions, where field values (or field/value pairs)
  are separated by delimiters such as colons, spaces, line breaks, and so on.
* Sets delimiter characters, first to separate data into field/value pairs,
  and then to separate field from value.
* Each individual ASCII character in the delimiter string is used as a
  delimiter to split the event.
* Delimiters must be specified within double quotes (eg. DELIMS="|,;").

```
  Special escape sequences are \t (tab), \n (newline), \r (carriage return),
  \\ (backslash) and \" (double quotes).
* When the event contains full delimiter-separated field/value pairs, you
  enter two sets of quoted characters for DELIMS:
* The first set of quoted delimiters extracts the field/value pairs.
* The second set of quoted delimiters separates the field name from its
  corresponding value.
* When the event only contains delimiter-separated values (no field names),
  use just one set of quoted delimiters to separate the field values. Then use
  the FIELDS setting to apply field names to the extracted values.
  * Alternately, Splunk software reads even tokens as field names and odd
    tokens as field values.
* Splunk software consumes consecutive delimiter characters unless you
  specify a list of field names.
* The following example of DELIMS usage applies to an event where
  field/value pairs are separated by '|' symbols and the field names are
  separated from their corresponding values by '=' symbols:
    [pipe_eq]
    DELIMS = "|", "="
* Default: ""

FIELDS = <quoted string list>
* NOTE: This setting is only valid for search-time field extractions.
* Used in conjunction with DELIMS when you are performing delimiter-based
  field extraction and only have field values to extract.
* FIELDS enables you to provide field names for the extracted field values,
  in list format according to the order in which the values are extracted.
* NOTE: If field names contain spaces or commas they must be quoted with " "
  To escape, use \.
* The following example is a delimiter-based field extraction where three
  field values appear in an event. They are separated by a comma and then a
  space.
    [commalist]
    DELIMS = ", "
    FIELDS = field1, field2, field3
* Default: ""

MV_ADD = <boolean>
* NOTE: This setting is only valid for search-time field extractions.
* Optional. Controls what the extractor does when it finds a field which
  already exists.
* If set to true, the extractor makes the field a multivalued field and
  appends the newly found value, otherwise the newly found value is
  discarded.
* Default: false

CLEAN_KEYS = <boolean>
* NOTE: This setting is only valid for search-time field extractions.
* Optional. Controls whether Splunk software "cleans" the keys (field names) it
  extracts at search time. "Key cleaning" is the practice of replacing any
  non-alphanumeric characters (characters other than those falling between the
  a-z, A-Z, or 0-9 ranges) in field names with underscores, as well as the
  stripping of leading underscores and 0-9 characters from field names.
* Add CLEAN_KEYS = false to your transform if you need to extract field
  names that include non-alphanumeric characters, or which begin with
  underscores or 0-9 characters.
* Default: true

KEEP_EMPTY_VALS = <boolean>
* NOTE: This setting is only valid for search-time field extractions.
* Optional. Controls whether Splunk software keeps field/value pairs when
  the value is an empty string.
```

```
* This option does not apply to field/value pairs that are generated by
  Splunk software autokv extraction. Autokv ignores field/value pairs with
  empty values.
* Default: false

CAN_OPTIMIZE = <boolean>
* NOTE: This setting is only valid for search-time field extractions.
* Optional. Controls whether Splunk software can optimize this extraction out
  (another way of saying the extraction is disabled).
* You might use this if you are running searches under a Search Mode setting
  that disables field discovery--it ensures that Software always discovers
  specific fields.
* Splunk software only disables an extraction if it can determine that none of
  the fields identified by the extraction will ever be needed for the successful
  evaluation of a search.
* NOTE: This option should be rarely set to false.
* Default: true

STOP_PROCESSING_IF = <evaluator expression>
* An evaluator expression that the regexreplacement processor uses to determine
  whether or not further processing is to occur for this event.
* If you set STOP_PROCESSING_IF, and the regexreplacement processor evaluates the
  expression that you supply to be true, then the processor stops further
  processing of this event.
* When you set STOP_PROCESSING_IF, like INGEST_EVAL, this setting overrides
  all of the other index-time settings (such as REGEX, DEST_KEY, etc) except
  for INGEST_EVAL. STOP_PROCESSING_IF executes after INGEST_EVAL.
* The processor treats the return value for <evaluator expression> as a boolean value.
  The final value depends on the value to which the expression initially calculates.
  See the following list:
    Numeric "0": false
    Boolean: true/false
    Null value: false
    Any other value: true
* If this setting appears in multiple rules, then the processor applies the settings
  in the following order:
  * All TRANSFORMS, alphabetically
  * All RULESETs, alphabetically
  * Within a single rule set class, where they appear in the rule set class
    determines the order. For example, in the following configuration:

    [rule1]
    STOP_PROCESSING_IF = <expression1>

    [rule2]
    STOP_PROCESSING_IF = <expression2>

    RULESET-ruleset1 = rule1, rule2, ...

    rule1 executes first because rule1 appears before rule2 in ruleset1.
    If <expression1> evaluates to "false", then rule2 and its associated
    STOP_PROCESSING_IF setting executes.
    If <expression1> evaluates to "true", then the processor skips rule2
    and all rules after rule2 in ruleset1.
* Optional.
* Default: empty string
* NOTE: This setting is only valid for index-time field extractions.
```

Lookup tables

```
# NOTE: Lookup tables are used ONLY during search time

filename = <string>
* Name of static lookup file.
* File should be in $SPLUNK_HOME/etc/system/lookups/, or in
  $SPLUNK_HOME/etc/apps/<app_name>/lookups/ if the lookup belongs to a specific
  app.
* If file is in multiple 'lookups' directories, no layering is done.
* Standard conf file precedence is used to disambiguate.
* Only file names are supported. Paths are explicitly not supported. If you
  specify a path, Splunk software strips the path to use the value after
  the final path separator.
* Splunk software then looks for this filename in
  $SPLUNK_HOME/etc/system/lookups/ or $SPLUNK_HOME/etc/apps/<app_name>/lookups/.
* Default: empty string

collection = <string>
* Name of the collection to use for this lookup.
* Collection should be defined in $SPLUNK_HOME/etc/apps/<app_name>/local/collections.conf
  for an <app_name>
* If collection is in multiple collections.conf file, no layering is done.
* Standard conf file precedence is used to disambiguate.
* Default: empty string (in which case the name of the stanza is used).

max_matches = <integer>
* The maximum number of possible matches for each input lookup value
  (range 1 - 1000).
* If the lookup is non-temporal (not time-bound, meaning the time_field
  setting is not specified), Splunk software uses the first <integer> entries,
  in file order.
* If the lookup is temporal, Splunk software uses the first <integer> entries
  in descending time order. In other words, only <max_matches> lookup entries
  are allowed to match. If the number of lookup entries exceeds <max_matches>,
  only the ones nearest to the lookup value are used.
* Default: 100 matches if the time_field setting is not specified for the
  lookup. If the time_field setting is specified for the lookup, the default is
  1 match.

min_matches = <integer>
* Minimum number of possible matches for each input lookup value.
* Default = 0 for both temporal and non-temporal lookups, which means that
  Splunk software outputs nothing if it cannot find any matches.
* However, if min_matches > 0, and Splunk software gets less than min_matches,
  it provides the default_match value provided (see below).

default_match = <string>
* If min_matches > 0 and Splunk software has less than min_matches for any
  given input, it provides this default_match value one or more times until the
  min_matches threshold is reached.
* Default: empty string.

case_sensitive_match = <boolean>
* If set to true, Splunk software performs case sensitive matching for all
  fields in a lookup table.
* If set to false, Splunk software performs case insensitive matching for all
  fields in a lookup table.
* NOTE: For KV Store lookups, a setting of 'case_sensitive_match=false' is
  honored only when the data in the KV Store lookup table is entirely in lower
  case. The input data can be in any case.
```

```
* For case sensitive field matching in reverse lookups see
  reverse_lookup_honor_case_sensitive_match.
* Default: true

reverse_lookup_honor_case_sensitive_match = <boolean>
* Determines whether field matching for a reverse lookup is case sensitive or
  case insensitive.
* When set to true, and 'case_sensitive_match' is true Splunk software performs
  case-sensitive matching for all fields in a reverse lookup.
* When set to true, and 'case_sensitive_match' is false Splunk software
  performs case-insensitive matching for all fields in a reverse lookup.
* When set to false, Splunk software performs case-insensitive matching for
  all fields in a reverse lookup.
* NOTE: This setting does not apply to KV Store lookups.
* Default: true

match_type = <string>
* A comma and space-delimited list of <match_type>(<field_name>)
  specification to allow for non-exact matching
* The available match_type values are WILDCARD, CIDR, and EXACT. Only fields
  that should use WILDCARD or CIDR matching should be specified in this list.
* Default: EXACT

external_cmd = <string>
* Provides the command and arguments to invoke to perform a lookup. Use this
  for external (or "scripted") lookups, where you interface with with an
  external script rather than a lookup table.
* This string is parsed like a shell command.
* The first argument is expected to be a python script (or executable file)
  located in $SPLUNK_HOME/etc/apps/<app_name>/bin.
* Presence of this field indicates that the lookup is external and command
  based.
* Default: empty string

fields_list = <string>
* A comma- and space-delimited list of all fields that are supported by the
  external command.

index_fields_list = <string>
* A comma- and space-delimited list of fields that need to be indexed
  for a static .csv lookup file.
* The other fields are not indexed and not searchable.
* Restricting the fields enables better lookup performance.
* Default: all fields that are defined in the .csv lookup file header.

external_type = [python|executable|kvstore|geo|geo_hex]
* This setting describes the external lookup type.
* Use 'python' for external lookups that use a python script.
* Use 'executable' for external lookups that use a binary executable, such as a
  C++ executable.
* Use 'kvstore' for KV store lookups.
* Use 'geo' for geospatial lookups.
* 'geo_hex' is reserved for the geo_hex H3 lookup.
* Default: python

python.version = {default|python|python2|python3}
* For Python scripts only, selects which Python version to use.
* Set to either "default" or "python" to use the system-wide default Python
  version.
* Optional.
* Default: Not set; uses the system-wide Python version.
```

```
time_field = <string>
* Used for temporal (time-bound) lookups. Specifies the name of the field
  in the lookup table that represents the timestamp.
* Default: empty string
  * This means that lookups are not temporal by default.

time_format = <string>
* For temporal lookups this specifies the 'strptime' format of the timestamp
  field.
* You can include subseconds but Splunk software ignores them.
* Default: %s.%Q (seconds from unix epoch in UTC and optional milliseconds)

max_offset_secs = <integer>
* For temporal lookups, this is the maximum time (in seconds) that the event
  timestamp can be later than the lookup entry time for a match to occur.
* Default: 2000000000, or the offset in seconds from 0:00 UTC Jan 1, 1970.
  Whichever is reached first.

min_offset_secs = <integer>
* For temporal lookups, this is the minimum time (in seconds) that the event
  timestamp can be later than the lookup entry timestamp for a match to
  occur.
* Default: 0

batch_index_query = <boolean>
* For large file-based lookups, batch_index_query determines whether queries
  can be grouped to improve search performance.
* Default (this level): not set
* Default (global level, at limits.conf): true

allow_caching = <boolean>
* Allow output from lookup scripts to be cached
* Default: true

cache_size = <integer>
* Cache size to be used for a particular lookup. If a previously looked up
  value is already present in the cache, it is applied.
* The cache size represents the number of input values for which to cache
  output values from a lookup table.
* Do not change this value unless you are advised to do so by Splunk Support or
  a similar authority.
* Default: 10000

max_ext_batch = <integer>
* The maximum size of external batch (range 1 - 1000).
* This setting applies only to KV Store lookup configurations.
* Default: 300

filter = <string>
* Filter results from the lookup table before returning data. Create this filter
  like you would a typical search query using Boolean expressions and/or
  comparison operators.
* For KV Store lookups, filtering is done when data is initially retrieved to
  improve performance.
* For CSV lookups, filtering is done in memory.

feature_id_element = <string>
* If the lookup file is a kmz file, this field can be used to specify the xml
  path from placemark down to the name of this placemark.
* This setting applies only to geospatial lookup configurations.
* Default: /Placemark/name
```

```
check_permission = <boolean>
* Specifies whether the system can verify that a user has write permission to a
  lookup file when that user uses the outputlookup command to modify that file.
  If the user does not have write permissions, the system prevents the
  modification.
* The check_permission setting is only respected when you set
  'outputlookup_check_permission'
  to "true" in limits.conf.
* You can set lookup table file permissions in the .meta file for each lookup
  file, or through the Lookup Table Files page in Settings. By default, only
  users who have the admin or power role can write to a shared CSV lookup file.
* This setting applies only to CSV lookup configurations.
* Default: false

replicate = <boolean>
* Indicates whether to replicate CSV lookups to indexers.
* When false, the CSV lookup is replicated only to search heads in a search
  head cluster so that input lookup commands can use this lookup on the search
  heads.
* When true, the CSV lookup is replicated to both indexers and search heads.
* Only for CSV lookup files.
* Note that replicate=true works only if it is included in the replication
  allow list. See the 'replicationAllowlist' setting in distSearch.conf.
* Default: true
```

METRICS - STATSD DIMENSION EXTRACTION

Metrics

```
[statsd-dims:<unique_transforms_stanza_name>]
* 'statsd-dims' prefix indicates this stanza is applicable only to statsd metric
  type input data.
* This stanza is used to define regular expression to match and extract
  dimensions out of statsd dotted name segments.
* By default, only the unmatched segments of the statsd dotted name segment
  become the metric_name.

REGEX = <regular expression>
* Splunk software supports a named capturing group extraction format to provide
  dimension names of the corresponding values being extracted out. For example:
    (?<dim1>group)(?<dim2>group)..

REMOVE_DIMS_FROM_METRIC_NAME = <boolean>
* If set to false, the matched dimension values from the REGEX above would also
  be a part of the metric name.
* If true, the matched dimension values would not be a part of metric name.
* Default: true

[metric-schema:<unique_transforms_stanza_name>]
* Helps in transformation of index-time field extractions from a log events
  into a metrics data point with a required measurement fields.
* The other extracted fields from the log event become dimensions in the
  generated metrics data point.
* You must provide one of the following two settings:
  METRIC-SCHEMA-MEASURES-<unique_metric_name_prefix> or METRIC-SCHEMA-MEASURES. These
```

```
  settings are required and will inform which measurement indexed-time fields get
  created with key::value = metric_name:<metric_name>::<measurement>

METRIC-SCHEMA-MEASURES-<unique_metric_name_prefix> = (_ALLNUMS_ | (_NUMS_EXCEPT_ )? <field1>, <field2>,...
)
* Optional.
* <unique_metric_name_prefix> should match the value of a field extracted from
  the event.
* If this setting is exactly equal to _ALLNUMS_, the Splunk software treats
  all numeric fields as measures.
* If this setting starts with _NUMS_EXCEPT_, the Splunk software treats all
  numerical fields except those that match the given field names as  measures.
  * NOTE: a space is required between the '_NUMS_EXCEPT_' prefix and '<field1>'.
* Otherwise, the Splunk software treats all fields that are listed and which
  have a numerical value as measures.
* If the value of the 'metric_name' index-time extraction matches with the
  <unique_metric_name_prefix>, the Splunk platform:
  * Creates a metric with a new metric_name for each measure field where the
    metric_name value is the name of the field prefixed by the
    <unique_metric_name_prefix>.
  * Saves the corresponding numeric value for each measure field as '_value'
    within each metric.
* The Splunk platform saves the remaining index-time field extractions as
  dimensions in each of the created metrics.
* Use the wildcard character ("*") to match multiple similar <field>
  values in your event data. For example, say your event data contains the
  following measurement fields: 'current_size_kb', 'max_size_kb', and
  'min_size_kb'. You can set a <field> value of '*_size_kb' to include all
  three of those measurement fields in the field list without listing each one
  separately.
* Default: empty string

METRIC-SCHEMA-BLACKLIST-DIMS-<unique_metric_name_prefix> = <dimension_field1>,
<dimension_field2>,...
* Optional.
* This deny list configuration lets the Splunk platform omit unnecessary
  dimensions when it transforms event data to metrics data. You might set this
  up if some of the dimensions in your event data are high-cardinality and are
  unnecessary for your metrics.
* Use this configuration in conjunction with a corresponding
  METRIC-SCHEMA-MEASURES-<unique_metric_name_prefix> configuration.
* <unique_metric_name_prefix> should match the value of a field extracted from
  the log event.
* <dimension_field> should match the name of a field in the log event that is
  not extracted as a measure field in the corresponding METRIC-SCHEMA-
  MEASURES-<unique_metric_name_prefix> configuration.
* Use the wildcard character ("*") to match multiple similar <dimension_field>
  values in your event data. For example, say your event data contains the
  following dimensions: 'customer_id', 'employee_id', and 'consultant_id'. You
  can set a <dimension_name> value of '*_id' to include all three of those
  dimensions in the dimension field list without listing each one separately.
* The Splunk platform applies the following evaluation logic when you use the
  METRIC-SCHEMA-BLACKLIST-DIMS-<unique_metric_name_prefix> and the
  METRIC-SCHEMA-WHITELIST-DIMS-<unique_metric_name_prefix>
  configurations simultaneously in a stanza:
  * If a dimension is in the deny list (METRIC-SCHEMA-BLACKLIST-DIMS), it will
    not be present in the resulting metric data points, even if it also appears
    in the allow list (METRIC-SCHEMA-WHITELIST-DIMS).
  * If a dimension is not in the allow list, it will not be present in the
    resulting metric data points, even if it also does not appear in the
    deny list.
* Default: empty string
```

```
METRIC-SCHEMA-WHITELIST-DIMS-<unique_metric_name_prefix> = <dimension_field1>,
<dimension_field2>,...
* Optional.
* This allow list configuration allows the Splunk platform to include only a
  specified subset of dimensions when it transforms event data to metrics data.
  You might include an allow list in your log-to-metrics configuraton if many of
  the dimensions in your event data are high-cardinality and are unnecessary
  for your metrics.
* Use this configuration in conjunction with a corresponding
  METRIC-SCHEMA-MEASURES-<unique_metric_name_prefix> configuration.
* <unique_metric_name_prefix> should match the value of a field extracted from
  the log event.
* <dimension_field> should match the name of a field in the log event that is
  not extracted as a measure field in the corresponding METRIC-SCHEMA-
  MEASURES-<unique_metric_name_prefix> configuration.
* Use the wildcard character ("*") to match multiple similar <dimension_field>
  values in your event data. For example, say your event data contains the
  following dimensions: 'customer_id', 'employee_id', and 'consultant_id'. You
  can set a <dimension_name> value of '*_id' to include all three of those
  dimensions in the dimension field list without listing each one separately.
* The Splunk platform applies the following evaluation logic when you use the
  METRIC-SCHEMA-BLACKLIST-DIMS-<unique_metric_name_prefix> and the
  METRIC-SCHEMA-WHITELIST-DIMS-<unique_metric_name_prefix>
  configurations simultaneously in a stanza:
  * If a dimension is in the deny list (METRIC-SCHEMA-BLACKLIST-DIMS), it will
    not be present in the resulting metric data points, even if it also appears
    in the allow list (METRIC-SCHEMA-WHITELIST-DIMS).
  * If a dimension is not in the allow list, it will not be present in the
    resulting metric data points, even if it also does not appear in the
    deny list.
* When the allow list is empty, it behaves as if it contains all fields.
* Default: empty string

METRIC-SCHEMA-MEASURES = (_ALLNUMS_ | (_NUMS_EXCEPT_ )? <field1>, <field2>,... )
* Optional.
* This configuration has a lower precedence over METRIC-SCHEMA-MEASURES-<unique_metric_name_prefix>
  if event has a match for unique_metric_name_prefix
* When no prefix can be identified, this configuration is active
  to create a new metric for each measure field in the event data, as defined
  in the previous description for METRIC-SCHEMA-MEASURES-<unique_metric_name_prefix>
* The Splunk platform saves the remaining index-time field extractions as
  dimensions in each of the created metrics.
* Use the wildcard character ("*") to match multiple similar <field>
  values in your event data. For example, say your event data contains the
  following measurement fields: 'current_size_kb', 'max_size_kb', and
  'min_size_kb'. You can set a <field> value of '*_size_kb' to include all
  three of those measurement fields in the field list without listing each one
  separately.
* Default: empty string

METRIC-SCHEMA-BLACKLIST-DIMS = <dimension_field1>, <dimension_field2>,...
* Optional.
* This deny list configuration allows the Splunk platform to omit unnecessary
  dimensions when it transforms event data to metrics data. You might set this
  up if some of the dimensions in your event data are high-cardinality and are
  unnecessary for your metrics.
* Use this configuration in conjunction with a corresponding
  METRIC-SCHEMA-MEASURES configuration.
* <dimension_field> should match the name of a field in the log event that is
  not extracted as a <measure_field> in the corresponding METRIC-SCHEMA-
  MEASURES configuration.
```

```
* Use the wildcard character ("*") to match multiple similar <dimension_field>
  values in your event data. For example, say your event data contains the
  following dimensions: 'customer_id', 'employee_id', and 'consultant_id'. You
  can set a <dimension_name> value of '*_id' to include all three of those
  dimensions in the dimension field list without listing each one separately.
* The Splunk platform applies the following evaluation logic when you use the
  METRIC-SCHEMA-BLACKLIST-DIMS and the METRIC-SCHEMA-WHITELIST-DIMS
  configurations simultaneously in a stanza:
  * If a dimension is in the deny list (METRIC-SCHEMA-BLACKLIST-DIMS), it will
    not be present in the resulting metric data points, even if it also appears
    in the allow list (METRIC-SCHEMA-WHITELIST-DIMS).
  * If a dimension is not in the allow list, it will not be present in the
    resulting metric data points, even if it also does not appear in the
    deny list.
* Default: empty string

METRIC-SCHEMA-WHITELIST-DIMS = <dimension_field1>, <dimension_field2>,...
* Optional.
* This allow list configuration allows the Splunk platform to include only a
  specified subset of dimensions when it transforms event data to metrics data.
  You might include an allow list in your log-to-metrics configuraton if many of
  the dimensions in your event data are high-cardinality and are unnecessary
  for your metrics.
* Use this configuration in conjunction with a corresponding
  METRIC-SCHEMA-MEASURES configuration.
* <dimension_field> should match the name of a field in the log event that is
  not extracted as a <measure_field> in the corresponding METRIC-SCHEMA-
  MEASURES configuration.
* Use the wildcard character ("*") to match multiple similar <dimension_field>
  values in your event data. For example, say your event data contains the
  following dimensions: 'customer_id', 'employee_id', and 'consultant_id'. You
  can set a <dimension_name> value of '*_id' to include all three of those
  dimensions in the dimension field list without listing each one separately.
* The Splunk platform applies the following evaluation logic when you use the
  METRIC-SCHEMA-BLACKLIST-DIMS and the METRIC-SCHEMA-WHITELIST-DIMS
  configurations simultaneously in a stanza:
  * If a dimension is in the deny list (METRIC-SCHEMA-BLACKLIST-DIMS), it will
    not be present in the resulting metric data points, even if it also appears
    in the allow list (METRIC-SCHEMA-WHITELIST-DIMS).
  * If a dimension is not in the allow list, it will not be present in the
    resulting metric data points, even if it also does not appear in the
    deny list.
* Default: empty string
  * When the allow list is empty it behaves as if it contains all fields.
```

KEYS:

```
* NOTE: Keys are case-sensitive. Use the following keys exactly as they
        appear.

queue : Specify which queue to send the event to (can be nullQueue, indexQueue).
        * indexQueue is the usual destination for events going through the
          transform-handling processor.
        * nullQueue is a destination which causes the events to be
          dropped entirely.
_raw  : The raw text of the event.
_meta : A space-separated list of metadata for an event.
_time : The timestamp of the event, in seconds since 1/1/1970 UTC.

MetaData:Host       : The host associated with the event.
```

```
                      The value must be prefixed by "host::"

_MetaData:Index     : The index where the event should be stored.

MetaData:Source     : The source associated with the event.
                      The value must be prefixed by "source::"

MetaData:Sourcetype : The source type of the event.
                      The value must be prefixed by "sourcetype::"

_TCP_ROUTING        : Comma separated list of tcpout group names (from
                      outputs.conf)
                      Defaults to groups present in 'defaultGroup' for [tcpout].

_SYSLOG_ROUTING     : Comma separated list of syslog-stanza  names (from
                      outputs.conf)
                      Defaults to groups present in 'defaultGroup' for [syslog].

* NOTE: Any KEY (field name) prefixed by '_' is not indexed by Splunk software,   in general.

[accepted_keys]

<name> = <key>
* Modifies the list of valid SOURCE_KEY and DEST_KEY values. Splunk software
  checks the SOURCE_KEY and DEST_KEY values in your transforms against this
  list when it performs index-time field transformations.
* Add entries to [accepted_keys] to provide valid keys for specific
  environments, apps, or similar domains.
* The 'name' element disambiguates entries, similar to -class entries in
  props.conf.
* The 'name' element can be anything you choose, including a description of
  the purpose of the key.
* The entire stanza defaults to not being present, causing all keys not
  documented just above to be flagged.
* Default: not set

############################################################################
# Per transform rule metrics
#
# When enabled, the indexer collects and reports data on metrics events
# processed by each transform rule qualified by the 'prefix_filter'
# setting: the event count, the raw size, and where the events are routed.
# The data goes to the metric.log file.
############################################################################
[_ruleset:global_settings]
metrics.disabled = <boolean>
* Determines whether data for transform rule metrics is collected.
* Default: true

metrics.report_interval = <interval>
* Specifies how often to generate the per transform rule metrics logs.
* The interval can be specified as a string for seconds, minutes, hours, days.
  For example; 30s, 1m etc.
* It will be rounded to integer times of the interval value defined under
  the [metrics] stanza in limits.conf.
* Default: 30s

metrics.rule_filter = <string>
* Per transform rule metrics will be collected only for rule names that match
  this filter. In cases where a large number of transform rules are defined,
  this setting prevents metrics.log from being flooded with per transform rule
  metrics log entries.
```

```
* Wildcards (*) are supported. Multiple rules shall be seperated by commas,
  for example: abc*,*def,g*h*i.
* If set to the default, metrics data will be collected for all transform
  rules.
* Default: empty string
```

transforms.conf.example

```
#   Version 9.2.2
#
# This is an example transforms.conf.  Use this file to create regexes and
# rules for transforms.  Use this file in tandem with props.conf.
#
# To use one or more of these configurations, copy the configuration block
# into transforms.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# Note: These are examples.  Replace the values with your own customizations.


# Indexed field:

[netscreen-error]
REGEX =  device_id=\[w+\](?<err_code>[^:]+)
FORMAT = err_code::$1
WRITE_META = true


# Override host:

[hostoverride]
DEST_KEY = MetaData:Host
REGEX = \s(\w*)$
FORMAT = host::$1


# Extracted fields:

[netscreen-error-field]
REGEX = device_id=\[w+\](?<err_code>[^:]+)
FORMAT = err_code::$1

# Index-time evaluations:

[discard-long-lines]
INGEST_EVAL = queue=if(length(_raw) > 500, "nullQueue", "indexQueue")

[split-into-sixteen-indexes-for-no-good-reason]
INGEST_EVAL = index="split_" . substr(md5(_raw),1,1)

[add-two-numeric-fields]
INGEST_EVAL = loglen_raw=ln(length(_raw)), loglen_src=ln(length(source))

# In this example the Splunk platform only creates the new index-time field if
```

```
# the hostname has a dot in it; assigning null() to a new field is a no-op:

[add-hostdomain-field]
INGEST_EVAL = hostdomain=if(host LIKE "%.%", replace(host,"^[^\\.]+\\.",""), null())

# Static lookup table

[mylookuptable]
filename = mytable.csv

# One-to-one lookup guarantees that the Splunk platform outputs a single
# lookup value for each input value. When no match exists, the Splunk platform
# uses the value for "default_match", which by default is nothing.

[mylook]
filename = mytable.csv
max_matches = 1
min_matches = 1
default_match =

# Lookup and filter results:

[myfilteredlookup]
filename = mytable.csv
filter = id<500 AND color="red"

# external command lookup table:

[myexternaltable]
external_cmd = testadapter.py blah
fields_list = foo bar

# Temporal based static lookup table:

[staticwtime]
filename = mytable.csv
time_field = timestamp
time_format = %d/%m/%y %H:%M:%S

# Mask sensitive data:

[session-anonymizer]
REGEX = (?m)^(.*)SessionId=\w+(\w{4}[&"].*)$
FORMAT = $1SessionId=########$2
DEST_KEY = _raw

# Route to an alternate index:

[AppRedirect]
REGEX = (Application)
DEST_KEY = _MetaData:Index
FORMAT = Verbose

# Extract comma-delimited values into fields:
# This example assigns extracted values that do not have file names
# from _raw to field1, field2 and field3, in the order that the
# fields are extracted.
#If the Splunk platform extracts more than three values that do not
# have field names, then the Splunk platform ignores those values.

[extract_csv]
DELIMS = ","
```

```
FIELDS = "field1", "field2", "field3"

# This example extracts key-value pairs which are separated by '|'
# while the key is delimited from value by '='

[pipe_eq]
DELIMS = "|", "="

# This example extracts key-value pairs which are separated by '|' or
# ';', while the key is delimited from value by '=' or ':'

[multiple_delims]
DELIMS = "|;", "=:"


###### BASIC MODULAR REGULAR EXPRESSIONS DEFINITION START ###########
# When you add a new basic modular regex you must add a comment that
# lists the fields that it extracts as named capturing groups.
# If there are no field names, note the placeholders
# for the group name as: Extracts: field1, field2....

[all_lazy]
REGEX = .*?

[all]
REGEX = .*

[nspaces]
# Matches one or more NON space characters:
REGEX = \S+

[alphas]
# Matches a string containing only letters a-zA-Z:
REGEX = [a-zA-Z]+

[alnums]
# Matches a string containing letters + digits:
REGEX = [a-zA-Z0-9]+

[qstring]
# Matches a quoted "string" and extracts an unnamed variable
# Name MUST be provided as: [[qstring:name]]
# Extracts: empty-name-group (needs name)
REGEX = "(?<>[^"]*+)"

[sbstring]
# Matches a string enclosed in [] and extracts an unnamed variable
# Name must be provided as: [[sbstring:name]]
# Extracts: empty-name-group (needs name)
REGEX = \[(?<>[^\]]*+)\]

[digits]
REGEX = \d+

[int]
# Matches an integer or a hex number:
REGEX = 0x[a-fA-F0-9]+|\d+

[float]
# Matches a float (or an int):
REGEX = \d*\.\d+|[[int]]
```

```
[octet]
# Matches only numbers from 0-255 (one octet in an ip):
REGEX = (?:2(?:5[0-5]|[0-4][0-9])|[0-1][0-9][0-9]|[0-9][0-9]?)

[ipv4]
# Matches a valid IPv4 optionally followed by :port_num. The octets in the IP
# are also be validated in the 0-255 range.
# Extracts: ip, port
REGEX = (?<ip>[[octet]](?:\.[[octet]]){3})(?::[[int:port]])?

[simple_url]
# Matches a url of the form proto://domain.tld/uri
# Extracts: url, domain
REGEX = (?<url>\w++://(?<domain>[a-zA-Z0-9\-.:]++)(?:/[^\s"]*)?)

[url]
# Matches a url in the form of: proto://domain.tld/uri
# Extracts: url, proto, domain, uri
REGEX = (?<url>[[alphas:proto]]://(?<domain>[a-zA-Z0-9\-.:]++)(?<uri>/[^\s"]*)?)

[simple_uri]
# Matches a uri in the form of: /path/to/resource?query
# Extracts: uri, uri_path, uri_query
REGEX = (?<uri>(?<uri_path>[^\s\?"]++)(?:\\?(?<uri_query>[^\s"]+))?)

[uri]
# uri  = path optionally followed by query [/this/path/file.js?query=part&other=var]
# path = root part followed by file        [/root/part/file.part]
# Extracts: uri, uri_path, uri_root, uri_file, uri_query, uri_domain (optional if in proxy mode)
REGEX = (?<uri>(?:\w++://(?<uri_domain>[^/\s]++))?(?<uri_path>(?<uri_root>/+(?:[^\s\?;=/]*+/+)*)(?<uri
_file>[^\s\?;=?/]*+))(?:\?(?<uri_query>[^\s"]+))?)

[hide-ip-address]
# When you make a clone of an event with the sourcetype masked_ip_address, the clone's
# text is changed to mask the IP address.
# The cloned event is further processed by index-time transforms and
# SEDCMD expressions according to its new sourcetype.
# In most scenarios an additional transform directs the
# masked_ip_address event to a different index than the original data.
REGEX = ^(.*?)src=\d+\.\d+\.\d+\.\d+(.*)$
FORMAT = $1src=XXXXX$2
DEST_KEY = _raw
CLONE_SOURCETYPE = masked_ip_addresses

# Set repeat_match to true to repeatedly match the regex in the data.
# When repeat_match is set to true, regex is added as indexed
# fields: a, b, c, d, e, etc. For example: 1483382050 a=1 b=2 c=3 d=4 e=5
# If repeat_match is not set, the match stops at a=1.
[repeat_regex]
REGEX = ([a-z])=(\d+)
FORMAT = $1::$2
REPEAT_MATCH = true
WRITE_META = true

###### BASIC MODULAR REGULAR EXPRESSIONS DEFINITION END ############

# Statsd dimensions extraction:

# In most cases the Splunk platform needs only one regex to run per
# sourcetype. By default the Splunk platform would look for the sourcetype
# name in transforms.conf. There there is no need to provide
# the STATSD-DIM-TRANSFORMS setting in props.conf.
```

```
# For example, these two stanzas would extract dimensions as ipv4=10.2.3.4
# and os=windows from statsd data=mem.percent.used.10.2.3.4.windows:33|g
[statsd-dims:regex_stanza1]
REGEX = (?<ipv4>\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})
REMOVE_DIMS_FROM_METRIC_NAME = true

[statsd-dims:regex_stanza2]
REGEX = \S+\.(?<os>\w+):
REMOVE_DIMS_FROM_METRIC_NAME = true



[statsd-dims:metric_sourcetype_name]
# In this example, we extract both ipv4 and os dimension using a single regex:
REGEX = (?<ipv4>\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})\.(?<os>\w+):
REMOVE_DIMS_FROM_METRIC_NAME = true


# In this metrics example, we start with this log line:
#
# 01-26-2018 07:49:49.030 -0800 INFO  Metrics - group=queue, name=aggqueue, max_size_kb=1024,
current_size_kb=1,
# current_size=3, largest_size=49, smallest_size=0, dc_latitude=37.3187706, dc_longitude=-121.9515042
#
# The following stanza converts that single event into multiple metrics at
# index-time. It deny lists the "dc_latitude" and "dc_longitude" dimensions,
# which means they are omitted from the generated metric data points. It also
# allow lists the "name" and "dc_latitude" dimensions, which means that those
# dimensions potentially are the only dimensions that appear in the
# generated metric data points.
# When a log-to-metrics configuration simultaneously includes allow list and
# deny list dimensions, the Splunk platform includes the dimensions that
# appear in the allow list and also do not appear in the deny list
# for the generated metric data points. For example, "dc_latitude" appears in
# the allow list, but also in the deny list, so it is not included in the generated
# metric data points. The metric data points generated by this configuration
# have "name" as their sole dimension.
[metric-schema:logtometrics]
METRIC-SCHEMA-MEASURES-queue = max_size_kb,current_size_kb,current_size,largest_size,smallest_size
METRIC-SCHEMA-BLACKLIST-DIMS-queue = dc_latitude,dc_longitude
METRIC-SCHEMA-WHITELIST-DIMS-queue = name,dc_latitude

# Here are the metrics generated by that stanza:
# {'metric_name' : 'queue.max_size_kb',     '_value' : 1024, 'name': 'aggqueue'},
# {'metric_name' : 'queue.current_size_kb', '_value' : 1,    'name': 'aggqueue'},
# {'metric_name' : 'queue.current_size',    '_value' : 3,    'name': 'aggqueue'},
# {'metric_name' : 'queue.largest_size',    '_value' : 49,   'name': 'aggqueue'},
# {'metric_name' : 'queue.smallest_size',   '_value' : 0,    'name': 'aggqueue'}

# You can use wildcard characters ('*') in METRIC-SCHEMA configurations. In
# the preceding example, '*_size' matches 'current_size', 'largest_size', and
# 'smallest_size'. The following configuration uses a wildcard to include all
# three of those fields without individually listing each one.
# METRIC-SCHEMA-MEASURES-queue = max_size_kb,current_size_kb,*_size

# In the sample log above, group=queue represents the unique metric name prefix. Hence, it needs to be
# formatted and saved as metric_name::queue for Splunk to identify queue as a metric name prefix.
[extract_group]
REGEX = group=(\w+)
FORMAT = metric_name::$1
WRITE_META = true
```

```
[extract_name]
REGEX = name=(\w+)
FORMAT = name::$1
WRITE_META = true

[extract_max_size_kb]
REGEX = max_size_kb=(\w+)
FORMAT = max_size_kb::$1
WRITE_META = true

[extract_current_size_kb]
REGEX = current_size_kb=(\w+)
FORMAT = current_size_kb::$1
WRITE_META = true

[extract_current_size]
REGEX = max_size_kb=(\w+)
FORMAT = max_size_kb::$1
WRITE_META = true

[extract_largest_size]
REGEX = largest_size=(\w+)
FORMAT = largest_size::$1
WRITE_META = true

[extract_smallest_size]
REGEX = smallest_size=(\w+)
FORMAT = smallest_size::$1
WRITE_META = true
```

ui-prefs.conf

The following are the spec and example files for `ui-prefs.conf`.

ui-prefs.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure the ui for a view.
#
# There is a ui-prefs.conf in $SPLUNK_HOME/etc/system/default directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name ui-prefs.conf in
# the $SPLUNK_HOME/etc/apps/<app_name>/local/ directory. Then add the specific
# settings that you want to customize to the local configuration file.
# For examples, see ui-prefs.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
```

```
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file takes precedence.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

[<stanza name>]
* The name of the xml view file

dispatch.earliest_time =
dispatch.latest_time =
```

Preference options

```
display.prefs.autoOpenSearchAssistant = 0 | 1
display.prefs.timeline.height = <string>
display.prefs.timeline.minimized = 0 | 1
display.prefs.timeline.minimalMode = 0 | 1
display.prefs.aclFilter = [none|app|owner]
display.prefs.appFilter = <string>
display.prefs.listMode = [tiles|table]
display.prefs.searchContext = <string>
display.prefs.events.count = [10|20|50]
display.prefs.statistics.count = [10|20|50|100]
display.prefs.fieldCoverage = [0|.01|.50|.90|1]
display.prefs.enableMetaData = 0 | 1
display.prefs.showDataSummary = 0 | 1
display.prefs.customSampleRatio = <int>
display.prefs.showSPL = 0 | 1
display.prefs.livetail = 0 | 1

# Count per page for listing pages
countPerPage = [10|20|50]
```

Display Formatting Options

```
# General options
display.general.enablePreview = 0 | 1

# Event options
display.events.fields = <string>
display.events.type = [raw|list|table]
```

```
display.events.rowNumbers = 0 | 1
display.events.maxLines = [0|5|10|20|50|100|200]
display.events.raw.drilldown = [inner|outer|full|none]
display.events.list.drilldown = [inner|outer|full|none]
display.events.list.wrap = 0 | 1
display.events.table.drilldown = 0 | 1
display.events.table.wrap = 0 | 1

# Statistics options
display.statistics.rowNumbers = 0 | 1
display.statistics.wrap = 0 | 1
display.statistics.drilldown = [row|cell|none]

# Visualization options
display.visualizations.type = [charting|singlevalue]
display.visualizations.custom.type = <string>
display.visualizations.chartHeight = <int>
display.visualizations.charting.chart =
[line|area|column|bar|pie|scatter|radialGauge|fillerGauge|markerGauge]
display.visualizations.charting.chart.style = [minimal|shiny]
display.visualizations.charting.legend.labelStyle.overflowMode = [ellipsisEnd|ellipsisMiddle|ellipsisStart]

# Patterns options
display.page.search.patterns.sensitivity = <float>

# Page options
display.page.search.mode = [fast|smart|verbose]
display.page.search.timeline.format = [hidden|compact|full]
display.page.search.timeline.scale = [linear|log]
display.page.search.showFields = 0 | 1
display.page.home.showGettingStarted = 0 | 1
display.page.search.searchHistoryTimeFilter = [0|@d|-7d@d|-30d@d]
display.page.search.searchHistoryCount = [10|20|50]
```

ui-prefs.conf.example

```
#   Version 9.2.2
#
# This file contains example of ui preferences for a view.
#
# To use one or more of these configurations, copy the configuration block into
# ui-prefs.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to
# enable configurations.
#
# To learn more about configuration files (including precedence) please see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# The following ui preferences will default timerange picker on the search page
# from All time to Today We will store this ui-prefs.conf in
# $SPLUNK_HOME/etc/apps/search/local/ to only update search view of search app.
[search]
dispatch.earliest_time = @d
dispatch.latest_time = now
```

ui-tour.conf

The following are the spec and example files for `ui-tour.conf`.

ui-tour.conf.spec

```
#   Version 9.2.2
#
# This file contains the available product tours for Splunk onboarding.
#
# There is a default ui-tour.conf in $SPLUNK_HOME/etc/system/default.
# To create custom tours, place a ui-tour.conf in
# $SPLUNK_HOME/etc/system/local/. To create custom tours for an app, place
# ui-tour.conf in $SPLUNK_HOME/etc/apps/<app_name>/local/.
#
# To learn more about configuration files (including precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * This is not a typical conf file for configurations. It is used to set/create
#     tours to demonstrate product functionality to users.
#   * If an attribute is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.

[<stanza name>]
* The name of the UI tour.

useTour = <string>
* Used to redirect this tour to another when called by Splunk.
* Optional.

nextTour = <string>
* Determines what tour to start when the current tour is finished.
* Optional.

intro = <string>
* A custom string used in a modal to describe which tour is about to be taken.
* Optional.

type = image|interactive
* Determines the type of tour.
* Required.
* If set to "image", the tour is a simple image tour where the user clicks through
  a series of screenshots or images.
* If set to "interactive", the user participates in an interactive UI tour.

label = <string>
* The identifying name for the tour used in the tour creation app.
* Required only if the tour is being linked to another tour using the 'nextTour' setting.

tourPage = <string>
```

```
* The Splunk view the tour is associated with.
* Required only if the tour is being linked to another tour using the 'nextTour' setting.

managerPage = <boolean>
* Used to signifiy that the 'tourPage' is a manager page. This changes the URL of
  when the 'tourPage' is rendered from "/app/{app}/{view}" to "/manager/{app}/{view}".
* Optional

viewed = <boolean>
* Whether the tour has been viewed by a user.
* Set by Splunk.

skipText = <string>
* The string for the skip button.
* Optional.
* This setting applies to both interactive and image tours.
* Default: Skip tour

doneText = <string>
* The string for the button at the end of a tour.
* Optional.
* This setting applies to both interactive and image tours.
* Default: Try it now

doneURL = <string>
* A Splunk URL that redirects the user once the tour is over and they click a
  link or button to exit.
* Optional.
* Helpful to use with the 'doneText' setting to specify a starting location for the user
  after they take the tour.
* The Splunk link is formed after the localization portion of the full URL. For example, if the link
* is localhost:8000/en-US/app/search/reports, the doneURL will be "app/search/reports".

forceTour = <boolean>
* Used with auto tours to force users to take the tour and not be able to skip.
* Optional
```

For image-based tours

```
# You can list as many images with captions as you want. Each new image is created by
# incrementing the number.

imageName<int> = <string>
* The name of the image file.
* For example, 'example.png'.
* Required but optional only after the first is set.

imageCaption<int> = <string>
* The caption string for the corresponding image.
* Optional.

imgPath = <string>
* The subdirectory relative to Splunk's 'img' directory in which users put the images.
  This will be appended to the URL for image access and not make a server request within Splunk.
  Ex) If the user puts images in a subdirectory 'foo': imgPath = /foo.
  Ex) If within an app, imgPath = /foo will point to the app's img path of
      appserver/static/img/foo
* Required only if images are not in the main 'img' directory.

context = <system|<specific app name>>
```

```
* String consisting of either 'system' or the app name where the tour images are to be stored.
* Required.
* If set to "system", it reverts to Splunk's native img path.
```

For interactive tours

```
# You can list as many steps with captions as you want. Each new step is created by
# incrementing the number.

urlData = <string>
* The string of any querystring variables used with the 'tourPage' setting
  to create the full URL executing this tour.
* Optional.
* Don't add "?" to the beginning of this string.

stepText<int> = <string>
* The string used in a specified step to describe the UI being showcased.
* Required but optional only after the first is set.

stepElement<int> = <selector>
* The UI selector used for highlighting the DOM element for the corresponding step.
* Optional.

stepPosition<int> = <bottom|right|left|top>
* String that sets the position of the tooltip for the corresponding step.
* Optional.

stepClickEvent<int> = <click|mousedown|mouseup>
* Sets a specific click event for an element for the corresponding step.
* Optional.

stepClickElement<int> = <string>
* The UI selector used for a DOM element used in conjunction with `stepClickEvent<int>`.
* Optional.
```

ui-tour.conf.example

```
#   Version 9.2.2
#
# This file contains the tours available for Splunk Onboarding
#
# To update tours, copy the configuration block into
# ui-tour.conf in $SPLUNK_HOME/etc/system/local/. Restart the Splunk software to
# see the changes.
#
# To learn more about configuration files (including precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#

# Image Tour
[tour-name]
type = image
imageName1 = TourStep1.png
imageCaption1 = This is the first caption
imageName2 = TourStep2.png
imageCaption2 = This is the second caption
```

```
imgPath = /testtour
context = system
doneText = Continue to Tour Page
doneURL = app/toursapp/home

# Interactive Tour
[test-interactive-tour]
type = interactive
tourPage = reports
urlData = data=foo&moredata=bar
label = Interactive Tour Test
stepText1 = Welcome to this test tour
stepText2 = This is the first step in the tour
stepElement2 = .test-selector
stepText3 = This is the second step in the tour
stepElement3 = .test-selector
stepClickEvent3 = mousedown
stepClickElement3 = .test-click-element
forceTour = 1
```

user-prefs.conf

The following are the spec and example files for `user-prefs.conf`.

user-prefs.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure on a per-user basis for use by the Splunk Web UI.
#
# There is a user-prefs.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name user-prefs.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see user-prefs.conf.example. You must restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# NOTES:
#
# Settings in this file are requested with user and application scope of the
# relevant user, and the user-prefs app.
#
# Additionally, settings by the same name which are available in the roles
# the user belongs to will be used at lower precedence.
#
```

```
# This means interactive setting of these values will cause the values to be
# updated in
# $SPLUNK_HOME/etc/users/<username>/user-prefs/local/user-prefs.conf where
# <username> is the username for the user altering their preferences.
#
# It also means that values in another app will never be used unless they
# are exported globally (to system scope) or to the user-prefs app.
#
# In practice, providing values in other apps isn't very interesting, since
# values from the authorize.conf file 'roles' settings are more typically sensible
# ways to defaults for values in user-prefs.
```

[general]

```
default_namespace = <app name>
* Specifies the app that the user will see initially on login to the
  Splunk Web User Interface.
* This uses the "short name" of the app, such as launcher, or search,
  which is synonymous with the app directory name.
* Default: launcher (via the default authorize.conf file)

tz = <timezone>
* Specifies the per-user timezone to use.
* If unset, the timezone of the Splunk Server or Search Head is used.
* Only canonical timezone names such as America/Los_Angeles should be
  used (for best results use the Splunk UI).
* No default.

lang = <string>
* Specifies the per-user language preference for non-web ui operations, where
  multiple tags are separated by commas.
* If unset, English "en-US" is used when required.
* Only tags used in the "Accept-Language" HTTP header are allowed, such as
  "en-US" or "fr-FR".
* Fuzzy matching is supported, where "en" will match "en-US".
* Optional quality settings are supported, such as "en-US,en;q=0.8,fr;q=0.6"
* No default.


install_source_checksum = <string>
* Records a checksum of the tarball from which a given set of private user
  configurations was installed.
* Analogous to <install_source_checksum> in the app.conf file.

search_syntax_highlighting = [default-system-theme|light|dark|black-white]
* Highlights different parts of a search string with different colors.
* Dashboards ignore this setting.
* default-system-theme = Inherits the default system theme if the current app
  supports theming.
* light = White background with dark colored text.
* dark = Black background with light colored text.
* black-white = White background with black text.
* Default: default-system-theme

search_use_advanced_editor = <boolean>
* Specifies whether the search bar is run using the advanced editor or in just plain text.
* If set to false, 'search_auto_format' and 'search_line_numbers' will be "false" and 'search_assistant'
cannot be "compact".
* Default: true

search_assistant = [full|compact|none]
```

```
* Specifies the type of search assistant to use when constructing a search.
* Default: compact

theme = [default_system_theme|light|dark]
* Specifies the preferred theme for the user.
* Not all apps used with the Splunk platform support the dark theme. If
  supported by the app, the theme is applied to the UI by Splunk Web.
  Otherwise, Splunk Web applies the default system theme.
* Default: default_system_theme

search_auto_format = <boolean>
* Specifies if auto-format is enabled in the search input.
* Default: false

search_line_numbers = <boolean>
* Display the line numbers with the search.
* Default: false

dismissedInstrumentationOptInVersion = <integer>
* Set by splunk_instrumentation app to its current value of optInVersion when the opt-in modal is dismissed.

hideInstrumentationOptInModal = <boolean>
* Set to 1 by splunk_instrumentation app when the opt-in modal is dismissed.
```

[default]

```
# Additional settings exist, but are entirely UI managed.
<setting> = <value>
```

[general_default]

```
default_earliest_time = <string>
default_latest_time = <string>
* Sets the global default time range across all apps, users, and roles on the search page.

notification_python_3_impact = <string>
* Flag to enable, disable, or snooze the Python 3 impact notification dialog.
* Default: true

notification_python_2_removal = <string>
* Flag to enable, disable, or snooze the Python 2 removal notification.
* Default: false

notification_noah_upgrade = <string>
* Flag to enable, disable, or snooze the Noah notification dialog.
* Default: true
```

[role_<name>]

```
<name> = <value>
```

user-prefs.conf.example

```
#   Version 9.2.2
```

```
#
# This is an example user-prefs.conf.  Use this file to configure settings
# on a per-user basis for use by the Splunk Web UI.
#
# To use one or more of these configurations, copy the configuration block
# into user-prefs.conf in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# Note: These are examples.  Replace the values with your own
# customizations.


# EXAMPLE: Setting the default timezone to GMT for all Power and User role
# members, and setting a different language preference for each.

[role_power]
tz = GMT
lang = en-US

[role_user]
tz = GMT
lang = fr-FR,fr-CA;q=0
```

user-seed.conf

The following are the spec and example files for `user-seed.conf`.

user-seed.conf.spec

```
#   Version 9.2.2
#
# Specification for user-seed.conf.  Allows configuration of Splunk's
# initial username and password.  Currently, only one user can be configured
# with user-seed.conf.
#
# Specification for user-seed.conf.  Allows configuration of Splunk's initial username and password.
# Currently, only one user can be configured with user-seed.conf.
#
# To set the default username and password, place user-seed.conf in
# $SPLUNK_HOME/etc/system/local. You must restart Splunk to enable configurations.
# If the $SPLUNK_HOME/etc/passwd file is present, the settings in this file (user-seed.conf) are not used.
#
# Use HASHED_PASSWORD for a more secure installation. To hash a clear-text password,
# use the 'splunk hash-passwd' command then copy the output to this file.
#
# If a clear text password is set (not recommended) and last character is '\', it should
# be followed by a space for value to be read correctly. Password does not include extra
# space at the end, it is required to ignore the special meaning of backslash in conf file.
#
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
# To learn more about configuration files (including precedence) please see the documentation
# located at http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

[user_info]

```
* Default is Admin.

USERNAME = <string>
          * Username you want to associate with a password.
          * Default is Admin.

PASSWORD = <password>
          * Password you wish to set for that user.
          * Password must meet complexity requirements.

HASHED_PASSWORD = <password hash>
          * Password hash you wish to set for that user.
```

user-seed.conf.example

```
#   Version 9.2.2
#
# This is an example user-seed.conf.  Use this file to create an initial login.
#
# NOTE: When starting Splunk for first time, hash of password is stored in
# $SPLUNK_HOME/etc/system/local/user-seed.conf and password file is seeded
# with this hash. This file can also be used to set default username and
# password, if $SPLUNK_HOME/etc/passwd is not present. If the $SPLUNK_HOME/etc/passwd
# file is present, the settings in this file (user-seed.conf)
# are not used.
#
# To use this configuration, copy the configuration block into user-seed.conf
# in $SPLUNK_HOME/etc/system/local/. You must restart Splunk to enable configurations.
#
# To learn more about configuration files (including precedence) please see the documentation
# located at http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


[user_info]
USERNAME = admin
HASHED_PASSWORD =
$6$TOs.jXjSRTCsfPsw$2St.t9lH9fpXd9mCEmCizWbb67gMFfBIJU37QF8wsHKSGud1QNMCuUdWkD8IFSgCZr5.W6zkjmNACGhGafQZj1
```

viewstates.conf

The following are the spec and example files for `viewstates.conf`.

viewstates.conf.spec

```
#   Version 9.2.2
#
# This file explains how to format viewstates.
#
# To use this configuration, copy the configuration block into
# viewstates.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see
```

```
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top
#    of the file.
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, attributes are combined. In the case of
#    multiple definitions of the same attribute, the last definition in the
#    file wins.
#  * If an attribute is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.
```

[<view_name>:<viewstate_id>]

```
* Auto-generated persistence stanza label that corresponds to UI views
* The <view_name> is the URI name (not label) of the view to persist
* if <view_name> = "*", then this viewstate is considered to be 'global'
* The <viewstate_id> is the unique identifier assigned to this set of
  parameters
* <viewstate_id> = '_current' is a reserved name for normal view
  'sticky state'
* <viewstate_id> = '_empty' is a reserved name for no persistence,
  i.e., all defaults

<module_id>.<setting_name> = <string>
* The <module_id> is the runtime id of the UI module requesting persistence
* The <setting_name> is the setting designated by <module_id> to persist
```

viewstates.conf.example

```
#   Version 9.2.2
#
# This is an example viewstates.conf.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

[charting:g3b5fa7l]
ChartTypeFormatter_0_7_0.default = area
Count_0_6_0.count = 10
LegendFormatter_0_13_0.default = right
LineMarkerFormatter_0_10_0.default = false
NullValueFormatter_0_12_0.default = gaps

[*:g3jck9ey]
Count_0_7_1.count = 20
DataOverlay_0_12_0.dataOverlayMode = none
DataOverlay_1_13_0.dataOverlayMode = none
FieldPicker_0_6_1.fields = host sourcetype source date_hour date_mday date_minute date_month
FieldPicker_0_6_1.sidebarDisplay = True
```

```
FlashTimeline_0_5_0.annotationSearch = search index=twink
FlashTimeline_0_5_0.enableAnnotations = true
FlashTimeline_0_5_0.minimized = false
MaxLines_0_13_0.maxLines = 10
RowNumbers_0_12_0.displayRowNumbers = true
RowNumbers_1_11_0.displayRowNumbers = true
RowNumbers_2_12_0.displayRowNumbers = true
Segmentation_0_14_0.segmentation = full
SoftWrap_0_11_0.enable = true

[dashboard:_current]
TimeRangePicker_0_1_0.selected = All time
```

visualizations.conf

The following are the spec and example files for `visualizations.conf`.

visualizations.conf.spec

```
#   Version 9.2.2
#
# This file contains definitions for visualizations an app makes available
# to the system. If you want your app to share visualizations with the system,
# include a visualizations.conf in $SPLUNK_HOME/etc/apps/<appname>/default
# Within the file, include one stanza for each visualization to be shared.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

#*******
# The following attribute/value pairs are possible for stanzas in visualizations.conf:
#*******
```

[<stanza name>]

```
* Create a unique stanza name for each visualization that matches the visualization's name.
* Follow the stanza name with any number of the following attribute/value
  pairs.
* If you don't specify an attribute, Splunk uses the default.

disabled = <boolean>
* Disable the visualization by setting to true.
* Optional.
* If set to true, the visualization is not available anywhere in Splunk
* Default: false.

allow_user_selection = <boolean>
* Whether the visualization is available for users to select.
* Optional.
* Default: true

label = <string>
* The human-readable label or title of the visualization.
* Required.
* The label is used in dropdowns and lists as the name of the visualization.
* Default: <app_name>.<viz_name>
```

```
description = <string>
* A short description that appears in the visualizations picker.
* Required.
* Default: ""

search_fragment = <string>
* An example part of a search that formats the data correctly for the visualization.
* Required.
* Typically the last pipe or pipes in a search query.
* Default: ""

default_height = <integer>
* The default height of the visualization, in pixels.
* Optional.
* Default: 250

default_width = <integer>
* The default width of the visualization, in pixels
* Optional.
* Default: 250

min_height = <integer>
* The minimum height the visualizations can be rendered in, in pixels.
* Optional.
* Default: 50

min_width = <integer>
* The minimum width the visualizations can be rendered in, in pixels.
* Optional.
* Default: 50

max_height = <integer>
* The maximum height the visualizations can be rendered in, in pixels.
* Optional.
* Default: unbounded

max_width = <integer>
* The maximum width the visualizations can be rendered in, in pixels.
* Optional.
* Default: unbounded.

trellis_default_height = <integer>
* The default height of the visualization if using trellis layout.
* Default: 400

trellis_min_widths = <string>
* The minimum width of a visualization if using trellis layout.
* Default: undefined

trellis_per_row = <string>
* The number of trellises per row.
* Default: undefined

# The following settings define data sources supported by the visualization and their initial fetch
parameters for search results data:

data_sources = <comma-separated list>
* A list of data source types supported by the visualization.
* The visualization system currently provides the following types of data sources:
* - primary: Main data source driving the visualization.
* - annotation: Additional data source for time series visualizations to show discrete event annotation on
the time axis.
```

```
* Default: primary

data_sources.<data-source-type>.params.output_mode = [json_rows|json_cols|json]
* The data format that the visualization expects. Must be one of the following:
  - "json_rows": corresponds to SplunkVisualizationBase.ROW_MAJOR_OUTPUT_MODE
  - "json_cols": corresponds to SplunkVisualizationBase.COLUMN_MAJOR_OUTPUT_MODE
  - "json": corresponds to SplunkVisualizationBase.RAW_OUTPUT_MODE
* Optional.
* Requires the javascript implementation to supply initial data parameters.
* Default: undefined

data_sources.<data-source-type>.params.count = <integer>
* How many rows of results to request
* Optional.
* Default: 1000

data_sources.<data-source-type>.params.offset = <integer>
* The index of the first requested result row.
* Optional.
* Default: 0

data_sources.<data-source-type>.params.sort_key = <string>
* The field name to sort the results by.
* Optional.

data_sources.<data-source-type>.params.sort_direction = [asc|desc]
* The direction of the sort:
  - asc: Sort in ascending order
  - desc: Sort in descending order
* Optional.
* Default: desc

data_sources.<data-source-type>.params.search = <string>
* A post-processing search to apply to generate the results.
* Optional.
* There is no default.

data_sources.<data-source-type>.mapping_filter = <boolean>

data_sources.<data-source-type>.mapping_filter.center = <string>

data_sources.<data-source-type>.mapping_filter.zoom = <string>

supports_trellis = <boolean>
* Whether trellis layout is available for this visualization.
* Optional.
* Default: false

supports_drilldown = <boolean>
* Whether the visualization supports drilldown.
* Optional.
* A drilldown is a responsive actions triggered when users click on the visualization.
* Default: false

supports_export = <boolean>
* Whether the visualization supports being exported as a PDF.
* Optional.
* This setting has no effect in third-party visualizations.
* Default: false

# Internal settings for bundled visualizations. They are ignored for third party visualizations.
core.type = <string>
```

```
core.viz_type = <string>
core.charting_type = <string>
core.mapping_type = <string>
core.order = <int>
core.icon = <string>
core.preview_image = <string>
core.recommend_for = <string>
core.height_attribute = <string>
```

visualizations.conf.example

```
No example
```

web.conf

The following are the spec and example files for `web.conf`.

web.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attributes and values you can use to configure
# the Splunk Web interface.
#
# There is a web.conf in $SPLUNK_HOME/etc/system/default/.  To set custom
# configurations, place a web.conf in $SPLUNK_HOME/etc/system/local/.  For
# examples, see web.conf.example.  You must restart Splunk software to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles


[settings]
* Set general Splunk Web configuration options under this stanza name.
* Follow this stanza name with any number of the following setting/value
  pairs.
* If you do not specify an entry for each setting, Splunk Web uses the
  default value.

startwebserver = [0 | 1]
* Set whether or not to start Splunk Web.
* 0 disables Splunk Web, 1 enables it.
* Default: 1

httpport = <positive integer>
* The TCP port on which Splunk Web listens for incoming connections.
* Must be present for Splunk Web to start.
* If omitted or 0 the server will NOT start an http listener.
* If using SSL, set to the HTTPS port number.
* Default: 8000

mgmtHostPort = <string>
* The host port of the splunkd process.
* The IP address and host port where Splunk Web looks for the splunkd process.
```

```
* The port listens on all avalible host IP addresses (0.0.0.0)
* Don't include "http[s]://" when specifying this setting. Only
  include the IP address and port.
* Default (on universal forwarders): localhost:8089
* Default (on all other Splunk platform instance types): 0.0.0.0:8089

appServerPorts = <positive integer>[, <positive integer>, <positive integer> ...]
* Port number(s) for the python-based application server to listen on.
  This port is bound only on the loopback interface -- it is not
  exposed to the network at large.
* Generally, you should only set one port number here. For most
  deployments a single application server won't be a performance
  bottleneck. However you can provide a comma-separated list of
  port numbers here and splunkd will start a load-balanced
  application server on each one.
* At one time, setting this to zero indicated that the web service
  should be run in a legacy mode as a separate service, but as of
  Splunk 8.0 this is no longer supported.
* Default: 8065

splunkdConnectionTimeout = <integer>
* The amount of time, in seconds, to wait before timing out when communicating with
  splunkd.
* Must be at least 30.
* Values smaller than 30 will be ignored, resulting in the use of the
  default value
* Default: 30

enableSplunkWebClientNetloc = <boolean>
* Control if the Splunk Web client can override the client network location.
* Default: false

enableSplunkWebSSL = <boolean>
* Toggle between http or https.
* Set to true to enable https and SSL.
* Default: false

privKeyPath = <path>
* The path to the file containing the web server SSL certificate private key.
* A relative path is interpreted relative to $SPLUNK_HOME and may not refer
  outside of $SPLUNK_HOME (e.g., no ../somewhere).
* You can also specify an absolute path to an external key.
* See also 'enableSplunkWebSSL' and 'serverCert'.
* Default: $SPLUNK_HOME/etc/auth/splunkweb/privkey.pem

serverCert = <path>
* Full path to the Privacy Enhanced Mail (PEM) format Splunk web server certificate file.
* The file may also contain root and intermediate certificates, if required.
  They should be listed sequentially in the order:
    [ Server SSL certificate ]
    [ One or more intermediate certificates, if required ]
    [ Root certificate, if required ]
* See also 'enableSplunkWebSSL' and 'privKeyPath'.
* Default: $SPLUNK_HOME/etc/auth/splunkweb/cert.pem

sslPassword = <password>
* Password that protects the private key specified by 'privKeyPath'.
* If encrypted private key is used, do not enable client-authentication
  on splunkd server. In [sslConfig] stanza of server.conf,
  'requireClientCert' must be 'false'.
* Optional.
* Default: The unencrypted private key.
```

```
caCertPath = <path>
* DEPRECATED.
* Use 'serverCert' instead.
* A relative path is interpreted relative to $SPLUNK_HOME and may not refer
  outside of $SPLUNK_HOME (e.g., no ../somewhere).
* No default.

sslRootCAPath = <path>
* The path to a root certificate authority (CA) certificate, in privacy-enhanced
  mail (PEM) format, that splunkd is to use to authenticate client certificates
  under certain specific conditions.
* Splunkd uses the certificate specified at the path defined in this setting only
  when both 'requireClientCert' and 'enableCertBasedUserAuth' have a value of "true".
* If this setting has no value, splunkd falls back to the value of the 'sslRootCAPath'
  setting in server.conf.
* If you have already configured 'sslRootCAPath' in server.conf, the value of this
  setting does not override the setting of the same name in server.conf.
* No default.

enableCertBasedUserAuth = <boolean>
* Whether or not user authentication with certificates is enabled.
* When certificate-based authentication is enabled, splunkd uses a digital certificate
  to identify and grant a user access to a Splunk platform instance resource.
* A value of "true" means that splunkd uses certificates for authentication.
  * When this setting has a value of "true", 'requireClientCert' must *also* have a value of "true".
* A value of "false" means that splunkd does not use certificates for authentication.
* NOTE: Splunkd disables the check to determine if Splunk Web is serving web
  requests after it completes startup when this setting has a value of "true".
  If you need this check to happen, then this setting *must* have a
  value of "false".
* Default: false

certBasedUserAuthMethod = <string>
* The method that the Splunk platform uses to extract LDAP credentials from client certificates.
* This setting takes one of the following values:
  * CommonName: Use the value contained in the Common Name field of a client certificate in its entirety
  * EDIPI (Electronic Data Interchange Personal Identifier): Extract the EDIPI, the 10-digit numeric
identifier
    from the Common Name. If the platform can't find the EDIPI, then it uses the Common Name in its
entirety.
  * PIV (Personal Identity Verification): Use PIV, a 16-digit numeric identifier typically formatted
    as xxxxxxxxxxxxxxxx@mil. It is extracted from an "Other Name" field in the Subject Alternate Name which
    corresponds to one of the object identifiers (OIDs) that you configure in
'certBasedUserAuthPivOidList'.
* No default.

certBasedUserAuthPivOidList = <comma-separated list>
* A list of object identifiers (OIDs) that the Splunk platform uses to
  lookup an end-user's PIV info in the Subject Alternate Name extension of the client certificate.
* The Splunk platform queries OIDs sequentially in a client certificate until it finds an OID with a value.
* The value contained in the matched OID is then used to authenticate the user.
* Default: 1.3.6.1.4.1.311.20.2.3, Microsoft Universal Principal Name, Microsoft User Principal Name

requireClientCert = <boolean>
* Whether or not an HTTPS client that connects to the Splunk Web HTTPS server
  must present a certificate that was signed by the same certificate authority (CA)
  that signed the certificate that was installed on this instance.
* A value of "true" means the following:
  * A client can connect *only* if it presents a certificate that was created
    and signed by the same CA that created the certificate that the instance uses
  * You must configure splunkd with the same root CA in the server.conf file.
```

```
    This requirement ensures proper communication between splunkd and Splunk Web.
* If you give 'enableCertBasedUserAuth' a value of "true", then the previous
  statements do not apply. Instead, the instance uses the root CA certificate
  defined in the 'sslRootCAPath' setting in web.conf, and if no
  certificate path is defined in that file, it then uses the certificate
  defined in the 'sslRootCAPath' setting in server.conf.
* A value of "false" means that clients do not need to present a certificate
  to connect to the instance.
* Default: false

sslCommonNameToCheck = <commonName1>, <commonName2>, ...
* Checks the common name of the client's certificate against this list of names.
* 'requireClientCert' must be set to "true" for this setting to work.
* Optional.
* Default: empty string (No common name checking).

sslAltNameToCheck = <alternateName1>, <alternateName2>, ...
* If this value is set, and 'requireClientCert' is set to true,
  Splunk Web will verify certificates which have a so-called
  "Subject Alternate Name" that matches any of the alternate names in this list.
  * Subject Alternate Names are effectively extended descriptive
    fields in SSL certs beyond the commonName. A common practice for
    HTTPS certs is to use these values to store additional valid
    hostnames or domains where the cert should be considered valid.
* Accepts a comma-separated list of Subject Alternate Names to consider valid.
* Optional.
* Default: empty string (no alternate name checking).

serviceFormPostURL = http://docs.splunk.com/Documentation/Splunk
* DEPRECATED.
* This setting has been deprecated since Splunk Enterprise version 5.0.3.

userRegistrationURL = https://www.splunk.com/page/sign_up
updateCheckerBaseURL = http://quickdraw.Splunk.com/js/
docsCheckerBaseURL = http://quickdraw.splunk.com/help
* These are various Splunk.com urls that are configurable.
* Setting 'updateCheckerBaseURL' to 0 stops Splunk Web from pinging
  Splunk.com for new versions of Splunk software.

enable_insecure_login = <boolean>
* Whether or not the GET-based "/account/insecurelogin" REST endpoint is enabled.
* Provides an alternate GET-based authentication mechanism.
* If "true", the following url is available:
http://localhost:8000/en-US/account/insecurelogin?loginType=splunk&username=noc&password=XXXXXXX
* If "false", only the main /account/login endpoint is available
* Default: false

enable_secure_entity_move = <boolean>
* Whether or not you can perform an HTTP GET request on the "move" REST endpoint
  for any entity that has such an endpoint, to move that entity from one Splunk app
  to another.
* Entities are configurable components of the Splunk Web framework, such as views,
  styles, and drilldown actions. This is not an exhaustive list.
* If set to "true", you can perform only HTTP POST requests against the "move" endpoint
  for an entity.
  * For example, if you have an endpoint "/en_US/manager/launcher/data/ui/views/move",
    you can only perform an HTTP POST request to access that endpoint to move
    an entity from one app to another.
* If set to "false", you can perform both HTTP GET and POST requests against the
  "move" endpoint of an entity.
* Default: true
```

```
enable_insecure_pdfgen = <boolean>
* Whether or not the "/services/pdfgen/render" REST endpoint allows GET requests.
* If "true", allows PDFs to be generated using GET or POST requests.
* If "false", only allows PDFs to be generated using POST requests.
* Default: false

simple_error_page = <boolean>
* Whether or not to display a simplified error page for HTTP errors that only contains the error status.
* If set to "true", Splunk Web displays a simplified error page for errors (404, 500, etc.) that only
contain the error status.
* If set to "false", Splunk Web displays a more verbose error page that contains the home link, message, a
more_results_link, crashes, referrer, debug output, and byline
* Default: false

login_content = <string>
* Lets you add custom content to the login page.
* Supports any text including HTML.
* No default.

sslVersions = <comma-separated list>
* A comma-separated list of SSL versions to support.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2"
* The special version "*" selects all supported versions. The version "tls"
  selects all versions tls1.0 or newer
* If you prefix a version with "-", it is removed from the list.
* SSLv2 is always disabled; "-ssl2" is accepted in the version list, but does nothing.
* When configured in FIPS mode, "ssl3" is always disabled regardless
  of this configuration.
* For the default, see $SPLUNK_HOME/etc/system/default/web.conf.

supportSSLV3Only = <boolean>
* This setting is DEPRECATED. SSLv2 is now always disabled.
  The exact set of SSL versions allowed is now configurable via the
  'sslVersions' setting above.

cipherSuite = <cipher suite string>
* If set, uses the specified cipher string for the HTTP server.
* If not set, uses the default cipher string provided by OpenSSL. This is
  used to ensure that the server does not accept connections using weak
  encryption protocols.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* The default can vary. See the cipherSuite setting in
* $SPLUNK_HOME/etc/system/default/web.conf for the current default.

ecdhCurveName = <string>
* DEPRECATED.
* Use the 'ecdhCurves' setting instead.
* This setting specifies the Elliptic Curve Diffie-Hellman (ECDH) curve to
  use for ECDH key negotiation.
* Splunk only supports named curves that have been specified by their
  SHORT name.
* The list of valid named curves by their short and long names
  can be obtained by running this CLI command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Default: empty string.

ecdhCurves = <comma-separated list>
* A list of ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of an SSL Client Hello.
* The server supports only the curves specified in the list.
* Splunk software only supports named curves that have been specified
```

```
  by their SHORT names.
* The list of valid named curves by their short and long names can be obtained
  by running this CLI command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
* Example setting: "ecdhCurves = prime256v1,secp384r1,secp521r1"
* The default can vary. See the 'ecdhCurves' setting in
  $SPLUNK_HOME/etc/system/default/web.conf for the current default.

dhFile = <path>
* Full path to the Diffie-Hellman parameter file.
* Relative paths are interpreted as relative to $SPLUNK_HOME, and must
  not refer to a location outside of $SPLUNK_HOME.
* This file is required in order to enable any Diffie-Hellman ciphers.
* Default: not set.

root_endpoint = <URI_prefix_string>
* Defines the root URI path on which the appserver will listen
* For example, if you want to proxy the splunk UI at http://splunk:8000/splunkui,
  then set root_endpoint = /splunkui
* Default: /

static_endpoint = <URI_prefix_string>
* Path to static content.
* The path here is automatically appended to root_endpoint defined above
* Default: /static

static_dir = <relative_filesystem_path>
* The directory that holds the static content
* This can be an absolute URL if you want to put it elsewhere
* Default: share/splunk/search_mrsparkle/exposed

rss_endpoint = <URI_prefix_string>
* Path to static rss content
* The path here is automatically appended to what you defined in the
  'root_endpoint' setting
* Default: /rss

embed_uri = <URI>
* Optional URI scheme/host/port prefix for embedded content
* This presents an optional strategy for exposing embedded shared
  content that does not require authentication in a reverse proxy/single
  sign on environment.
* Default: empty string, resolves to the client
  window.location.protocol + "//" + window.location.host

embed_footer = <html_string>
* A block of HTML code that defines the footer for an embedded report.
* Any valid HTML code is acceptable.
* Default: "splunk>"

tools.staticdir.generate_indexes = [1 | 0]
* Whether or not the webserver serves a directory listing for static
  directories.
* Default: 0 (false)

template_dir = <relative_filesystem_path>
* The base path to the Mako templates.
* Default: "share/splunk/search_mrsparkle/templates"

module_dir = <relative_filesystem_path>
* The base path to Splunk Web module assets.
* Default: "share/splunk/search_mrsparkle/modules"
```

```
enable_gzip = <boolean>
* Whether or not the webserver applies gzip compression to responses.
* Default: true

use_future_expires = <boolean>
* Whether or not the Expires header of /static files is set to a far-future date
* Default: true

flash_major_version = <integer>
* DEPRECATED.
flash_minor_version = <integer>
* DEPRECATED.
flash_revision_version = <integer>
* DEPRECATED.
* Specifies the minimum Flash plugin version requirements
* Flash support, broken into three parts.
* We currently require a min baseline of Shockwave Flash 9.0 r124

override_JSON_MIME_type_with_text_plain = <boolean>
* Whether or not to override the MIME type for JSON data served up
  by Splunk Web endpoints with content-type="text/plain; charset=UTF-8"
* If "true", Splunk Web endpoints (other than proxy) that serve JSON data will
  serve as "text/plain; charset=UTF-8"
* If "false", Splunk Web endpoints that serve JSON data will serve as "application/json; charset=UTF-8"

enable_proxy_write = <boolean>
* Indicates if the /splunkd proxy endpoint allows POST operations.
* If "true", both GET and POST operations are proxied through to splunkd.
* If "false", only GET operations are proxied through to splunkd.
* Setting to "false" prevents many client-side packages (such as the
  Splunk JavaScript SDK) from working correctly.
* Default: true

js_logger_mode = [None | Firebug | Server]
* The JavaScript Logger mode.
* Available modes: None, Firebug, Server
* Mode None: Does not log anything.
* Mode Firebug: Use firebug by default if it exists, or defer to the older
  less promiscuous version of firebug lite.
* Mode Server: Log to a defined server endpoint.
* See js/logger.js Splunk.Logger.Mode for mode implementation details and if
  you would like to author your own.
* Default: None

js_logger_mode_server_end_point = <URI_relative_path>
* The server endpoint to post JavaScript log messages
* Used when js_logger_mode = Server
* Default: util/log/js

js_logger_mode_server_poll_buffer = <integer>
* The interval, in milliseconds, to check, post, and cleanse the JavaScript log buffer
* Default: 1000

js_logger_mode_server_max_buffer = <integer>
* The maximum size threshold, in megabytes, to post and cleanse the JavaScript log buffer
* Default: 100

ui_inactivity_timeout = <integer>
* The length of time lapsed, in minutes, for notification when
  there is no user interface clicking, mouseover, scrolling, or resizing.
* Notifies client side pollers to stop, resulting in sessions expiring at
```

```
  the 'tools.sessions.timeout' value.
* If less than 1, results in no timeout notification ever being triggered
  (Sessions stay alive for as long as the browser is open).
* Default: 60

js_no_cache = <boolean>
* DEPRECATED.
* Toggles the JavaScript cache control.
* Default: false

cacheBytesLimit = <integer>
* Splunkd can keep a small cache of static web assets in memory.
  When the total size of the objects in cache grows larger than this setting,
  in bytes, splunkd begins ageing entries out of the cache.
* If set to zero, disables the cache.
* Default: 4194304

cacheEntriesLimit = <integer>
* Splunkd can keep a small cache of static web assets in memory.
  When the number of the objects in cache grows larger than this,
  splunkd begins ageing entries out of the cache.
* If set to zero, disables the cache.
* Default: 16384

staticCompressionLevel = <integer>
* Splunkd can keep a small cache of static web assets in memory.
  Splunkd stores these assets in a compressed format, and the assets can
  usually be served directly to the web browser in compressed format.
* This level can be a number between 1 and 9.  Lower numbers use less
  CPU time to compress objects, but the resulting compressed objects
  will be larger.
* There is not much benefit to decreasing the value of this setting from
  its default. Not much CPU time is spent compressing the objects.
* Default: 9

enable_autocomplete_login = <boolean>
* Indicates if the main login page lets browsers autocomplete the username.
* If "true", browsers may display an autocomplete drop down in the username field.
* If "false", browsers may not show autocomplete drop down in the username field.
* Default: false

verifyCookiesWorkDuringLogin = <boolean>
* Normally, the login page makes an attempt to see if cookies work
  properly in the user's browser before allowing them to log in.
* If you set this to "false", this check is skipped.
* Do not set to "false" in normal operations.
* Default: true

minify_js = <boolean>
* Whether or not the static JavaScript files for modules are consolidated and minified.
* A value of "true" means that JavaScript files for modules are consolidated and
  minified. This improves client-side performance by reducing the number of HTTP
  requests and the size of HTTP responses.
* A value of "false" means that JavaScript files for modules are not consolidated
  or minified.
* Default: true

minify_css = <boolean>
* Whether or not the static CSS files for modules are consolidated and minified.
* A value of "true" means that static CSS files for modules are consolidated and
  minified. This improves client-side performance by reducing the number of HTTP
  requests and the size of HTTP responses.
```

```
* A value of "false" means that static CSS files for modules are not consolidated
  or minified.
* Due to browser limitations, setting this to "false" when using Internet Explorer
  version 9 and lower might result in display problems.
* Default: true

trap_module_exceptions = <boolean>
* Whether or not the JavaScript for individual modules is wrapped in a try/catch
* If "true", syntax errors in individual modules do not cause the UI to
  hang, other than when using the module in question.
* Set to "false" when developing apps.

enable_pivot_adhoc_acceleration = <boolean>
* DEPRECATED in version 6.1 and later, use 'pivot_adhoc_acceleration_mode'
  instead
* Whether or not the pivot interface uses its own ad-hoc acceleration
  when a data model is not accelerated.
* If "true", the pivot interface uses ad-hoc acceleration to make reporting
  in pivot faster and more responsive.
* In situations where data is not stored in time order, or where the majority
  of events are far in the past, disabling this behavior can improve the
  pivot experience.

pivot_adhoc_acceleration_mode = [Elastic | AllTime | None]
* Specifies the type of ad-hoc acceleration used by the pivot interface when a
  data model is not accelerated.
* If "Elastic", the pivot interface only accelerates the time range
  specified for reporting, and dynamically adjusts when this time range
  is changed.
* If "AllTime", the pivot interface accelerates the relevant data over all
  time. This makes the interface more responsive to time-range changes
  but places a larger load on system resources.
* If "None", the pivot interface does not use any acceleration. This means
  any change to the report requires restarting the search.
* Default: Elastic

jschart_test_mode = <boolean>
* Whether or not the JSChart module runs in Test Mode.
* If "true", JSChart module attaches HTML classes to chart elements for
  introspection.
* This negatively impacts performance and should be disabled unless you
  are actively using JSChart Test Mode.

#
# To avoid browser performance impacts, the JSChart library limits
# the amount of data rendered in an individual chart.

jschart_truncation_limit = <integer>
* Cross-broswer truncation limit.
* If set, takes precedence over the browser-specific limits below

jschart_truncation_limit.chrome = <integer>
* Chart truncation limit.
* For Chrome only.
* Default: 50000

jschart_truncation_limit.firefox = <integer>
* Chart truncation limit.
* For Firefox only.
* Default: 50000

jschart_truncation_limit.safari = <integer>
```

```
* Chart truncation limit.
* For Safari only.
* Default: 50000

jschart_truncation_limit.ie11 = <integer>
* Chart truncation limit.
* For Internet Explorer version 11 only
* Default: 50000

jschart_series_limit = <integer>
* Chart series limit for all browsers.
* Default: 100

jschart_results_limit = <integer>
* DEPRECATED.
* Use 'data_sources.primary.params.count' in visualizations.conf instead.
* Chart results per series limit for all browsers.
* Overrides the results per series limit for individual visualizations.
* Default: 10000

choropleth_shape_limit = <integer>
* Choropleth map shape limit for all browsers.
* Default: 10000

dashboard_html_allow_inline_styles = <boolean>
* Whether or not to allow style attributes from inline HTML elements in dashboards.
* If "false", style attributes from inline HTML elements in dashboards will be removed
  to prevent potential attacks.
* Default: true

dashboard_html_allow_embeddable_content = <boolean>
* Whether or not to allow <embed> and <iframe> HTML elements in dashboards.
* If set to "true", <embed> and <iframe> HTML elements in dashboards will not be removed
  and can lead to a potential security risk.
* If set to the default value of "false", <embed> and <iframe> HTML elements will be stripped
  from the dashboard HTML.
* Default: false

dashboard_html_wrap_embed = <boolean>
* Whether or not to wrap <embed> HTML elements in dashboards with an <iframe>.
* If set to "false", <embed> HTML elements in dashboards will not be wrapped, leading to
  a potential security risk.
* If set to "true", <embed> HTML elements will be wrapped by an <iframe sandbox> element to help
  mitigate potential security risks.
* Default: true

dashboard_html_allow_iframes = <boolean>
* Whether or not to allow iframes from HTML elements in dashboards.
* If "false", iframes from HTML elements in dashboards will be removed to prevent
  potential attacks.
* Default: true

dashboard_html_allowed_domains = <comma-separated list>
* A list of allowed domains for inline iframe element
  source ('<iframe src="<URL>">') attributes in dashboards.
* If the domain for an <iframe> src attribute is not an allowed
  domain, the Simple XML dashboard adds the 'sandbox' attribute to
  the <iframe>, which further restricts the content within the <iframe>
  by treating it as coming from a unique origin. Simple XML dashboards
  will allow <iframe> src attributes by default if the src is the same
  hostname and port number as the Splunk Web server's hostname and port number.
* You can specify these domains as a hostname or an IPV4 address or an IPV6 address.
```

```
* You can configure a hostname as a full name or with a wildcard
  to allow for any subdomains. For example, *.example.com would
  allow for any subdomain of example.com as well as example.com itself.
* You can specify an IPV4 address as an exact address or:
  * You can use an asterisk to specify a wildcard (Example: 192.168.1.*).
    Asterisks allow for any address within that byte segment.
  * You can use a dash to specify a range of addresses (Example: 192.168.1.1-99).
    Dashes will only match IP addresses within that range.
* You can specify an IPV6 address either as an exact address or with
  a subnet mask. If you specify a subnet mask, any IPV6 address within
  the subnet will be an allowed domain.
* You can specify a port number for any of the domains. If you do, the '<iframe src>'
  must match the port number as well.
* Additional configuration examples:
  * Hostname: docs.splunk.com, *.splunk.com
  * IPV4: 127.0.0.1, 127.0.0.*, 127.0-10.0.*, 127.0.0.1:8000
  * IPV6: ::1, [::1]:8000, 2001:db8:abcd:12::, 2001:db8::/32
* Default: not set

pdfgen_trusted_hosts = <string> [, <string>]
* A list of trusted hosts for inline image element source ('<image src="<URL>">')
  links used during a pdf export.
* If the domain for an <image> src attribute is not in the list of trusted hosts,
  the image will not download during PDF export.
* Separate multiple rules with commas.
* Each rule can be in one of the following formats:
    1. A single IPv4 or IPv6 address (examples: "203.0.113.2", "2001:db8:3c4d")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
       (examples: "192.0.2.0/24", "2001:DB8::/32")
    3. A DNS name. Use "*" as a wildcard
       (examples: "myhost.example.com", "*.splunk.com")
    4. "*", which matches anything
* Any link which resolves to a loopback address will not download, unless the "*"
  rule is used.
* You can prefix an entry with '!' to cause the rule to reject the
  connection. The input applies rules in order, and uses the first one that
  matches.
  For example, "!192.0.2.0/24, *" allows connections from everywhere except
  the 192.0.2.* network.
* Default: not set. All links will fail by default.

max_view_cache_size = <integer>
* The maximum number of views to cache in the appserver.
* Default: 1000

pdfgen_is_available = [0 | 1]
* Specifies whether Integrated PDF Generation is available on this search
  head.
* This is used to bypass an extra call to splunkd.
* Default (on platforms where node is supported): 1
* Default (on platforms where node is not supported): 0

version_label_format = <printf_string>
* Internal configuration.
* Overrides the version reported by the UI to *.splunk.com resources
* Default: %s

auto_refresh_views = [0 | 1]
* Specifies whether the following actions cause the appserver to ask splunkd
  to reload views from disk.
  * Logging in through Splunk Web
  * Switching apps
```

```
  * Clicking the Splunk logo
* Default: 0

show_app_context = <boolean>
* Whether or not Splunk Web will show app context in certain locations.
* You can set this to "false" in situations where you do not want to display app contexts,
  for example, when apps are under cluster management.
* Default: true

cookieSameSite = [ not_specified | lax | strict | none ]
* The value of the "SameSite" cookie attribute for which the Splunk web server
  is to set in the web browser.
* A value of "not_specified" means the Splunk web server does not set the "SameSite"
  attribute. Some browsers, like Chrome, interpret this as if the web server set
  "SameSite=Lax", according to the latest Internet-Drafts as published by the
  Internet Engineering Task Force (IETF). Other browsers might interpret this as
  if the web server set "SameSite=None".
* A value of "lax" means the Splunk web server sets the "SameSite=Lax" attribute.
* A value of "strict" means the Splunk web server sets the "SameSite=Strict" attribute.
* A value of "none" means the Splunk web server sets the "SameSite=None"
  cookie attribute, to let browsers send the cookies in all contexts.
  A cookie with this attribute lets the browser embed a Splunk dashboard
  into a third-party <iframe> component.
  * <iframe> stands for "inline frame", and is a web page component within
    which you can embed a dashboard.
* If you want to embed a Splunk dashboard into an outside web application,
  you must give this setting a value of "none". Otherwise, the third-party
  <iframe> won't let the user authenticate and use the dashboard that
  you embedded.
* The "SameSite=None" attribute in a cookie requires that you set the
  "Secure" attribute, otherwise the browser might reject the cookie. The Splunk web server
  adds the "Secure" attribute for connections over HTTPS by default. Use the
  'tools.sessions.secure' setting to configure this behavior.
* You must also give the 'x_frame_options_sameorigin' setting a value of "false"
  to allow for the embedding of a Splunk dashboard in the <iframe>.
* Default: not_specified

#
# Splunk bar options
#
# Internal config. May change without notice.
# Only takes effect if 'instanceType' is 'cloud'.
#

showProductMenu = <boolean>
* Used to indicate visibility of product menu.
* Default: False.

productMenuUriPrefix = <string>
* The domain product menu links to.
* Required if 'showProductMenu' is set to "true".

productMenuLabel = <string>
* Used to change the text label for product menu.
* Default: 'My Splunk'

showUserMenuProfile = <boolean>
* Used to indicate visibility of 'Profile' link within user menu.
* Default: false


#
```

```
# Header options
#
x_frame_options_sameorigin = <boolean>
* adds a X-Frame-Options header set to "SAMEORIGIN" to every response served
* by cherrypy
* Default: true

#
# Single Sign On (SSO)
#

remoteUser = <http_header_string>
* Remote user HTTP header sent by the authenticating proxy server.
* This header should be set to the authenticated user.
* CAUTION: There is a potential security concern regarding the
  treatment of HTTP headers.
* Your proxy provides the selected username as an HTTP header as specified
  above.
* If the browser or other HTTP agent were to specify the value of this
  header, probably any proxy would overwrite it, or in the case that the
  username cannot be determined, refuse to pass along the request or set
  it blank.
* However, Splunk Web (specifically, cherrypy) normalizes headers containing
  the dash and the underscore to the same value. For example, USER-NAME and
  USER_NAME are treated as the same in Splunk Web.
* This means that if the browser provides REMOTE-USER and Splunk Web accepts
  REMOTE_USER, theoretically the browser could dictate the username.
* In practice, however, the proxy adds its headers last, which causes them
  to take precedence, making the problem moot.
* See also the 'remoteUserMatchExact' setting which can enforce more exact
  header matching.
* Default: 'REMOTE_USER'

remoteGroups = <http_header_string>
* Remote groups HTTP header name sent by the authenticating proxy server.
* This value is used by Splunk Web to match against the header name.
* The header value format should be set to comma-separated groups that
  the user belongs to.
* Example of header value: Products,Engineering,Quality Assurance
* No default.

remoteGroupsQuoted = <boolean>
* Whether or not the group header value can be comma-separated quoted entries.
* This setting is considered only when 'remoteGroups' is set.
* If "true", the group header value can be comma-separated quoted entries.
* NOTE: Entries themselves can contain commas.
* Example of header value with quoted entries:
  "Products","North America, Engineering","Quality Assurance"
* Default: false (group entries should be without quotes.)

remoteUserMatchExact = [0 | 1]
* Whether or not to consider dashes and underscores in a remoteUser header
  to be distinct.
* When set to "1", considers dashes and underscores distinct (so
  "Remote-User" and "Remote_User" are considered different headers.)
* When set to 0, dashes and underscores are not considered to be distinct,
  to retain compatibility with older versions of Splunk software.
* Set to 1 when you set up SSO
* Default: 0

remoteGroupsMatchExact = [0 | 1]
* Whether or not to consider dashes and underscores in a remoteGroup header
```

```
  to be distinct.
* When set to 1, considers dashes and underscores distinct (so
  "Remote-Groups" and "Remote_Groups" are considered different headers)
* When set to 0, dashes and underscores are not considered to be distinct,
  to retain compatibility with older versions of Splunk software.
* Set to 1 when you set up SSO
* Default: 0

SSOMode = [permissive | strict]
* Whether SSO behaves in either permissive or strict mode.
* When set to "permissive": Requests to Splunk Web that originate from an
  untrusted IP address are redirected to a login page where they can log into
  Splunk Web without using SSO.
* When set to "strict": All requests to Splunk Web will be restricted to those
  originating from a trusted IP except those to endpoints that do not require
  authentication.
* Default: strict

trustedIP = <ip_addresses>
* IP addresses of the authenticating proxy (trusted IP).
* Splunk Web verifies it is receiving data from the proxy host for all
  SSO requests.
* Set to a valid IP address to enable SSO.
* This setting can accept a list of IPs or networks, using the same format
  as the 'acceptFrom' setting.
* Default: not set; the normal value is the loopback address (127.0.0.1).

allowSsoWithoutChangingServerConf = [0 | 1]
* Whether or not to allow SSO without setting the 'trustedIP' setting in
  server.conf as well as in web.conf.
* If set to 1, enables web-based SSO without a 'trustedIP' setting configured
  in server.conf.
* Default: 0

ssoAuthFailureRedirect = <scheme>://<URL>
* The redirect URL to use if SSO authentication fails.
* Examples:
  * http://www.example.com
  * https://www.example.com
* Default: empty string; Splunk Web shows the default unauthorized error
  page if SSO authentication fails.

# Results export server config

export_timeout = <integer>
* When exporting results, the number of seconds the server waits before
  closing the connection with splunkd.
* If you do not set a value for export_timeout, Splunk Web uses the value
  for the 'splunkdConnectionTimeout' setting.
* Set 'export_timeout' to a value greater than 30 in normal operations.
* No default.

#
# cherrypy HTTP server config
#

server.thread_pool = <integer>
* Determines the minimum number of threads the appserver is allowed to maintain.
* The default value of this setting provides acceptable performance for most use
  cases.
* If you are experiencing issues with UI latency, you can increase the value
  based on need, to a maximum value of 200.
```

```
* Values that exceed 200 can cause memory spikes.
* Default: 50

server.socket_host = <ip_address>
* Host values may be any IPv4 or IPv6 address, or any valid hostname.
* The string 'localhost' is a synonym for '127.0.0.1' (or '::1', if your
  hosts file prefers IPv6).
* The string '0.0.0.0' is a special IPv4 entry meaning "any active interface"
  (INADDR_ANY), and "::" is the similar IN6ADDR_ANY for IPv6.
* Default (if 'listenOnIPV6' is set to "no": 0.0.0.0
* Default (otherwise): "::"

server.socket_timeout = <integer>
* The timeout, in seconds, for accepted connections between the browser and
  Splunk Web
* Default: 10

listenOnIPv6 = <no | yes | only>
* By default, Splunk Web listens for incoming connections using
  IPv4 only.
* To enable IPv6 support in splunkweb, set this to "yes". Splunk Web
  simultaneously listens for connections on both IPv4 and IPv6 protocols.
* To disable IPv4 entirely, set to "only", which causes SPlunk Web
  to exclusively accept connections over IPv6.
* To listen on an IPV6 address, also set 'server.socket_host' to "::".

max_upload_size = <integer>
* The hard maximum limit, in megabytes, of uploaded files.
* Default: 500

log.access_file = <filename>
* The HTTP access log filename.
* This file is written in the default $SPLUNK_HOME/var/log directory.
* Default: web_access.log

log.access_maxsize = <integer>
* The maximum size, in bytes, that the web_access.log file can be.
* Comment out or set to 0 for unlimited file size.
* Splunk Web rotates the file to web_access.log.0 after the 'log.access_maxsize' is reached.
* See the 'log.access_maxfiles' setting to limit the number of backup files
  created.
* Default: 25000000 (25 MB).

log.access_maxfiles = <integer>
* The maximum number of backup files to keep after the web_access.log
  file has reached its maximum size.
* CAUTION: Setting this to very high numbers (for example, 10000) can affect
  performance during log rotation.
* Default (if 'access_maxsize' is set): 5

log.error_maxsize = <integer>
* The maximum size, in bytes, the web_service.log can be.
* Comment out or set to 0 for unlimited file size.
* Splunk Web rotates the file to web_service.log.0 after the
  max file size is reached.
* See 'log.error_maxfiles' to limit the number of backup files created.
* Default: 25000000 (25 MB).

log.error_maxfiles = <integer>
* The maximum number of backup files to keep after the web_service.log
  file has reached its maximum size.
* CAUTION: Setting this to very high numbers (for example, 10000) can affect
```

```
  performance during log rotations
* Default (if 'access_maxsize' is set): 5

log.screen = <boolean>
* Whether or not runtime output is displayed inside an interactive TTY.
* Default: true

request.show_tracebacks = <boolean>
* Whether or not an exception traceback is displayed to the user on fatal
  exceptions.
* Default: true

engine.autoreload.on = <boolean>
* Whether or not the appserver will auto-restart if it detects a python file
  has changed.
* Default: false

tools.sessions.on = true
* Whether or not user session support is enabled.
* Always set this to true.

tools.sessions.timeout = <integer>
* The number of minutes of inactivity before a user session is
  expired.
* The countdown for this setting effectively resets every minute through
  browser activity until the 'ui_inactivity_timeout' setting is reached.
* Use a value of 2 or higher, as a value of 1 causes a race condition with
  the browser refresh, producing unpredictable behavior.
* Low values are not useful except for testing.
* Default: 60

tools.sessions.restart_persist = <boolean>
* Whether or not the session cookie is deleted from the browser when the
  browser quits.
* If set to "false", then the session cookie is deleted from the browser
  upon the browser quitting.
* If set to "true", then sessions persist across browser restarts, assuming
  the 'tools.sessions.timeout' has not been reached.
* Default: true

tools.sessions.httponly = <boolean>
* Whether or not the session cookie is available to running JavaScript scripts.
* If set to "true", the session cookie is not available to running JavaScript
  scripts. This improves session security.
* If set to "false", the session cookie is available to running JavaScript
  scripts.
* Default: true

tools.sessions.secure = <boolean>
* Whether or not the browser must transmit session cookies over an HTTPS
  connection when Splunk Web is configured to serve requests using HTTPS
  (the 'enableSplunkWebSSL' setting is "true".)
* If set to "true" and 'enableSplunkWebSSL' is also "true", then the
  browser must transmit the session cookie over HTTPS connections.
  This improves session security.
* See the 'enableSplunkWebSSL' setting for details on configuring HTTPS
  session support.
* Default: true

tools.sessions.forceSecure = <boolean>
* Whether or not the secure bit of a session cookie that has been sent
  over HTTPS is set.
```

```
* If a client connects to a proxy server over HTTPS, and the back end
  connects to Splunk over HTTP, then setting this to "true" forces the
  session cookie being sent back to the client over HTTPS to have the
  secure bit set.
* Default: false

response.timeout = <integer>
* The timeout, in seconds, to wait for the server to complete a
  response.
* Some requests, such as uploading large files, can take a long time.
* Default: 7200 (2 hours).

tools.sessions.storage_type = [file]
tools.sessions.storage_path = <filepath>
* Specifies the session information storage mechanisms.
* Set 'tools.sessions.storage_type' and 'tools.sessions.storage_path' to
  use RAM based sessions instead.
* Use an absolute path to store sessions outside of $SPLUNK_HOME.
* Default: storage_type=file, storage_path=var/run/splunk

tools.decode.on = <boolean>
* Whether or not all strings that come into CherryPy controller methods are
  decoded as unicode (assumes UTF-8 encoding).
* CAUTION: Setting this to false will likely break the application, as
  all incoming strings are assumed to be unicode.
* Default: true

tools.encode.on = <boolean>
* Whether or not to encode all controller method response strings into
  UTF-8 str objects in Python.
* CAUTION: Disabling this will likely cause high byte character encoding to
  fail.
* Default: true

tools.encode.encoding = <codec>
* Forces all outgoing characters to be encoded into UTF-8.
* This setting only takes effect when 'tools.encode.on' is set to "true".
* By setting this to "utf-8", CherryPy default behavior of observing the
  Accept-Charset header is overwritten and forces utf-8 output.
* Only change this if you know a particular browser installation must
  receive some other character encoding (Latin-1 iso-8859-1, etc)
* CAUTION: Change this setting at your own risk.
* Default: utf-8

tools.encode.text_only = <boolean>
# Controls CherryPy's ability to encode content type. If set to True, CherryPy will only encode
# text (text/*) content. As of the Python 3 conversion we are defaulting to False as the current
# controller responses are in Unicode.
# WARNING: Change this at your own risk.
* Default: False

tools.proxy.on = <boolean>
* Whether or not the Splunk platform instance is behind a reverse proxy server.
* If set to "true", the instance assumes that it is behind a reverse proxy and
  uses HTTP header information from the proxy to log access requests, secure
  its cookies properly, and generate valid URLs for redirect responses.
* All of the instance's HTTP services will use information from
  "X-Forwarded-*", "Front-End-Https", and "X-Url-Scheme" headers, where
  available, to override what it receives from proxied requests.
* If you set this to "true", you must also set 'tools.proxy.base' to a valid
  host name and network port.
* If set to "false", the instance relies on its own internal HTTP server
```

```
  settings and the immediate client's HTTP headers for the information needed
  for access request logging, cookie securing, and redirect URL generation.
* Default: false

tools.proxy.base = <scheme>://<URL>
* The proxy base URL in Splunk Web.
* Default: empty string

pid_path = <filepath>
* Specifies the path to the Process IDentification (pid) number file.
* Must be set to "var/run/splunk/splunkweb.pid".
* CAUTION: Do not change this parameter.

enabled_decomposers = <intention> [, <intention>]...
* Added in Splunk 4.2 as a short term workaround measure for apps which
  happen to still require search decomposition, which is deprecated
  with 4.2.
* Search decomposition will be entirely removed in a future release.
* A comma-separated list of allowed intentions.
* Modifies search decomposition, which is a Splunk Web internal behavior.
* Can be controlled on a per-app basis.
* If set to an empty string, no search decomposition occurs, which causes
  some usability problems with Report Builder.
* The current possible values are: addcommand, stats, addterm, addtermgt,
  addtermlt, setfields, excludefields, audit, sort, plot
* Default: "plot", leaving only the plot intention enabled.

simple_xml_perf_debug = <boolean>
* Whether or not Simple XML dashboards log performance metrics to the
  browser console.
* If set to "true", Simple XML dashboards log some performance metrics to
  the browser console.
* Default: false

job_default_auto_cancel = <integer>
* The amount of time, in seconds, of inactivity in Splunk Web, after which the search job automatically
cancels.
* Default: 62

job_min_polling_interval = <integer>
* The minimum polling interval, in milliseconds, for search jobs.
* This is the intial wait time for fetching results.
* The poll period increases gradually from the minimum interval
  to the maximum interval when search is in a queued or parsing
  state (and not a running state) for some time.
* Set this value between 100 and 'job_max_polling_interval' milliseconds.
* Default: 100

job_max_polling_interval = <integer>
* The maximum polling interval, in milliseconds, for search jobs.
* This is the maximum wait time for fetching results.
* In normal operations, set to 3000.
* Default: 1000

acceptFrom = <network_acl> ...

* Lists a set of networks or addresses from which to accept connections.
* Separate multiple rules with commas or spaces.
* Each rule can be in one of the following formats:
    1. A single IPv4 or IPv6 address (examples: "10.1.2.3", "fe80::4a3")
    2. A Classless Inter-Domain Routing (CIDR) block of addresses
       (examples: "10/8", "192.168.1/24", "fe80:1234/32")
```

```
    3. A DNS name, possibly with a "*" used as a wildcard
       (examples: "myhost.example.com", "*.splunk.com")
    4. "*", which matches anything
* You can also prefix an entry with '!' to cause the rule to reject the
  connection. The input applies rules in order, and uses the first one that
  matches.
  For example, "!10.1/16, *" allows connections from everywhere except
  the 10.1.*.* network.
* Default: "*" (accept from anywhere)

maxThreads = <integer>
* The number of threads that can be used for active HTTP transactions.
* This value can be limited to constrain resource usage.
* If set to 0, a limit is automatically picked based on
  estimated server capacity.
* If set to a negative number, no limits are enforced.
* Default: 0

maxSockets = <integer>
* The number of simultaneous HTTP connections that Splunk Web can accept.
* This value can be limited to constrain resource usage.
* If set to 0, a limit is automatically picked based on estimated
  server capacity.
* If set to a negative number, no limits are enforced.
* Default: 0

keepAliveIdleTimeout = <integer>
* How long, in seconds, that the Splunk Web HTTP server lets a keep-alive
  connection remain idle before forcibly disconnecting it.
* If this number is less than 7200, it will be set to 7200.
* Default: 7200

busyKeepAliveIdleTimeout = <integer>
* How long, in seconds, that the Splunk Web HTTP server lets a keep-alive
  connection remain idle while in a busy state before forcibly
  disconnecting it.
* CAUTION: Too large a value that can result in file descriptor exhaustion
  due to idling connections.
* If this number is less than 12, it will be set to 12.
* Default: 12

forceHttp10 = auto|never|always
* How the HTTP server deals with HTTP/1.0 support for incoming
  clients.
* When set to "always", the REST HTTP server does not use some
  HTTP 1.1 features such as persistent connections or chunked
  transfer encoding.
* When set to "auto", it limits HTTP 1.1 features only if the
  client sent no User-Agent header, or if the user agent is known
  to have bugs in its HTTP/1.1 support.
* When set to "never", it always allows HTTP 1.1, even to
  clients it suspects might be buggy.
* Default: auto

crossOriginSharingPolicy = <origin_acl> ...
* A list of HTTP Origins for which to return Access-Control-Allow-*
  (CORS) headers.
* These headers tell browsers that Splunk Web trusts web applications
  at those sites to make requests to the REST interface.
* The origin is passed as a URL without a path component (for example
  "https://app.example.com:8000")
* This setting can take a list of acceptable origins, separated
```

```
  by spaces and/or commas
* Each origin can also contain wildcards for any part. Examples:
  *://app.example.com:* (either HTTP or HTTPS on any port)
  https://*.example.com (any host under example.com, including example.com itself)
* An address can be prefixed with a '!' to negate the match, with
  the first matching origin taking precedence. For example,
  "!*://evil.example.com:* *://*.example.com:*" to not avoid
  matching one host in a domain.
* "*" can also be used to match all origins.
* Default: empty string

crossOriginSharingHeaders = <string>
* A list of the HTTP headers to which splunkd sets
  "Access-Control-Allow-Headers" when replying to
  Cross-Origin Resource Sharing (CORS) preflight requests.
* The "Access-Control-Allow-Headers" header is used in response to
  a CORS preflight request to tell browsers which HTTP headers can be
  used during the actual request.
* A CORS preflight request is a CORS request that checks to see if
  the CORS protocol is understood and a server is aware of using
  specific methods and headers.
* This setting can take a list of acceptable HTTP headers, separated
  by commas.
* A single "*" can also be used to match all headers.
* Default: Empty string.

allowSslCompression = <boolean>
* Whether or not the server lets clients negotiate SSL-layer data
  compression.
* If set to "true", the server lets clients negotiate SSL-layer
  data compression.
* The HTTP layer has its own compression layer which is usually sufficient.
* Default: false

allowSslRenegotiation = <boolean>
* Whether or not the server lets clients renegotiate SSL connections.
* In the SSL protocol, a client may request renegotiation of the connection
  settings from time to time.
* Setting this to "false" causes the server to reject all renegotiation
  attempts, breaking the connection.
* This limits the amount of CPU a single TCP connection can use, but it
  can cause connectivity problems especially for long-lived connections.
* Default: true

sslServerHandshakeTimeout = <integer>
* The timeout, in seconds, for an SSL handshake to complete between an
  SSL client and the Splunk SSL server.
* If the SSL server does not receive a "Client Hello" from the SSL client within
  'sslServerHandshakeTimeout' seconds, the server terminates
  the connection.
* Default: 60

sendStrictTransportSecurityHeader = <boolean>
* Whether or not the REST interface sends a "Strict-Transport-Security"
  header with all responses to requests made over SSL.
* If set to "true", the REST interface sends a "Strict-Transport-Security"
  header with all responses to requests made over SSL.
* This can help avoid a client being tricked later by a Man-In-The-Middle
  attack to accept a non-SSL request.
* This requires a commitment that no non-SSL web hosts will ever be
  run on this hostname on any port. For example, if splunkweb is in default
  non-SSL mode this can break the ability of browser to connect to it.
```

```
* Enable this setting with caution.
* Default: false

includeSubDomains = <boolean>
* Whether or not the REST interface includes the "includeSubDomains"
  directive in the "Strict-Transport-Security" header with all responses
  to requests made over SSL.
* If set to "true", all subdomains of the current domain name will be
  enforced with the same HTTP Strict-Transport-Security (HSTS) policy.
* Can only be enabled if 'sendStrictTransportSecurityHeader' is set
  to "true".
* Enable this setting with caution. Enabling 'includeSubDomains' can have
  consquences by blocking access to subdomains that can only be served
  over HTTP.
* Default: false

preload = <boolean>
* Whether or not the REST interface includes the "preload" directive in the
  "Strict-Transport-Security" header with all responses to requests made
  over SSL.
* If set to "true", domains can be loaded on the HSTS preload list service
  that the Chromium project maintains for Google Chrome and various other
  browsers.
* Can only be enabled if 'sendStrictTransportSecurityHeader' is set
  to "true".
* Enable this setting with caution. Enabling 'preload' can have
  consequences by preventing users from accessing your domain and
  subdomains in the case of switching back to HTTP.
* Default: false

dedicatedIoThreads = <integer>
* The number of dedicated threads to use for HTTP input/output operations.
* If set to zero, HTTP I/O is performed in the same thread
  that accepted the TCP connection.
* If set set to a non-zero value, separate threads run
  to handle the HTTP I/O, including SSL encryption.
* Typically this does not need to be changed.  For most usage
  scenarios using the same the thread offers the best performance.
* Default: 0

replyHeader.<name> = <string>
* Adds a static header to all HTTP responses that this server generates.
* For example, "replyHeader.My-Header = value" causes Splunk Web to include
  the response header "My-Header: value" in the reply to every HTTP request
  to it.
* No default.

termsOfServiceDirectory = <directory>
* The directory to look in for a "Terms of Service" document that each
  user must accept before logging into Splunk Web.
  * Inside the directory the TOS should have a filename in the format
    "<number>.html"
  * <number> is in the range 1 to 18446744073709551615.
  * The active TOS is the filename with the larger number. For example, if
    there are two files in the directory named "123.html" and "456.html", then
    456 will be the active TOS version.
  * If a user has not accepted the current version of the TOS, they must
    accept it the next time they try to log in. The acceptance times will be recorded inside a "tos.conf"
file inside an app called "tos".
  * If the "tos" app does not exist, you must create it for acceptance
    times to be recorded.
  * The TOS file can either be a full HTML document or plain text, but it must
```

```
    have the ".html" suffix.
  * You do not need to restart Splunk Enterprise when adding files to the
    TOS directory.
* Default: empty string (no TOS)

appServerProcessShutdownTimeout = <nonnegative integer>[smhd]
* The amount of time splunkd waits for a Python-based application server
  process to handle outstanding or existing requests.
* If a Python-based application server process "outlives" this timeout,
  splunkd forcibly terminates the process.
* Default: '30s' (30 seconds).

appServerProcessLogStderr = <boolean>
* If set to true, messages written to the standard error stream by the
  Python-based application server processes will be logged to splunkd.log
  under the "UiAppServer" channel.
* This can be useful when debugging issues when the appserver process
  fails to start
* However, some appserver code may print sensitive information such as
  session ID strings to standard error so this defaults to disabled.
* Default: false

enableWebDebug = <boolean>
* Whether or not the debug REST endpoints are accessible, for example.,
  /debug/**splat.
* Default: false

allowableTemplatePaths =  <directory> [, <directory>]...
* A comma-separated list of template paths that might be added to
  the template lookup allow list.
* Paths are relative to $SPLUNK_HOME.
* Default: empty string

enable_risky_command_check = <boolean>
* Whether or not checks for data-exfiltrating search commands are enabled.
* Default: true

enable_risky_command_check_dashboard = <boolean>
* Whether or not checks for data-exfiltrating search commands within a dashboard are enabled.
* Default: true

enableSearchJobXslt = <boolean>
* REMOVED. This setting no longer has any effect.
* Whether or not the search job request accepts XML stylesheet language (XSL)
  as input to format search results.
* If set to "true", the search job request accepts XSL as input
  to format search results.
* If set to "false", the search job request does not accept XSL as input
  to format search results.
* Default: false

customFavicon = <pathToMyFile, myApp:pathToMyFile, or blank for default>
* Customizes the favicon image across the entire application.
* If no favicon image file, the favicon default: the Splunk favicon.
  * Supported favicon image files are .ico files, and should be square images.
  * Place the favicon image file in the default or manual location:
    * Default destination folder: $SPLUNK_HOME/etc/apps/search/appserver/static/customfavicon.
      * Example: If your favicon image is located at
$SPLUNK_HOME/etc/apps/search/appserver/static/customfavicon/favicon.ico, set 'customFavicon' to
"customfavicon/favicon.ico".
    * Manual location: Place the file in $SPLUNK_HOME/etc/apps/<myApp>/appserver/static/<pathToMyFile>, and
set 'customFavicon' to
```

```
    "<myApp:pathToMyFile>".
* Default: not set, Splunk Web uses the Splunk favicon.

loginCustomLogo = <fullUrl, pathToMyFile, myApp:pathToMyFile, or blank for default>
* Customizes the logo image on the login page.
* If no image file, the logo Default: the Splunk logo.
* Supported images are:
  * Full URL image file (secured or not secured), such as https://www.splunk.com/logo.png or
http://www.splunk.com/logo.png.
  * Image file, such as .jpg or .png. All image formats are supported.
    * Place logo image file in default or manual location:
      * Default destination folder: $SPLUNK_HOME/etc/apps/search/appserver/static/logincustomlogo.
        * Example: If your logo image is located at
$SPLUNK_HOME/etc/apps/search/appserver/static/logincustomlogo/logo.png, type loginCustomLogo =
logincustomlogo/logo.png.
      * Manual location: $SPLUNK_HOME/etc/apps/<myApp>/appserver/static/<pathToMyFile>, and type
loginCustomLogo = <myApp:pathToMyFile>.
* The maximum image size is 485px wide and 100px high. If the image exceeds these limits, the image is
automatically resized.
* Default: not set, Splunk Web uses the Splunk logo.

loginBackgroundImageOption = [default| custom | none]
* Controls display of the background image of the login page.
* "default" displays the Splunk default background image.
* "custom" uses the background image defined by the backgroundImageCustomName setting.
* "none" removes any background image on the login page. A dark background color is applied.
* Default: "default".

loginCustomBackgroundImage = <pathToMyFile or myApp:pathToMyFile>
* Customizes the login page background image.
  * Supported image files include .jpg, .jpeg or .png with a maximum file size of 20MB.
  * A landscape image is recommended, with a minimum resolution of 1024x640
    pixels.
  * Using Splunk Web:
    * Upload a custom image to a manager page under General Settings.
    * The login page background image updates automatically.
  * Using the CLI or a text editor:
    * Set 'loginBackgroundImageOption' to "custom".
    * Place the custom image file in the default or manual location:
      * Default destination folder: $SPLUNK_HOME/etc/apps/search/appserver/static/logincustombg.
        * Example: If your image is located at
$SPLUNK_HOME/etc/apps/search/appserver/static/logincustombg/img.png, set
        'loginCustomBackgroundImage' to "logincustombg/img.png".
      * Manual location: $SPLUNK_HOME/etc/apps/<myApp>/appserver/static/<pathToMyFile>, and set
'loginCustomBackgroundImage' to
      "<myApp:pathToMyFile>".
    * The login page background image updates automatically.
* Default: not set (If no custom image is used, the default Splunk background image displays).

loginFooterOption = [default | custom | none]
* Controls display of the footer message of the login page.
* "default" displays the Splunk copyright text.
* "custom" uses the footer text defined by the loginFooterText setting.
* "none" removes any footer text on the login page.
* NOTE: This option is made available only to OEM customers participating in
  the Splunk OEM Partner Program and is subject to the relevant terms of the Master OEM Agreement. All other
customers or partners are prohibited from
  removing or altering any copyright, trademark, and/or other intellectual
  property or proprietary rights notices of Splunk placed on or embedded
  in any Splunk materials.
* Default: "default".
```

```
loginFooterText = <footer_text>
* The text to display in the footer of the login page.
* Supports any text, including HTML.
* To display, the parameter 'loginFooterOption' must be set to "custom".

loginDocumentTitleOption = [default | custom | none]
* Controls display of the document title of the login page.
* Default: "default".
* "default" displays: "<page_title> | Splunk".
* "none" removes the branding on the document title of the login page: "<page_title>".
* "custom" uses the document title text defined by the loginDocumentTitleText setting.
* NOTE: This option is made available only to OEM customers participating in
  the Splunk OEM Partner Program and is subject to the relevant terms of the
  Master OEM Agreement. All other customers or partners are prohibited from
  removing or altering any copyright, trademark, and/or other intellectual
  property or proprietary rights notices of Splunk placed on or embedded
  in any Splunk materials.
* Default: "default".

loginDocumentTitleText = <document_title_text>
* The text to display in the document title of the login page.
* Text only.
* To display, the parameter 'loginDocumentTitleOption' must be set to "custom".

firstTimeLoginMessageOption = [default | custom | none]
* Controls display of the first time login message of the login page.
* "default" displays: "If you installed this instance, use the username and password you created at
installation.
   Otherwise, use the username and password that your Splunk administrator gave you. If you've forgotten
your
   credentials, contact your Splunk administrator."
* "none" removes the branding on the first time message of the login page: "".
* "custom" uses the document title text defined by the firstTimeLoginMessage setting.
* CAUTION: This setting is only configurable for original equipment manufacturer (OEM) customers that
participate
  in the Splunk OEM Partner Program. It is subject to the terms of the Master OEM Agreement. If you are not
  a member of this program, you MUST NOT remove or alter any Splunk copyright, trademark, and/or other
intellectual
  property or proprietary rights notices that Splunk embeds into any of its material. This action includes
but
  is not limited to configuring this setting.
* Default: default

firstTimeLoginMessage = <document_title_text>
* The text to display in the first time message of the login page.
* Text only.
* To display this message, you must first set 'firstTimeLoginMessageOption' to "custom".

loginPasswordHint = <default_password_hint>
* The text to display the password hint at first time login on the login page.
* Text only.
* Default: "The password you created when you installed this instance"

appNavReportsLimit = <integer>
* Maximum number of reports to fetch to populate the navigation drop-down
  menu of an app.
* An app must be configured to list reports in its navigation XML
  configuration before it can list any reports.
* Set to -1 to display all the available reports in the navigation menu.
* NOTE: Setting to either -1 or a value that is higher than the default might
  result in decreased browser performance due to listing large numbers of
  available reports in the drop-down menu.
```

```
* Default: 500

simplexml_dashboard_create_version = <string>
* DEPRECATED. The dashboard framework uses the latest Simple XML dashboard version for newly created
dashboards.
* CAUTION: Do not change this setting without contacting Splunk Support.
* The Simple XML dashboard version used for newly created Simple XML dashboards.
* Version must be a valid Simple XML dashboard version of the form 1.x (for example, 1.1).
* Default: 1.1

allow_insecure_libraries_toggle = <boolean>
* Determines whether or not Splunk Web can use insecure libraries which Splunk will deprecate.
* A value of "false" means Splunk Web cannot use insecure libraries.
* CAUTION: Do not change this setting.
* Default: true

# The Django bindings component and all associated [framework] settings have been
# removed. Configuring these settings no longer has any effect, and Splunk Enterprise
# ignores any existing settings that are related to the component.

[remoteUI]
* Set options to control the loading of remote hosted UI pages in Splunk Cloud
* If you do not specify an entry for each setting, Splunk Web uses the
  default value.

optInRemoteUI = <boolean>
* This setting maps to the Splunk Cloud "Automatic UI updates" menu item.
* Determines whether Splunk Cloud fetches configured UI pages from an external
  content delivery network or from the $SPLUNK_HOME directory.
* Default: false

allowExternalRemote = <boolean>
* Determines whether or not Splunk Web displays the "Automatic UI updates" menu item.
* Default: false



# Monitoring Console config
[smc]
remoteRoot = <string>
* The URL of the content delivery network that hosts the remote UI assets for Splunk Assist.
* If this setting has no value, the client uses a URL that points to an API that
  the Teleport Supervisor that runs on the local node exposes. This API
  serves the remote UI assets for Splunk Assist.
* Optional.
* Default: Not set.

#
# custom cherrypy endpoints
#

[endpoint:<python_module_name>]
* Registers a custom python CherryPy endpoint.
* The expected file must be located at:
  $SPLUNK_HOME/etc/apps/<APP_NAME>/appserver/controllers/<PYTHON_NODULE_NAME>.py
* This module's methods will be exposed at
  /custom/<APP_NAME>/<PYTHON_NODULE_NAME>/<METHOD_NAME>

#
# exposed splunkd REST endpoints
#
[expose:<unique_name>]
```

```
* Registers a splunkd-based endpoint that should be made available to the UI
  under the "/splunkd" and "/splunkd/__raw" hierarchies.
* The name of the stanza does not matter as long as it begins with "expose:"
* Each stanza name must be unique.

pattern = <url_pattern>
* The pattern to match under the splunkd /services hierarchy.
* For instance, "a/b/c" would match URIs "/services/a/b/c" and
  "/servicesNS/*/*/a/b/c",
* The pattern cannot include leading or trailing slashes.
* Inside the pattern an element of "*" matches a single path element.
  For example, "a/*/c" would match "a/b/c" but not "a/1/2/c".
* A path element of "**" matches any number of elements. For example,
  "a/**/c" would match both "a/1/c" and "a/1/2/3/c".
* A path element can end with a "*" to match a prefix. For example,
  "a/elem-*/b" would match "a/elem-123/c".

methods = <method_lists>
* A comma-separated list of methods to allow from the web browser
  (example: "GET,POST,DELETE").
* Default: "GET"

oidEnabled = [0 | 1]
* Whether or not a REST endpoint is capable of taking an embed-id as a
  query parameter.
* If set to 1, the endpoint is capable of taking an embed-id
  as a query parameter.
* This is only needed for some internal splunk endpoints, you probably
  should not specify this for app-supplied endpoints
* Default: 0

skipCSRFProtection = [0 | 1]
* Whether or not Splunk Web can safely post to an endpoint without applying
  Cross-Site Request Forgery (CSRF) protection.
* If set to 1, tells Splunk Web that it is safe to post to this endpoint
  without applying CSRF protection.
* This should only be set on the login endpoint (which already contains
  sufficient auth credentials to avoid CSRF problems).
* Default: 0

allowRemoteProxy = <boolean>
* Determines whether or not splunkd lets the exposed REST endpoint be proxied
  to remote nodes using the "remote-proxy" REST endpoint.
* If set to "true", splunkd will let requests be proxied to remote nodes
  through the "remote-proxy".
* If set to "false", splunkd will not let requests be proxied to remote nodes
  through the "remote-proxy".
* This setting only works for full URIs without wildcards.
* Default: false
```

web.conf.example

```
#   Version 9.2.2
#
# This is an example web.conf.  Use this file to configure data web
# settings.
#
# To use one or more of these configurations, copy the configuration block
# into web.conf in $SPLUNK_HOME/etc/system/local/. You must restart Splunk
```

```
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# This stanza heading must precede any changes.
[settings]

# Change the default port number:
httpport = 12800
# Also run the python application server on a non-default port:
appServerPorts = 12801

# Turn on SSL:
enableSplunkWebSSL = true
# absolute paths may be used here.
privKeyPath = /home/user/certs/myprivatekey.pem
serverCert = /home/user/certs/mycacert.pem
# NOTE: non-absolute paths are relative to $SPLUNK_HOME

# Allowing embedabble content in dashboards
# Embed tags will appear as is in the dashboard source
dashboard_html_allow_embeddable_content = true
dashboard_html_wrap_embed = false

# Allowing remote images from trusted hosts in simple XML dashboards
pdfgen_trusted_hosts = *.splunk.com, 192.0.2.0/24
```

web-features.conf

The following are the spec and example files for `web-features.conf`.

web-features.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of Splunk Web features used to configure
# Splunk Enterprise. You can use the settings to configure Splunk Web features.
# These features are replicated in a search head cluster environment.
#
# Each stanza controls a different web feature.
#
# For more information on configuration files, including precedence, search for
# "Use Splunk Web to manage configuration files" in the Admin Manual in the Splunk Docs.
```

[feature:search_v2_endpoint]

```
enable_search_v2_endpoint = <boolean>
* Determines whether Splunk Web uses the v2 search endpoint.
* A value of "true" means Splunk Web will use the v2 search endpoint.
* Default: true
```

[feature:quarantine_files]

```
enable_jQuery2 = <boolean>
* DEPRECATED.
* Determines whether or not Splunk Web can use jQuery 2 JavaScript files
  packaged with the Splunk platform.
* A "false" value means Splunk Web cannot use jQuery 2 JavaScript files
  packaged with the Splunk platform.
* CAUTION: Do not change this setting.
* Default: false

enable_unsupported_hotlinked_imports = <boolean>
* Determines whether or not Splunk Web can use unsupported JavaScript
  files that the Splunk platform will delete in a future release.
* Unsupported hotlinked imports are dependencies in your Simple XML Custom
  JavaScript Extensions that directly reference Splunk software.
* A "false" value means Splunk Web cannot use hotlinked imports
  that the Splunk platform will delete in a future release.
* CAUTION: Do not change this setting.
* Default: false
```

[feature:dashboards_csp]

```
enable_dashboards_external_content_restriction = <boolean>
* Whether or not Splunk Web restricts the loading of external content in Studio Dashboards or
  Classic Dashboards.
* A value of "true" means the following:
  * For Studio Dashboards, Splunk Web sets the Content-Security-Policy header, causing the
    browser to block images from external domains not included in the Dashboards Trusted
    Domains List (DTDL).
  * For Classic Dashboards, when the user loads a dashboard with external URLs not included
    in the DTDL, the user sees a warning modal. The user can decide to load the dashboard
    with external content or without external content.
* A value of "false" means the following:
  * For Studio Dashboards, Splunk Web does not set the Content-Security-Policy header. All
    external images load as usual and the browser does not block images.
  * For Classic Dashboards, all external content loads without warnings.
* Default: true

enable_dashboards_redirection_restriction = <boolean>
* Whether or not Splunk Web restricts redirecting to external content from Studio Dashboards or
  Classic Dashboards.
* A value of "true" means that the user sees a warning modal when redirecting to an external
  URL not included in the Dashboards Trusted Domains List. The user has the option to continue
  with the redirect or to cancel the redirect.
* A value of "false" means that nothing warns the user when redirecting to an external URL.
* Default: true

dashboards_trusted_domain.<name> = <string>
* A list of external domains that Splunk Web trusts for content loads and redirects. This list is
  called the Dashboards Trusted Domains List (DTDL).
* You must prefix each trusted domain on its own line with the string "dashboards_trusted_domain."
* The list has a maximum size of 6500 characters, after which any excess content will be ignored.
* If web-features.conf:'enable_dashboards_external_content_restriction' has a value of "true",
  then the following happens:
  * In Studio Dashboards, Splunk Web includes the DTDL in the Content-Security-Policy (CSP) page
    header.
    * The CSP header determines which domains Studio Dashboard can use to load images.
    * By default, 'self', data:, and blob: are added to the CSP header.
```

```
    * The browser prevents the loading of images from URLs not within the DTDL.
  * In Classic Dashboards, if the dashboard uses external URLs not included in the DTDL to load
    content, the user sees a warning modal.
* If web-features.conf:'enable_dashboards_external_content_restriction' has a value of "false" then
  the DTDL does not effect Dashboard loading and external content loads without warning.
* If web-features.conf:'enable_dashboards_redirection_restriction' has a value of "true", users
  see a warning modal when redirecting to an external URL not included in the DTDL.
* If web-features.conf:'enable_dashboards_redirection_restriction' has a value of "false" then the
  DTDL does not affect when a user redirects to an external URL, and no warning modal appears.
* Examples:
  * Only allow images from splunk.com and mozilla.org:
      dashboards_trusted_domain.endpoint1 = www.splunk.com
      dashboards_trusted_domain.endpoint2 = www.mozilla.org
  * Allow images from all external domains:
      dashboards_trusted_domain.endpoint1 = *
  * Only allow images starting with splunk.com/download/
      dashboards_trusted_domain.endpoint1 = www.splunk.com/download/
* Further documentation can be found by:
  * searching for "Content Security Policy" on the Mozilla Developer Network Docs website.
  * searching for and reading the Content Security Policy Quick Reference Guide.
* Default: Not set

internal.dashboards_trusted_domain.<name> = <string>
* A list of internal domains that Splunk Web trusts for content loading and redirection. When
  checking for URL trustworthiness, these domains combine with the Dashboards Trusted Domains
  List. Refer to web-features.conf:'dashboards_trusted_domain.<name>' for information on usage.
* Do not modify these values.
* Default: List of trusted Splunk Platform domains.
```

[feature:highcharts_accessibility]

```
disable_highcharts_accessibility = <boolean>
* Disable accessibility module in the highcharts charting library.
* DEPRECATED.
* A value of "true" means that Splunk Web will not use the accessibility module in the Highcharts
  charting library.
* CAUTION: Do not change this setting.
* Default: true
```

[feature:dashboard_studio]

```
activate_conversion_report = <boolean>
* Controls whether conversion related information is added to the XML of Studio Dashboards
* converted from Classic Dashboards.
* A value of "true" means that conversion information is added to Studio Dashboards.
* Do not modify this value.
* Default: true

enable_inputs_on_canvas = <boolean>
* Allows inputs directly on the canvas in Dashboard Studio.
* A value of "true" will allow inputs directly on the dashboard canvas in Dashboard Studio.
* Do not modify this value.
* Default: true

enable_show_hide = <boolean>
* Allows absolute "Show/Hide" panels in Dashboard Studio.
* A value of "true" will allow "Show/Hide" panels in the editor of Dashboard Studio.
* Do not modify this value.
* Default: true
```

```
enable_events_viz = <boolean>
* Allows "splunk.events" visualization type in Dashboard Studio.
* A value of "true" means the "splunk.events" visualization type is available in Dashboard Studio.
* Do not modify this value.
* Default: true

activate_workflow_actions_for_events_viz = <boolean>
* Allows workflow actions in the events visualization in Dashboard Studio.
* A value of "true" means that workflow actions will appear on the events visualization in Dashboard Studio.
* Do not modify this value.
* Default: true

activate_link_to_report = <boolean>
* Allows the Link to Report Interaction in Dashboard Studio.
* A value of "true" means the Link to Report Interaction is available in Dashboard Studio.
* Do not modify this value.
* Default: true

activate_link_to_search = <boolean>
* Allows the Link to Search Interaction in Dashboard Studio.
* A value of "true" means the Link to Search Interaction is available in Dashboard Studio.
* Do not modify this value.
* Default: true

activate_trellis_for_visualizations = <boolean>
* Allows trellis layout for supported visualizations in Dashboard Studio.
* A value of "true" means trellis layout is available for supported visualizations in Dashboard Studio.
* Do not modify this value.
* Default: true

activate_expanded_source_editor = <boolean>
* Uses a larger inline source editor for Dashboard Studio.
* A value of "true" means the expanded source editor is available in Dashboard Studio.
* Do not modify this value.
* Default: true

activate_dsl_webworkers_for_visualizations = <boolean>
* Uses WebWorkers for Dynamic Options Syntax execution to isolate from overall dashboard loading and
performance.
* A value of "true" means the WebWorkers are being used in Dashboard Studio.
* Do not modify this value.
* Default: false

activate_save_report_to_dashboard_studio = <boolean>
* Determines if users see an Add to Dashboard dropdown list in the Splunk Web Reports page and Save Search
to Report dialogs.
  The dropdown menu allows adding a report to a new or existing Dashboard Studio dashboard.
* A value of "false" means Splunk Web does not display the dropdown menu, and users can only add reports to
Classic Simple XML dashboards.
* Do not modify this value.
* Default: true

activate_source_mode_validation = <boolean>
# Determines whether the source mode validation in Dashboard Studio is activated.
# A value of "true" means that source mode is validated in Dashboard Studio.
# Do not modify this value.
* Default: true
```

[feature::windows_rce]

```
enable_acuif_pages = <boolean>
* Determines whether to display the new Admin Config UI Framework
  version of the following Windows input pages: admin_win-event-log-collections,
  admin_win-perfmon, admin_win-wmi-collections, fwd_admin_win-perfmon.
* A value of "true" means that Splunk Cloud Platform will display the
  Admin Config UI Framework version of the page.
* Default: false
```

[feature:page_migration]

```
enable_triggered_alerts_vnext = <boolean>
* Determines whether or not Splunk Web loads the new triggered alerts page.
* DEPRECATED.
* A value of "true" means that Splunk Web does load the new triggered alerts page.
* CAUTION: Do not change this setting.
* Default: true

enable_home_vnext = <boolean>
* Determines whether or not Splunk Web loads the new home page.
* DEPRECATED.
* A value of "true" means that Splunk Web does load the new home page.
* CAUTION: Do not change this setting.
* Default: true

enable_datasets_vnext = <boolean>
* Determines whether or not Splunk Web loads the new datasets page.
* DEPRECATED.
* A value of "true" means that Splunk Web does load the new datasets page.
* CAUTION: Do not change this setting.
* Default: true
```

[feature:dashboard_inputs_localization]

```
enable_dashboard_inputs_localization = <boolean>
* Determines whether or not Splunk Web will attempt to localize input choices in
  Classic dashboards.
* A value of "true" means that localization for input choices will be enabled in
  Classic Dashboards.
* A value of "false" means that localization for input choices will be disabled in
  Classic Dashboards.
* Default: false
```

[feature:share_job]

```
enable_share_job_control = <boolean>
* Determines whether or not users can share jobs using the "Share Job" button in
  the Search app in Splunk Web.
* A value of "true" means that users can use the "Share Job" button in the
  Search app to share search jobs.
* A value of "false" means that users cannot use the "Share Job" button to
  share search jobs. Instead, they receive a notice that job sharing has
  been disabled and they can instead share a search query.
* Default: true
```

[feature:search_auto_format]

```
enable_autoformatted_comments = <boolean>
* Determines whether or not comments are auto-formatted by the search editor's auto-formatter.
* DEPRECATED.
* CAUTION: Do not change this setting.
* A value of "false" means that comments are not auto-formatted. Comment auto-formatting may
* result in undesirable output.
* Default: false
```

[feature:ui_prefs_optimizations]

```
optimize_ui_prefs_performance = <boolean>
* Determines whether or not Splunk Web will optimize performance of the API related to ui-prefs.conf.
* DEPRECATED.
* CAUTION: Do not change this setting.
* A value of "false" means that Splunk Web will not optimize performance of the API related to ui-prefs.
* Default: true
```

[feature:splunk_web_optimizations]

```
enable_app_bar_performance_optimizations = <boolean>
* Determines whether or not Splunk Web will optimize performance when generating the app bar.
* DEPRECATED.
* CAUTION: Do not change this setting.
* A value of "false" means that Splunk Web will not optimize performance when generating the app bar.
* Default: true

bypass_app_bar_performance_optimizations_apps = <comma separated list>
* Splunk Web will not optimize performance when generating the app bar for this comma separated list of
apps.
* CAUTION: Do not change this setting.
* A value of "splunk_monitoring_console,search" means that Splunk Web will not optimize performance when
generating the app bar for the splunk_monitoring_console and search apps.
* Default: ""
```

[feature:spotlight_search]

```
enable_spotlight_search = <boolean>
* Determines whether Splunk Web displays the Spotlight Search bar in the
  Settings menu.
* A value of "true" means that Splunk Web will display the Spotlight Search
  bar in the Settings menu.
* Default: false
```

[feature:o11y_preview]

```
enable_o11y_preview = <boolean>
* Determines whether Splunk Web displays the preview links and
  Splunk Observability preview sidebar in Search & Reporting.
* A value of "true" means that Splunk Web will show preview links and
  Splunk Observability preview sidebar in Search & Reporting.
* Default: true
```

web-features.conf.example

```
#   Version 9.2.2
#
# You can configure Splunk Web features for your custom application.
#
# To use one or more of these configurations, copy the configuration block into
# the web-features.conf file located in $SPLUNK_HOME/etc/system/local/. You must restart
# Splunk software after you make changes to this setting to enable configurations.
#
# For more information on configuration files, including precedence, search for
# "Use Splunk Web to manage configuration files" in the Admin Manual in the Splunk Docs.

[feature:search_v2_endpoint]
enable_search_v2_endpoint = true

[feature:quarantine_files]
enable_jQuery2 = false
enable_unsupported_hotlinked_imports = false

[feature:dashboards_csp]
enable_dashboards_external_content_restriction = true
enable_dashboards_redirection_restriction = true
dashboards_trusted_domain.splunk = *.splunk.com
dashboards_trusted_domain.example = www.example.com




[feature:page_migration]
enable_triggered_alerts_vnext = false
enable_home_vnext = false
enable_datasets_vnext = false

[feature:dashboard_studio]
enable_inputs_on_canvas = true
enable_show_hide = true

[feature:dashboard_inputs_localization]
enable_dashboard_inputs_localization = false

[feature:share_job]
enable_share_job_control = true

[feature:search_auto_format]
enable_autoformatted_comments = false

[feature:ui_prefs_optimizations]
optimize_ui_prefs_performance = true

[feature:splunk_web_optimizations]
enable_app_bar_performance_optimizations = true
bypass_app_bar_performance_optimizations_apps = "search"

[feature:spotlight_search]
enable_spotlight_search = false

[feature:o11y_preview]
enable_o11y_preview = true
```

wmi.conf

The following are the spec and example files for `wmi.conf`.

wmi.conf.spec

```
#   Version 9.2.2
#
# This file contains possible setting/value pairs for configuring Windows
# Management Instrumentation (WMI) access from Splunk Enterprise.
#
# There is a wmi.conf in $SPLUNK_HOME\etc\system\default\.  To set custom
# configurations, place a wmi.conf in $SPLUNK_HOME\etc\system\local\. For
# examples, see wmi.conf.example.
#
# You must restart Splunk Enterprise to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

############################################################################
#----GLOBAL SETTINGS-----
 ############################################################################

[settings]
* Specifies parameters for the WMI input.
* The entire stanza and every parameter within it is optional.
* If the stanza is missing, Splunk Enterprise assumes system defaults.

initial_backoff = <integer>
* How long, in seconds, to wait before retrying the connection to
  the WMI provider after the first connection error.
* If connection errors continue, the wait time doubles until it reaches
  the integer specified in 'max_backoff'.
* Default: 5

max_backoff = <integer>
* How long, in seconds, to attempt to reconnect to the
  WMI provider.
* Default: 20

max_retries_at_max_backoff = <integer>
* When the WMI input has connection errors to the WMI provider, it
  backs off connection attempts by doubling the amount of time it
  waits between connection attempts. It modifies attempts from an initial interval of
  'initial_backoff' seconds to an interval specified 'max_backoff'
  seconds.
* After the input has waited 'max_backoff' seconds between connection
  attempts, and while connection errors persist, this setting tells
  the input how many times it should continue trying to connect at
  the 'max_backoff' interval.
* If reconnection to the WMI provider fails after 'max_retries' attempts,
  the input gives up and does not attempt further connections until
  you restart Splunk Enterprise.
* Default: 2

checkpoint_sync_interval = <integer>
* How long, in seconds, to wait for state data (event log checkpoint)
```

```
  to be written to disk.
* Default: 2
```

INPUT-SPECIFIC SETTINGS-----

```
[WMI:<name>]
* There are two types of WMI input stanza:
  * Event log stanza: Used to collect Windows Event Logs. You must configure the
    'event_log_file' setting.
  * Windows Query Language (WQL): Used to issue raw Windows Query Language (WQL)
    requests. You must configure the 'wql' setting.
* Do not use both the 'event_log_file' amd 'wql' attributes. Use one or the other.

server = <comma-separated strings>
* A comma-separated list of WMI providers (Windows machines) from which to get data.
* Default: the local machine

interval = <integer>
* How often, in seconds, to poll the WMI provider for new data.
* You must supply this setting. No default is supplied and the input does not run if the setting is
  not specified.
* No default.

disabled = <boolean>
* Whether or not the input is enabled.
* Set to 1 to disable the input, 0 to enable it.
* Default: 0 (enabled).

hostname = <string>
* All results generated by this stanza will appear to have arrived from
  the string you specify here.
* This setting is optional.
* Default: input detects the host automatically

current_only = <boolean>
* Changes the characteristics and interaction of WMI-based event
  collections.
* When you set 'current_only' to 1:
  * For event log stanzas, captures events that occur
    only while Splunk Enterprise is running.
  * For WQL stanzas, the input expects event notification queries. The
    WMI class you query must support sending events. Failure to supply
    the correct event notification query structure causes WMI to return
    a syntax error to the input.
  * An example event notification query that watches for process creation:
    * SELECT * FROM __InstanceCreationEvent WITHIN 1 WHERE
      TargetInstance ISA 'Win32_Process'.
* When you set 'current_only' to 0:
  * For event log stanzas, Splunk Enterprise gathers all the events from
    the checkpoint. If there is no checkpoint, Splunk Enterprise retrieves
    all events starting from the oldest.
  * For WQL stanzas, Splunk Enterprise executes the query and retrieves
    the results. The query is a non-notification query.
  * For example
    * Select * Win32_Process where caption = "explorer.exe"
* Default: 0

use_old_eventlog_api = <boolean>
* Whether or not to read Event Log events with the Event Logging API rather
```

```
  than the Windows Event Log API.
* This is an advanced setting. Contact Splunk Support before you change it.
* If set to "true", the input uses the Event Logging API (instead of the Windows Event Log API)
  to read from the Event Log on Windows Server 2008, Windows Vista, and later installations.
* Default: false (Use the API that is specific to the OS.)

use_threads = <integer>
* The number of threads, in addition to the default writer thread, that can
  be created to filter events with the deny list or allow list regular expression.
* This is an advanced setting. Contact Splunk Support before you change it.
* The maximum number of threads is 15.
* Default: 0

thread_wait_time_msec = <integer>
* The interval, in milliseconds, between attempts to re-read Event Log files when a read error occurs.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: 5000

suppress_checkpoint = <boolean>
* Whether or not the Event Log strictly follows the 'checkpointInterval' setting when it saves a checkpoint.
* By default, the Event Log input saves a checkpoint from between zero and 'checkpointInterval' seconds,
  depending on incoming event volume.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: false

suppress_sourcename = <boolean>
* Whether or not to exclude the 'sourcename' field from events.
* When set to "true", the input excludes the 'sourcename' field from events and throughput performance
  (the number of events processed per second) improves.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: false

suppress_keywords = <boolean>
* Whether or not to exclude the 'keywords' field from events.
* When set to "true", the input excludes the 'keywords' field from events and throughput performance
  (the number of events processed per second) improves.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: false

suppress_type = <boolean>
* Whether or not to exclude the 'type' field from events.
* When set to true, the input excludes the 'type' field from events and throughput performance
  (the number of events processed per second) improves.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: false

suppress_task = <boolean>
* Whether or not to exclude the 'task' field from events.
* When set to "true", the input excludes the 'task' field from events and thruput performance
  (the number of events processed per second) improves.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: false

suppress_opcode = <boolean>
* Whether or not to exclude the 'opcode' field from events.
* When set to "true", the input excludes the 'opcode' field from events and throughput performance
  (the number of events processed per second) improves.
* This is an advanced setting. Contact Splunk Support before you change it.
* Default: false

batch_size = <integer>
* Number of events to fetch on each query.
```

```
* Default: 10

checkpointInterval = <integer>
* How often, in seconds, that the Windows Event Log input saves a checkpoint.
* Checkpoints store the event ID of acquired events. This lets the input
  continue monitoring at the correct event after a shutdown or outage.
* Default: 0

index = <string>
* Specifies the index that this input should send the data to.
* This setting is optional.
* When you define 'index', the input prepends "index=" to <string>.
* Default: "index=main" (or whatever you have set as your default index).
```

Event log-specific attributes:

```
event_log_file = <string> <Application, System, etc>
* Tells the input to expect event log data for this stanza, and specifies
  the event log channels you want the input to monitor.
* To specify Event Log sources, use this setting instead of WQL.
* Specify one or more event log channels to poll. You must separate multiple
  Event Log channels with commas.
  * For exmaple, to include the Application and System channels, specify "Application, System".
* No default.

disable_hostname_normalization = <boolean>
* Whether or not the WMI input normalizes hostnames from 'localhost' to
  what is present in the %COMPUTERNAME% Windows system variable.
* If set to "true", hostname normalization is disabled.
* If set to "false" or not set, the input converts the hostname for
  'localhost' to %COMPUTERNAME%.
* 'localhost' refers to the following list of strings:
  * localhost
  * 127.0.0.1
  * ::1
  * the name of the DNS domain for the local computer
  * the fully qualified DNS name
  * the NetBIOS name
  * the DNS host name of the local computer
```

WQL-specific attributes:

```
wql = <string>
* Configures the WMI input to expect data from a WMI provider for this stanza, and
  specifies the Windows Query Language query you want the input to make to
  gather that data.
* Use this if you are not using the 'event_log_file' setting.
* Ensure that your WQL queries have the correct syntax and structure when you
  use this option.
  * For example,
    SELECT * FROM Win32_PerfFormattedData_PerfProc_Process WHERE Name = "splunkd".
* If you want to use event notification queries, you must also set the
  "current_only" attribute to "1" within the stanza, and your query must be
  appropriately structured for event notification (meaning its WQL string
  must contain one or more of the GROUP, WITHIN or HAVING clauses.)
```

```
  * For example,
    SELECT * FROM __InstanceCreationEvent WITHIN 1 WHERE TargetInstance ISA
    'Win32_Process'
* No default.

namespace = <string>
* The namespace where the WMI provider resides.
* The namespace specification can either be relative (root\cimv2) or absolute
  (\\server\root\cimv2).
* If the server attribute is present, you cannot specify an absolute
  namespace.
* Default: root\cimv2.
```

wmi.conf.example

```
#   Version 9.2.2
#
# This is an example wmi.conf.  These settings are used to control inputs
# from WMI providers. Refer to wmi.conf.spec and the documentation at
# splunk.com for more information about this file.
#
# To use one or more of these configurations, copy the configuration block
# into wmi.conf in $SPLUNK_HOME\etc\system\local\.  You must restart Splunk
# to enable configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# This stanza specifies runtime parameters.

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

# Pull events from the Application, System and Security event logs from the
# local system every 10 seconds. Store the events in the "wmi_eventlog"
# Splunk index.

[WMI:LocalApplication]
interval = 10
event_log_file = Application
disabled = 0
index = wmi_eventlog

[WMI:LocalSystem]
interval = 10
event_log_file = System
disabled = 0
index = wmi_eventlog

[WMI:LocalSecurity]
interval = 10
event_log_file = Security
disabled = 0
index = wmi_eventlog
```

```
# 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

# Collect all process-related performance metrics for the splunkd process,
# every second.  Store those events in the "wmi_perfmon" index.
[WMI:LocalSplunkdProcess]
interval = 1
wql = select * from Win32_PerfFormattedData_PerfProc_Process where Name = "splunkd"
disabled = 0
index = wmi_perfmon

# Listen from three event log channels, capturing log events that occur only
# while Splunk is running, every 10 seconds.  Gather data from three remote
# servers srv1, srv2 and srv3.

[WMI:TailApplicationLogs]
interval = 10
event_log_file = Application, Security, System
server = srv1, srv2, srv3
disabled = 0
current_only = 1
batch_size = 10

# Listen for process-creation events on a remote machine, once a second.

[WMI:ProcessCreation]
interval = 1
server = remote-machine
wql = select * from __InstanceCreationEvent within 1 where TargetInstance isa 'Win32_Process'
disabled = 0
current_only = 1
batch_size = 10

# Receive events whenever someone connects or removes a USB device on
# the computer, once a second.

[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
batch_size = 10
```

workflow_actions.conf

The following are the spec and example files for `workflow_actions.conf`.

workflow_actions.conf.spec

```
#   Version 9.2.2
#
# This file contains possible attribute/value pairs for configuring workflow
# actions in Splunk.
#
# There is a workflow_actions.conf in $SPLUNK_HOME/etc/apps/search/default/.
# To set custom configurations, place a workflow_actions.conf in either
# $SPLUNK_HOME/etc/system/local/ or add a workflow_actions.conf file to your
# app's local/ directory. For examples, see workflow_actions.conf.example.
# You must restart Splunk to enable configurations, unless editing them
# through the Splunk manager.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#  * You can also define global settings outside of any stanza, at the top
#    of the file.
#  * Each conf file should have at most one default stanza. If there are
#    multiple default stanzas, attributes are combined. In the case of
#    multiple definitions of the same attribute, the last definition in the
#    file wins.
#  * If an attribute is defined at both the global level and in a specific
#    stanza, the value in the specific stanza takes precedence.

###############################################################################
# General required settings:
# These apply to all workflow action types.
###############################################################################

type = <string>
* The type of the workflow action.
* If not set, the Splunk platform skips this workflow action.

label = <string>
* The label to display in the workflow action menu.
* If not set, the Splunk platform skips this workflow action.

###############################################################################
# General optional settings:
# These settings are not required but are available for all workflow
# actions.
###############################################################################

fields = <comma or space separated list>
* The fields required to be present on the event in order for the workflow
  action to be applied.
```

```
* When "display_location" is set to "both" or "field_menu", the workflow
  action will be applied to the menu's corresponding to the specified
  fields.
* If fields is undefined or set to *, the workflow action is applied to all
  field menus.
* If the * character is used in a field name, it is assumed to act as a
  "globber". For example host* would match the fields hostname, hostip, etc.
* Acceptable values are any valid field name, any field name including the *
  character, or * (e.g. *_ip).
* Default: *

eventtypes = <comma or space separated list>
* The eventtypes required to be present on the event in order for the
  workflow action to be applied.
* Acceptable values are any valid eventtype name, or any eventtype name plus
  the * character (e.g. host*).

display_location = <string>
* Dictates whether to display the workflow action in the event menu, the
  field menus or in both locations.
* Accepts field_menu, event_menu, or both.
* Default: both.

disabled = [True | False]
* Dictates whether the workflow action is currently disabled
* Default: False
```

Using field names to insert values into workflow action settings

```
# Several settings detailed below allow for the substitution of field values
# using a special variable syntax, where the field's name is enclosed in
# dollar signs.  For example, $_raw$, $hostip$, etc.
#
# The settings, label, link.uri, link.postargs, and search.search_string all
# accept the value of any valid field to be substituted into the final
# string.
#
# For example, you might construct a Google search using an error message
# field called error_msg like so:
# link.uri = http://www.google.com/search?q=$error_msg$.
#
# Some special variables exist to make constructing the settings simpler.

$@field_name$
* Allows for the name of the current field being clicked on to be used in a
  field action.
* Useful when constructing searches or links that apply to all fields.
* NOT AVAILABLE FOR EVENT MENUS

$@field_value$
* Allows for the value of the current field being clicked on to be used in a
  field action.
* Useful when constructing searches or links that apply to all fields.
* NOT AVAILABLE FOR EVENT MENUS

$@sid$
* The sid of the current search job.

$@offset$
```

```
* The offset of the event being clicked on in the list of search events.

$@namespace$
* The name of the application from which the search was run.

$@latest_time$
* The latest time the event occurred.  This is used to disambiguate similar
  events from one another. It is not often available for all fields.
```

Field action types

```
################################################################################
# Link type:
# Allows for the construction of GET and POST requests via links to external
# resources.
################################################################################

link.uri = <string>
* The URI for the resource to link to.
* Accepts field values in the form $<field name>$, (e.g $_raw$).
* All inserted values are URI encoded.
* Required

link.target = <string>
* Determines if clicking the link opens a new window, or redirects the
  current window to the resource defined in link.uri.
* Accepts: "blank" (opens a new window), "self" (opens in the same window)
* Default: "blank"

link.method = <string>
* Determines if clicking the link should generate a GET request or a POST
  request to the resource defined in link.uri.
* Accepts: "get" or "post".
* Default: "get".

link.postargs.<int>.<key/value> = <value>
* Only available when link.method = post.
* Defined as a list of key / value pairs like such that foo=bar becomes:
  link.postargs.1.key = "foo"
  link.postargs.1.value = "bar"
* Allows for a conf compatible method of defining multiple identical keys (e.g.):
  link.postargs.1.key = "foo"
  link.postargs.1.value = "bar"
  link.postargs.2.key = "foo"
  link.postargs.2.value = "boo"
  ...
* All values are html form encoded appropriately.

################################################################################
# Search type:
# Allows for the construction of a new search to run in a specified view.
################################################################################

search.search_string = <string>
* The search string to construct.
* Accepts field values in the form $<field name>$, (e.g. $_raw$).
* Does NOT attempt to determine if the inserted field values may break
  quoting or other search language escaping.
```

```
* Required

search.app = <string>
* The name of the Splunk application in which to perform the constructed
  search.
* By default this is set to the current app.

search.view = <string>
* The name of the view in which to preform the constructed search.
* By default this is set to the current view.

search.target = <string>
* Accepts: blank, self.
* Works in the same way as link.target. See link.target for more info.

search.earliest = <time>
* Accepts absolute and relative times (e.g. -10h).
* Determines the earliest time to search from.

search.latest = <time>
* Accepts absolute and relative times (e.g. -10h).
* Determines the latest time to search to.

search.preserve_timerange = <boolean>
* Ignored if either the search.earliest or search.latest values are set.
* When true, the time range from the original search which produced the
  events list will be used.
* Default: false.
```

workflow_actions.conf.example

```
#   Version 9.2.2
#
# This is an example workflow_actions.conf.  These settings are used to
# create workflow actions accessible in an event viewer.  Refer to
# workflow_actions.conf.spec and the documentation at splunk.com for more
# information about this file.
#
# To use one or more of these configurations, copy the configuration block
# into workflow_actions.conf in $SPLUNK_HOME/etc/system/local/, or into your
# application's local/ folder.  You must restart Splunk to enable
# configurations.
#
# To learn more about configuration files (including precedence) please see
# the documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles

# These are the default workflow actions and make extensive use of the
# special parameters: $@namespace$, $@sid$, etc.

[show_source]
type=link
fields = _cd, source, host, index
display_location = event_menu
label = Show Source
link.uri = /app/$@namespace$/show_source?sid=$@sid$&offset=$@offset$&latest_time=$@latest_time$

[ifx]
type = link
```

```
display_location = event_menu
label = Extract Fields
link.uri = /ifx?sid=$@sid$&offset=$@offset$&namespace=$@namespace$

[etb]
type = link
display_location = event_menu
label = Build Eventtype
link.uri = /etb?sid=$@sid$&offset=$@offset$&namespace=$@namespace$

# This is an example workflow action which will be displayed in a specific
# field menu (clientip).

[whois]
display_location = field_menu
fields = clientip
label = Whois: $clientip$
link.method = get
link.target = blank
link.uri = http://ws.arin.net/whois/?queryinput=$clientip$
type = link

# This is an example field action which will allow a user to search every
# field value in Google.

[Google]
display_location = field_menu
fields = *
label = Google $@field_name$
link.method = get
link.uri = http://www.google.com/search?q=$@field_value$
type = link

# This is an example post link that will send its field name and field value
# to a  fictional bug tracking system.

[Create JIRA issue]
display_location = field_menu
fields = error_msg
label = Create JIRA issue for $error_class$
link.method = post
link.postargs.1.key = error
link.postargs.1.value = $error_msg$
link.target = blank
link.uri = http://127.0.0.1:8000/jira/issue/create
type = link

# This is an example search workflow action that will be displayed in an
# event's menu, but requires the field "controller" to exist in the event in
# order for the workflow action to be available for that event.

[Controller req over time]
display_location = event_menu
fields = controller
label = Requests over last day for $controller$
search.earliest = -3d
search.search_string = sourcetype=rails_app controller=$controller$ | timechart span=1h count
search.target = blank
search.view = charting
type = search
```

workload_policy.conf

The following are the spec and example files for `workload_policy.conf`.

workload_policy.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure search admission control for splunk.
#
# There is a workload_policy.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name workload_policy.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see workload_policy.conf.example. You may need to restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
# Settings to configure search admission control, including enabling/disabling feature
# and other configurations.
```

[search_admission_control]

```
admission_rules_enabled = <bool>
* Determines whether admission rules are applied to searches.
* If set to true, admission rules for pre-filtering searches are applied when a search
  is dispatched.
* Default: 0
```

workload_policy.conf.example

```
# Enable the admission rules defined in workload_rules.conf.
[search_admission_control]
admission_rules_enabled = 1
```

workload_pools.conf

The following are the spec and example files for `workload_pools.conf`.

workload_pools.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure workloads for splunk.
#
# There is a workload_pools.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name workload_pools.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see workload_pools.conf.example. You may need to restart the Splunk instance
# to enable configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file takes precedence.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
#
# CAUTION: Do not alter the settings in the workload_pools.conf file unless you know
#     what you are doing.  Improperly configured worloads might result in
#     splunkd crashes, memory overuse, or both.
```

[general]

```
enabled = <bool>
* Specifies whether workload management has been enabled on the system or not.
* This setting only applies to the default stanza as a global setting.
* Default: false

default_pool = <string>
* Specifies the default workload pool to be used at runtime for search workloads.
* This setting is maintained for backward compatibility with previous releases.
  Its value is set but is not used in the current release. This value matches the
  default_pool value of [workload_category:search].
```

```
* This setting is only applicable when workload management has been enabled in
  the system. If workload management has been enabled, this is a mandatory setting.

ingest_pool = <string>
* Specifies the workload pool for splunkd and helper processes that control
  data ingestion and related actions in the Splunk deployment.
* This setting is maintained for backward compatibility with previous releases.
  Its value is set but is not used in the current release. This value matches the
  default_pool value of [workload_category:ingest].
* This setting is only applicable when workload management has been enabled in
  the system. If workload management has been enabled, this is a mandatory setting.

workload_pool_base_dir_name = <string>
* Specifies the base controller directory name for Splunk cgroups on Linux that is
  used by a Splunk deployment.
* Workload pools created from the workload management page are all created relative
  to this base directory.
* This setting is only applicable when workload management has been enabled in
  the system. If workload management has been enabled, this is a mandatory setting.
* Default: splunk
```

[workload_pool:<pool_name>]

```
cpu_weight = <number>
* Specifies the cpu weight to be used by this workload pool.
* This is a percentage of the total cpu resources available to the category to
  which the pool belongs.
* Default: not set

mem_weight = <number>
* Specifies the memory weight to be used by this workload pool.
* This is a percentage of the total memory resources available to the category to
  which the pool belongs.
* This is a mandatory parameter for the creation of a workload pool and only
  allows positive integral values.
* Default: not set

category = <string>
* Specifies the category to which this workload pool belongs.
* Required to create a workload pool.
* Valid categories are "search","misc" and "ingest".
* The "ingest" and "misc" categories each contain one pool only, which is the
  default_pool for the respective category.
* Default: not set

default_category_pool = <boolean>
* Specifies if this pool is the default pool for its category.
* Admin users can specify workload pools associated with roles. If no workload
  pool is found, the default_pool defined for this category is used.
* The first pool that is added to a category has this value set to 1.
* All other pools have this value set to 0.
* Required if workload management is enabled.
* Default: false
```

[workload_category:<category>]

```
* Specifies the resource allocation for workload pools in this category.
  The <category> value can be "search","ingest' or "misc".
cpu_weight = <number>
```

```
* Specifies the cpu weight to be used by this category.
* This is a percentage of the total cpu resources available to all categories.
* This parameter exists in the default configuration and is editable with values
  that are positive integer values less than 100.
* Default is set.

mem_weight = <number>
* Specifies the memory weight to be used by this category.
* This is a percentage of the total memory resources available to all categories.
* This parameter exists in the default configuration and is editable with values
  that are positive integer values less than 100.
* Default is set.
```

workload_pools.conf.example

```
#   Version 9.2.2
# CAUTION: Do not alter the settings in workload_pools.conf unless you know what you are doing.
# Improperly configured workloads may result in splunkd crashes and/or memory overuse.

[general]
enabled = false
default_pool = pool_1
ingest_pool = pool_2
workload_pool_base_dir_name = splunk

[workload_category:search]
cpu_weight = 70
mem_weight = 70

[workload_category:ingest]
cpu_weight = 20
mem_weight = 20

[workload_category:misc]
cpu_weight = 10
mem_weight = 10

[workload_pool:pool_1]
cpu_weight = 40
mem_weight = 40
category = search
default_category_pool = 1

[workload_pool:pool_2]
cpu_weight = 30
mem_weight = 30
category = ingest
default_category_pool = 1

[workload_pool:pool_3]
cpu_weight = 20
mem_weight = 20
category = misc
default_category_pool = 1

[workload_pool:pool_4]
cpu_weight = 10
mem_weight = 10
category = search
```

```
default_category_pool = 0
```

workload_rules.conf

The following are the spec and example files for `workload_rules.conf`.

workload_rules.conf.spec

```
#   Version 9.2.2
#
```

OVERVIEW

```
# This file contains descriptions of the settings that you can use to
# configure workloads classification rules for splunk.
#
# There is a workload_rules.conf file in the $SPLUNK_HOME/etc/system/default/ directory.
# Never change or copy the configuration files in the default directory.
# The files in the default directory must remain intact and in their original
# location.
#
# To set custom configurations, create a new file with the name workload_rules.conf in
# the $SPLUNK_HOME/etc/system/local/ directory. Then add the specific settings
# that you want to customize to the local configuration file.
# For examples, see workload_rules.conf.example. You do not need to restart the Splunk instance
# to enable workload_rules.conf configuration changes.
#
# To learn more about configuration files (including file precedence) see the
# documentation located at
# http://docs.splunk.com/Documentation/Splunk/latest/Admin/Aboutconfigurationfiles
#
```

GLOBAL SETTINGS

```
# Use the [default] stanza to define any global settings.
#   * You can also define global settings outside of any stanza, at the top of
#     the file.
#   * Each .conf file should have at most one default stanza. If there are
#     multiple default stanzas, settings are combined. In the case of
#     multiple definitions of the same setting, the last definition in the
#     file takes precedence.
#   * If a setting is defined at both the global level and in a specific
#     stanza, the value in the specific stanza takes precedence.
#
# CAUTION: Do not alter the settings in the workload_rules.conf file unless you know
#     what you are doing.  Improperly configured workload rules might result in
#     splunkd crashes, memory overuse, or both.
```

[general]

```
numeric_search_time_range = <bool>
* Specifies whether the search_time_range predicate accepts numerical values.
```

```
* A value of "true" means search_time_range accepts numerical values.
* Allows assigning numerical values to the search_time_range predicate when
  defining workload rules and admission rules. For example,
  "search_time_range>7d" or "search_time_range<=24h".
* The search_time_range predicate accepts the value "alltime" regardless of
  the value of this setting.
* This setting applies only to the default stanza as a global setting.
* Note: This setting can cause slower search performance.
* Default: false
```

[workload_rule:<rule_name>]

```
predicate = <string>
* Specifies the predicate of this workload classification rule.
* The format is logical expression with predicate as <type>=<value>.
* For example, "app=search AND (NOT role=power)".
* The valid <type> are "app", "role", "user", "index",
  "search_type", "search_mode", "search_time_range", and "runtime".
  The <value> is the exact value of the <type>.
* For "app" type, the value is the name of the app. For example, "app=search".
* For "role" type, the value is the name of the role. For example, "role=admin".
* For "index" type, the value is the name of the index. For example,
  "index=_internal". Note that the value can refer to an internal or public index.
* For "user" type, the value is the name of any valid user. For example,
  "user=bob". Note that the reserved internal user "noboby" is invalid; the
  reserved internal user "splunk-system-user" is valid.
* For "search_type" type, the value is the type of the search. Valid search
  types include "adhoc", "scheduled", "datamodel_acceleration",
  "report_acceleration" and "summary_index".
* For "search_mode" type, the value is the mode of the search. Valid modes
  include "realtime" and "historical".
* For "search_time_range" type, the value is the time range of the search.
  Value can be "alltime" or a numerical value. For example,
  "search_time_range>7d" or "search_time_range<=24h".
  To specify a numerical value, the 'numerical_search_time_range' flag
  must be set to "true".

* For "runtime" type, the value is the amount of time a search must run in a
  workload pool to trigger a specified action, such as alert, move or abort.
  Valid units for runtime values include s, second, seconds, m, minute, minutes,
  and h, hour, hours.
* Required.

workload_pool = <string>
* Specifies the name of the workload pool, for example "pool1".
* The pool name that you specify must already be defined in the
  [workload_pool:<pool_name>] stanza in workload_pools.conf.

action = alert | move | abort
* Specifies the action to take when a search exceeds the specified runtime value.
* The action "alert" sends a notification message to Splunk Web that indicates
  the runtime of the search.
* The action "move" moves the search from the original workload pool to a
  designated alternate workload pool, and sends a notification message to
  Splunk Web.
* The action "abort" kills the search, and sends a notification message to
  Splunk Web.
* Optional.

schedule = always_on | time_range | every_day | every_week | every_month
* Specifies whether the rule is always on or has a valid time range that
```

```
  expires.
* Optional. If it's empty, it means the rule is always on.

start_time = <string>
* This setting is required when 'schedule' is set to: "time_range",
 "every_week", "every_month", or "every_day".
* The time format for 'start_time' is HH:00.
* If 'schedule' is set to "time_range", the 'start_time' specifies the
 exact time that the valid time range starts, including 'start_date', 'end_date',
  time, and time zone.
* If 'schedule' is set to "every_week" or "every_month", the 'start_time'
 specifies the start hour.
* If 'schedule' is set to "every_day", the 'start_time' is set to 0.
* Default 0.

end_time = <string>
* This setting is required when 'schedule' is set to: "time_range",
 "every_week", "every_month", or "every_day".
* The time format for 'end_time' is HH:00.
* If 'schedule' is set to "time_range", the 'end_time' specifies the
 exact time that the valid time range ends, including 'start_date', 'end_date',
  time, and time zone.
* If 'schedule' is set to "every_week" or "every_month", the 'end_time'
 specifies the end hour.
* If 'schedule' is set to "every_day", the 'end_time' is set to 0.
* Default 0.

every_week_days = <string>
* This setting is required when 'schedule' is set to "every_week".
* Specifies recurring days of the week.
* Supports comma separated numbers from 0 to 6, where 0 represents
 Sunday.
* No default.

every_month_days = <string>
* This setting is required when 'schedule' is set to "every_month".
* Specifies recurring days of the month.
* Supports comma separated numbers from 1 to 31, where 1 represents
 the 1st day of the month.
* No default.

start_date = <string>
* This setting is required when 'schedule' is set to "time_range".
* The date format is YYYY-MM-DD.
* Default (in SplunkWeb): the current date.
* Default (manual configuration): none.

end_date = <string>
* This setting is required when 'schedule' is set to "time_range".
* The date format is YYYY-MM-DD
* Default (in SplunkWeb): the current date.
* Default (manual configuration): none.

user_message = <string>
* Specifies the message shown in the search job inspector if the rule is
  applied to a search.
* Cannot exceed 140 characters.
* Optional.

disabled = <boolean>
* Toggles a workload rule off and on.
* Set to "true" to disable a rule.
```

```
* Default: false
```

[workload_rules_order]

```
rules = <string>
* List of all workload classification rules.
* The format of the "string" is comma separated items, "rule1,rule2,...".
* The rules listed are defined in [workload_rule:<rule_name>] stanza.
* The order of the rule name in the list determines the priorities of that rule.
  For example, in "rule1,rule2", rule1 has higher priority than rule2.
* The default value for this property is empty, meaning there is no rule defined.
```

[search_filter_rule:<rule_name>]

```
predicate = <string>
* Specifies the predicate of this workload classification rule.
* The format is logical expression with predicate as <type>=<value>.
* For example, "app=search AND (NOT role=power)".
* The valid <type> are "app", "role", "user", "index",
  "search_type", "search_mode", "search_time_range", and "adhoc_search_percentage".
  The <value> is the exact value of the <type>.
* For "app" type, the value is the name of the app. For example, "app=search".
* For "role" type, the value is the name of the role. For example, "role=admin".
* For "index" type, the value is the name of the index. For example,
  "index=_internal". Note that the value can refer to an internal or public index.
* For "user" type, the value is the name of any valid user. For example,
  "user=bob". Note that the reserved internal user "noboby" is invalid; the
  reserved internal user "splunk-system-user" is valid.
* For "search_type" type, the value is the type of the search. Valid search
  types include "adhoc", "scheduled", "datamodel_acceleration",
  "report_acceleration" and "summary_index".
* For "search_mode" type, the value is the mode of the search. Valid modes
  include "realtime" and "historical".
* For "search_time_range" type, the value is the time range of the search.
  For now, value can only be "alltime".
* For "adhoc_search_percentage" type, the value is an integer in the range [0,100]
  indicating the percentage of total concurrent searches that adhoc searches can
  consume before being filtered or queued. If specified, predicate must also include
  "search_type=adhoc".
* Required.

action = filter | queue
* Specifies the action to take when a search meets the rule criteria.
* The action "filter" is defined for search filter rules. If a search meets the rule
  criteria, the search is not executed.
* The action "queue" is only defined for search filter rules with "adhoc_search_percentage"
  specified in the predicate. If an ad hoc search meets the rule criteria, it will be
  queued and attempted later. A search meeting criteria for both "filter" and "queue"
  actions will be filtered.
* Required.

schedule = always_on | time_range | every_day | every_week | every_month
* Specifies whether the rule is always on or has a valid time range that
  expires.
* Optional. If it's empty, it means the rule is always on.

start_time = <string>
* This setting is required when 'schedule' is set to: "time_range",
 "every_week", "every_month", or "every_day".
```

```
* The time format for 'start_time' is HH:00.
* If 'schedule' is set to "time_range", the 'start_time' specifies the
 exact time that the valid time range starts, including 'start_date', 'end_date',
  time, and time zone.
* If 'schedule' is set to "every_week" or "every_month", the 'start_time'
 specifies the start hour.
* If 'schedule' is set to "every_day", the 'start_time' is set to 0.
* Default 0.

end_time = <string>
* This setting is required when 'schedule' is set to: "time_range",
 "every_week", "every_month", or "every_day".
* The time format for 'end_time' is HH:00.
* If 'schedule' is set to "time_range", the 'end_time' specifies the
 exact time that the valid time range ends, including 'start_date', 'end_date',
  time, and time zone.
* If 'schedule' is set to "every_week" or "every_month", the 'end_time'
 specifies the end hour.
* If 'schedule' is set to "every_day", the 'end_time' is set to 0.
* Default 0.

every_week_days = <string>
* This setting is required when 'schedule' is set to "every_week".
* Specifies recurring days of the week.
* Supports comma separated numbers from 0 to 6, where 0 represents
 Sunday.
* No default.

every_month_days = <string>
* This setting is required when 'schedule' is set to "every_month".
* Specifies recurring days of the month.
* Supports comma separated numbers from 1 to 31, where 1 represents
 the 1st day of the month.
* No default.

start_date = <string>
* This setting is required when 'schedule' is set to "time_range".
* The date format is YYYY-MM-DD.
* Default (in SplunkWeb): the current date.
* Default (manual configuration): none.

end_date = <string>
* This setting is required when 'schedule' is set to "time_range".
* The date format is YYYY-MM-DD
* Default (in SplunkWeb): the current date.
* Default (manual configuration): none.

user_message = <string>
* Specifies the message when a search is filtered out by this rule.
* Cannot exceed 140 characters.
* Optional.

disabled = <boolean>
* Toggles a search filter rule off and on.
* Set to "true" to disable a rule.
* Default: false
```

workload_rules.conf.example

```
[workload_rules_order]
rules = my_analyst_rule,my_app_rule,my_user_rule,my_index_rule

[workload_rule:my_app_rule]
predicate = app=search
workload_pool = my_app_pool

[workload_rule:my_analyst_rule]
predicate = role=analyst
workload_pool = my_analyst_pool
schedule = always_on

[workload_rule:my_user_rule]
predicate = user=admin
workload_pool = my_user_pool
schedule = always_on

[workload_rule:my_index_rule]
predicate = index=_internal
workload_pool = my_index_pool
schedule = time_range
start_time = 2019-01-01T04:00:00-08:00
end_time = 2019-01-05T04:00:00-08:00

[workload_rule:my_search_type_rule]
predicate = search_type=adhoc
workload_pool = my_adhoc_pool
schedule = every_day
start_time = 10
end_time = 15

[workload_rule:my_logical_rule_1]
predicate = app=search AND (NOT index=_internal)
workload_pool = my_logical_pool_1
schedule = every_week
start_time = 10
end_time = 23
every_week_days = [0,4,6]

[workload_rule:my_logical_rule_2]
predicate = NOT role=power OR user=admin
workload_pool = my_logical_pool_2
schedule = every_month
start_time = 1
end_time = 2
every_month_days = [1,5,16,31]
```