Splunk® IT Service Intelligence Administration Manual 4.16.0 Cloud only

Generated: 4/29/2023 2:09 am

Copyright (c) 2023 Splunk Inc. All Rights Reserved

Table of Contents

Table of Contents

Configuration file reference

Overview

About administering IT Service Intelligence

Splunk IT Service Intelligence (ITSI) is a scalable IT monitoring and analytics solution that provides actionable insight into the performance and behavior of your IT operations. This manual covers the tasks of administering an ITSI deployment, including configuring users and roles, scheduling maintenance windows, and backing up your ITSI environment.

The following tables describes the tasks involved in administering an ITSI deployment:

Name	Description
Configure users and roles	Add users, assign users to roles, and assign those roles custom capabilities to provide granular, role-based access control for your organization.
Manage teams	Create teams to restrict service-level information to certain departments or organizations.
Schedule maintenance downtime	Set up maintenance windows to prevent alerts from triggering from machines and other devices that are undergoing maintenance operations or don't require active monitoring.
Back up and restore ITSI KV store data	Regularly back up the KV store and restore your ITSI data from a backup in the event of a disaster or if you add a search head to a cluster. You can perform both full backups and partial backups of your data.

Admins are also responsible for ingesting and analyzing entities, creating services and KPIs, and setting up alerts. These tasks are covered in the following manuals:

- For information about importing entities and setting up entity integrations, see the *Entity Integrations Manual*.
- For information about setting up services and defining KPIs, see the *Service Insights Manual*.
- For information about aggregating notable events and setting up alert actions, see the *Event Analytics Manual*.

See also

- For new features and bugs fixes, see the ITSI *Release Notes*.
- For information about installing and upgrading ITSI, see the *Install and Upgrade Manual*.

Permissions

Configure users and roles in ITSI

Splunk IT Service Intelligence (ITSI) uses the access control system integrated with the Splunk platform. The Splunk platform authorization allows you to add users, assign users to **roles**, and assign those roles custom **capabilities** to provide granular, role-based access control for your organization.

> Never delete the default admin user from your Splunk instance. The admin user is necessary for many IT Service Intelligence features, such as notable event grouping in Episode Review. For more information about users, see About user authentication in the Securing Splunk Enterprise manual.

Overview of ITSI roles

Splunk IT Service Intelligence provides four special roles with predefined capabilities:

Role	Description
`itoa_user`	Assign this role to users who need basic read access to ITSI.
`itoa_analyst`	Assign this role to knowledge managers in your organization who will create glass tables, deep dives, and service analyzers and work with episodes in Episode Review.
`itoa_team_admin`	Create team admin roles that inherit from this role. Team admins can create and administer services, and update objects for ITSI teams to which they are assigned read/write access. This role can also create and manage notable event aggregation policies.
`itoa_admin`	Assign this role to ITSI administrators. Admins create teams for team administrators to administer as well as create objects in the Global team. This role is required to assign access to objects such as glass tables to other ITSI roles. Note that users with the Splunk `admin` role also have the `itoa_admin` role.

Splunk Enterprise administrators can assign users to these roles to grant an appropriate level of access to specific ITSI functions. The role to which you assign a user depends on the specific tasks the user performs inside of ITSI, and level of security access that a user requires. Splunk Cloud Platform administrators with the role `sc_admin` need to request Splunk support to assign users to the ITSI roles.

You can also create custom roles. If your organization is planning to use teams to manage service-level permissions, you need to create custom roles that inherit from the provided ITSI roles. See Create custom roles for teams for information.

ITSI roles and capabilities

The following table summarizes ITSI roles, inheritance, and capabilities. ITSI roles inherit from lesser ITSI roles and thus inherit the capabilities of the lesser roles. For a full list of ITSI capabilities and the functions they provide, see ITSI capabilities reference.

Role	Inherits from role	Capabilities
`itoa_user`	user, user_ad_user*	• read services, KPIs, and entities • read service templates • read KPI base searches • read KPI threshold templates

Role	Inherits from role	Capabilities
		• read glass tables and write their own private glass tables • read the default Service Analyzer (homeview) • read deep dives • read/write/delete deep dives context (drilldown from Service Analyzer or notable events) • read correlation search • read/write/delete notable event management state • read notable events • read notable event actions • read team objects
`itoa_analyst`	itoa_user, user, power, user_ad_user*	All capabilities of itoa_user plus the following: • read/write/delete glass tables • read/write/delete deep dives • read/write/delete saved service analyzers • read/write/delete notable events • read/execute notable event actions • read notable event aggregation policies
`itoa_team_admin`	itoa_analyst, user, power, metric_ad_admin*	All capabilities of itoa_analyst plus the following: • configure permissions • read/write/delete services, KPIs, and entities • read/write/delete KPI base searches • read/write/delete KPI threshold templates • read/write/delete correlation search • read/write/delete maintenance windows • read/write/delete modules • read/write/delete notable event aggregation policies • write/delete team objects
`itoa_admin`	itoa_team_admin, user, power, metric_ad_admin*	All capabilities of itoa_team_admin plus the following: • read/write/delete service templates • perform bulk import of entities and services via CSV/search • read/write/delete backups and restores • edit the default notable event aggregation policy
`admin`	itoa_admin, itoa_analyst, itoa_user, user, power	All

*The `user_ad_user` and `metric_ad_admin` roles are inherited by ITSI roles for the purposes of using anomaly detection in ITSI. Do not assign these roles to users separately.

ITSI role capabilities apply only to shared objects. Users assigned to the `itoa_user` role can create and manage private service analyzers, glass tables, and deep dives.

If you have the itoa_admin or itoa_team_admin role, or the capabilities of these roles, you need write access to the Global team to write and delete global objects such as service templates, entities, KPI templates, base searches, and threshold templates.

To execute actions as part of the ITSI ServiceNow or Remedy integrations, all roles that are not admin roles require access to the list_storage_passwords and the edit_token_http capability. Note that this capability enables users to view stored passwords for other applications, which is typically an admin-level capability. Ensure that you provide access only to users with proper security credentials to view stored passwords for other applications. The list_storage_passwords capability does not provide access to individual user credentials.

Splunk Admin Capabilities and ITSI Roles

Some ITSI roles inherit capabilities that are typically only available to Splunk administration roles.

The following table lists the capabilities and ITSI roles that have these capabilities:

Capability	itoa_user	itoa_analyst	itoa_team_admin	itoa_admin
edit_token_http	x	x	x	x
list_storage_passwords		x	x	
list_search_head_clustering			x	x
dispatch_rest_to_indexers			x	x
list_settings			x	
edit_monitor			x	

Enable or disable ITSI capabilities for a role

You can enable or disable object capabilities for ITSI roles in authorize.conf.

1. Open or create a copy of `authorize.conf` in `$SPLUNK_HOME/etc/apps/itsi/local/` directory.
2. In the local file, enable or disable the appropriate capabilities for ITSI-specific roles. To disable a capability, replace `enabled` with `disabled` or delete the capability from the file.

For example, the following example shows a portion of the `authorize.conf` file with `read_itsi_glass_table = disabled` for `role_itoa_user`:

```
## ITOA User
## The ITOA user role inherits user role
## This allows users assigned to the itoa_user role to perform all capabilities of a Splunk user
## The itoa_user role can also perform RT search
[role_itoa_user]
importRoles = user;user_ad_user

## Core dependent capabilities
list_storage_passwords = enabled
rtsearch = enabled

# For event management
edit_token_http = enabled

## ITSI specific/controlled capabilities

# Glass Table
read_itsi_glass_table = disabled

# Deep Dive
read_itsi_deep_dive = enabled
read_itsi_deep_dive_context = enabled
write_itsi_deep_dive_context = enabled
delete_itsi_deep_dive_context = enabled
```

Create custom roles for teams

If you decide to create teams in ITSI to segment your service-level data, you must create custom roles that inherit from the standard ITSI roles. Then you can assign permissions to specific roles that correspond to specific teams. See Implement teams in ITSI for information about service-level permissions and teams.

Create a role in the Splunk platform for each ITSI team admin and configure the roles to inherit from the `itoa_team_admin` role so it has the appropriate capabilities. Then assign users to each team admin role you created.

For example, the Splunk admin creates an `itoa_finance_admin` role to administer the Finance team. The role inherits from the `itoa_team_admin`. The Splunk admin then assigns the Finance team administrator to the `itoa_finance_admin` role.

The Finance team administrator then creates custom roles for the analysts and users on the Finance team. For example, create an `itoa_finance_analyst` role that inherits from the `itoa_analyst` role for the analysts in the Finance department. Likewise, create an `itoa_finance_user` role that inherits from the `itoa_user role` for the users in the Finance department.

The team administrator can then assign permissions to the Finance team for the `itoa_finance_analyst` and `itoa_finance_user` roles without allowing access to analysts and users from other departments.

> You must configure the itoa_admin role to inherit from the custom roles you create, otherwise the itoa_admin role cannot assign permissions to the custom roles. Alternatively, use the admin role to assign permissions.

Splunk Cloud Platform administrators need to request Splunk Support to create the custom roles for teams.

For information about creating custom roles, see About configuring role-based user access in the *Securing Splunk Enterprise* manual.

Using teams in conjunction with other access controls

Teams provide a more granular level of access control than the roles provided with ITSI. Teams let you restrict read/write access to services and the KPIs associated with services within ITSI views such as glass tables, deep dives, and service analyzers.

For example, a user might have permission to view a particular glass table, but if a KPI in that glass table belongs to a service in a team for which the user does not have read permission, the KPI is not displayed. Only the data related to services for which the user has read access are displayed on the glass table.

To prevent users from being confronted with widgets they cannot view in glass tables or lanes they cannot view in deep dives, keep in mind the intended audience when creating a shared glass table or deep dive and create these visualizations for a particular team.

For example, if you are creating a glass table for the Finance team, create a shared glass table that only includes services and KPIs in the Finance team or Global team and assign read/write permissions for the glass table to the Finance team roles. Then users from other teams won't try to access the glass table and get frustrated when they can't view all of the information.

See Overview of teams in ITSI for detailed information about service-level permissions and teams.

Create a custom role in ITSI

If you create a new role that does not inherit from one of the standard ITSI roles, you need to do four things to ensure the custom role has the appropriate level of access in ITSI:

1. Assign the role proper capabilities.
2. Grant the role access to ITSI indexes.
3. Assign the role proper view-level access.
4. Assign the role KV store collection level access.

For example, in order to assign a new role write permissions to a deep dive, that new role must first be assigned the `write_deep_dives` capability. The new role must also have write access to the `saved_deep_dives_lister` view, and write access to the `itsi_pages` collection.

Step 1: Assign the role proper capabilities

You can enable or disable object capabilities for ITSI roles in authorize.conf.

Prerequisites

- Only users with file system access, such as system administrators, can assign object capabilities using a configuration file.
- Review the steps in How to edit a configuration file in the *Admin Manual*.

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.

Steps

1. Open or create a local copy of authorize.conf in `$SPLUNK_HOME/etc/apps/itsi/local/` directory.
2. In the local file, enable or disable the appropriate capabilities for ITSI-specific roles. To disable a capability, replace `enabled` with `disabled` or delete the capability from the file. For an example, see Enable or disable ITSI capabilities for a role.

Step 2: Grant the role access to ITSI indexes

By default, all ITSI-specific roles have access to ITSI indexes. If you create a custom role in ITSI, assign the role access to the ITSI indexes.

If you do not update the roles with the correct indexes, searches and other objects that rely on data from unassigned indexes do not update or display results.

1. Click **Settings** > **Roles** (or **Settings** > **Access controls** > **Roles** on Splunk versions prior to 8.1.0)
2. Open the custom role.
3. Go to the **Indexes** tab.
4. Check the box in the Included tab for each of the following indexes:
   - `anomaly_detection`
   - `itsi_grouped_alerts`
   - `itsi_notable_archive`
   - `itsi_notable_audit`
   - `itsi_summary`

        ◊ `itsi_summary_metrics`
        ◊ `itsi_tracked_alerts`
        ◊ `snmptrapd` (optional, used only if you're collecting SNMP traps)
5. Click **Save**.
6. (Optional) Repeat for additional roles, as needed.

Step 3: Assign the role proper view-level access

ITSI includes default entries in `itsi/metadata/default.meta` that determine access for ITSI roles to specific ITSI views. By default, only `itoa_admin` has read/write permissions for all ITSI views.

Set permissions to ITSI views in Splunk Web

1. In Splunk Web, go to **Settings > All configurations**.
2. Set the App to **IT Service Intelligence (itsi)**. Set the Owner to **Any**.
3. Change **Visible in the App** to **Created in the App** to narrow the view to only ITSI objects.
4. Filter by `views` to only display ITSI views.
5. For a specific view, click **Permissions** in the Sharing column.
6. Check the boxes to grant read and write permissions for ITSI roles.
7. Click **Save**.

This action updates the access permissions to ITSI views for ITSI roles in `$SPLUNK_HOME/etc/apps/itsi/metadata/local.meta`.

Set permissions to ITSI views from the command line

1. Create a `local.meta` file in the `itsi/metadata/` directory.

   ```
   cd $SPLUNK_HOME/etc/apps/itsi/metadata
   cp default.meta local.meta
   ```
2. Edit `itsi/metadata/local.meta`.
3. Set access for specific roles in `local.meta`. For example:

   ```
   [views/glass_tables_lister]
   access = read : [ itoa_admin, itoa_analyst, itoa_user ], write: [itoa_admin]
   ```

Step 4: Assign the role KV store collection level access

The `SA-ITOA` file includes default entries in `metadata/default.meta` that determine access to KV store collections for ITSI roles. For a list of default permissions to KV store collections for ITSI roles, see KV store collection permissions in ITSI. By default, only the `itoa_admin` role has read/write/delete access to all ITSI KV store collections.

Set permissions to KV store collections in Splunk Web

1. In Splunk Web, go to **Settings > All configurations**.
2. Set the App to **IT Service Intelligence (itsi)**. Set the Owner to **Any**.
3. Make sure **Visible in the App** is selected.
4. Filter by `collections-conf` to only display KV store collections.
5. For a specific view, click **Permissions** in the Sharing column.
6. Check the boxes to grant read and write permissions to the various collections for ITSI roles.
7. Click **Save**.

This action updates KV store access permissions for the specific ITSI roles in `$SPLUNK_HOME/etc/apps/SA-ITOA/metadata/local.meta`.

Set permissions to KV store collections from the command line

1. Create a `local.meta` file in the `SA-ITOA/metadata/` directory.

   ```
   cd $SPLUNK_HOME/etc/apps/SA-ITOA/metadata
   cp default.meta local.meta
   ```
2. Edit `SA-ITOA/metadata/local.meta`

   .
3. Set access for specific roles in `local.meta`. For example:

   ```
   [collections/itsi_services]
   access = read : [ itoa_admin, itoa_analyst, itoa_user ], write: [ itoa_admin ]
   ```

ITSI capabilities reference

This table lists ITSI capabilities for each default role. When you create a user in ITSI, you assign that user one or more roles. Each role contains a set of **capabilities**. You can add or edit capabilities for new, existing, and default roles. For example, you might give a role the capability to create a shared glass table or delete a KPI base search. A write capability implies create and update. Delete is its own capability. If you modify the capabilities for custom roles, you also need to assign the role proper view-level access. For instructions, see Assign the role proper view-level access.

Capabilities are subject to change. For the most up-to-date list of capabilities, see `$SPLUNK_HOME/etc/apps/SA-ITOA/default/authorize.conf`. For information about the capabilities assigned to ITSI roles, see Restrict access to objects in ITSI.

> A role that has a service capability has analogous capabilities for the KPI and entity type objects.

SA-ITOA Object type	Capability name	Capability description	itoa_user	itoa_analyst	itoa_team_admin	i
RBAC Permissions Configuration	configure_perms	Configure role based access control on shared service analyzers, deep dives, glass tables, correlation searches, and notable event aggregation policies.			X	X
Service/KPIs/Entity	read_itsi_service	Read service-based information in service analyzers, pull in service-based information on a glass table or deep dive, and list services and entities.	X	X	X	X
	write_itsi_service	Create a service, KPI, and entity, and bulk import entities and			X	X

SA-ITOA Object type	Capability name	Capability description	itoa_user	itoa_analyst	itoa_team_admin	i
		services.				
	delete_itsi_services	Delete a service, KPI, or entity.			X	X
Service Templates	read_itsi_base_service_template	View a service template.	X	X	X	X
	write_itsi_base_service_template	Create a service template.				X
	delete_itsi_base_service_template	Delete a service template.				X
Temporary KPIs	read_itsi_temporary_kpi	Read a KPI with time policy.	X	X	X	X
	write_itsi_temporary_kpi	Create a KPI with time policy.	X	X	X	X
	delete_itsi_temporary_kpi	Delete a KPI with time policy.	X	X	X	X
KPI Base Searches	read_itsi_kpi_base_search	Read a KPI base search.	X	X	X	X
	write_itsi_kpi_base_search	Write a KPI base search.			X	X
	delete_itsi_kpi_base_search	Delete a KPI base search.			X	X
KPI Threshold Templates	read_itsi_kpi_threshold_template	Read KPI threshold template type objects.	X	X	X	X
	write_itsi_kpi_threshold_template	Write a custom KPI threshold template.			X	X
	delete_itsi_kpi_threshold_template	Delete a KPI threshold template.			X	X
	create_external_ticket	Create a ticket in a third-party ticketing system.			X	X
Backup/Restore	read_itsi_backup_restore	Read backup/restore page.				X
	write_itsi_backup_restore	Create a backup/restore job.				X
	delete_itsi_backup_restore	Delete a backup/restore job.				X
Glass Table	read_itsi_glass_table	View shared glass tables.	X	X	X	X
	write_itsi_glass_table	Create and edit a shared glass table. Does not include the ability to drill down in view mode.		X	X	X
	delete_itsi_glass_table	Delete a shared glass table.		X	X	X
	interact_with_itsi_glass_table		X	X	X	X

SA-ITOA Object type	Capability name	Capability description	itoa_user	itoa_analyst	itoa_team_admin	i
		Drill down and interact with glass tables.				
Deep Dive	read_itsi_deep_dive	View a shared deep dive.	X	X	X	X
	write_itsi_deep_dive	Create a shared deep dive.		X	X	X
	delete_itsi_deep_dive	Delete a shared deep dive.		X	X	X
	interact_with_itsi_deep_dives	Drill down and interact with deep dives.	X	X	X	X
	read_itsi_deep_dive_context	Drill down to an automatically-generated deep dive object.	X	X	X	X
	write_itsi_deep_dive_context	Drill down to an automatically-generated deep dive object for the first time.	X	X	X	X
	delete_itsi_deep_dive_context	Delete an automatically-generated deep dive object.	X	X	X	X
	interact_with_itsi_deep_dives_context	Drill down and interact in deep dives context.	X	X	X	X
Service Analyzer	read_itsi_homeview	Read service analyzers.	X	X	X	X
	write_itsi_homeview	Create or edit a service analyzer.	X	X	X	X
	delete_itsi_homeview	Delete a service analyzer.	X	X	X	X
	interact_with_itsi_homeview	Drill down and interact with a service analyzer.	X	X	X	X
Correlation Search	read_itsi_correlation_search	Read a correlation search.		X	X	X
	write_itsi_correlation_search	Edit a correlation search.			X	X
	delete_itsi_correlation_search	Delete a correlation search.			X	X
	interact_with_itsi_correlation_search	Drill down and interact with a correlation search.			X	X
Event Management State	read_itsi_event_management_state	Read Episode Review dashboards.	X	X	X	X
	write_itsi_event_management_state	Save an Episode Review dashboard.	X	X	X	X
	delete_itsi_event_management_state	Delete an Episode Review dashboard.	X	X	X	X

SA-ITOA Object type	Capability name	Capability description	itoa_user	itoa_analyst	itoa_team_admin	i
	interact_with_itsi_event_management_state	Drill down and interact with an Episode Review dashboard.	X	X	X	X
Event management	edit_token_http	Run an episode action, and update episode owner, severity, and status.		X	X	X
Notable Event	read-notable_event	Read a notable event.	X	X	X	X
	write-notable_event	Modify a notable event on index. Requires delete_by_keyword and edit_token_http capabilities to be enabled.		X	X	X
	delete-notable_event	Delete an episode.		X	X	X
Notable Event Aggregation Policy	read_itsi_notable_aggregation_policy	Read a notable event aggregation policy.		X	X	X
	write_itsi_notable_aggregation_policy	Write a notable event aggregation policy.			X	X
	delete_itsi_notable_aggregation_policy	Delete a notable event aggregation policy.			X	X
	edit_default_itsi_notable_aggregation_policy	Edit the default notable event aggregation policy.				X
	interact_with_itsi_notable_aggregation_policy	Drill down and interact with notable event aggregation policies.			X	X
Episode actions	read-notable_event_action	Read an episode action.	X	X	X	X
	execute-notable_event_action	Run an episode action, and update episode owner, severity, and status.		X	X	X
Email templates	read_itsi_notable_event_email_template	Read an email template.		X	X	X
	write_itsi_notable_event_email_template	Edit an email template.		X	X	X
	delete_itsi_notable_event_email_template	Delete an email template.		X	X	X
Maintenance services	read-maintenance_calendar	Read a maintenance window.	X	X	X	X
	write-maintenance_calendar	Write a maintenance window.			X	X
	delete-maintenance_calendar	Delete a maintenance window.			X	X
	delete-module_interface	Delete an ITSI module and KPIs provided by modules.			X	X

SA-ITOA Object type	Capability name	Capability description	itoa_user	itoa_analyst	itoa_team_admin	i
CSV Import mod input	edit_modinput_itsi_csv_import	Save the modular input for CSV import.				X
Teams	read_itsi_team	Read objects for a team.	X	X	X	X
	write_itsi_team	Create or update objects for a team.			X	X
	delete_itsi_team	Delete objects for a team.			X	X
Bulk import	bulk_import_service_or_entity	Create services or entities using bulk import.				X

KV store collection permissions in ITSI

The table shows default permissions to KV store collections for IT Service Intelligence (ITSI) roles. By default, only itoa_admin has read/write/delete access to all ITSI KV store collections. SA-ITOA includes default entries in `metadata/default.meta` that determine access to KV store collections for ITSI roles.

Collection name	itoa_admin	itoa_team_admin	itoa_analyst	itoa_user
itsi_backfill	read/write/delete	read/write/delete	read	read
itsi_backup_restore_queue	read/write/delete	read	-	-
itsi_base_service_template	read/write/delete	read	read	read
itsi_content_pack_status	read/write/delete	read/write/delete	-	-
itsi_correlation_search	read/write/delete	read/write/delete	read	read
itsi_entity_dashboard_drilldown	read/write/delete	read/write/delete	read	read
itsi_entity_data_drilldown	read/write/delete	read/write/delete	read	read
itsi_entity_type	read/write/delete	read/write/delete	read	read
itsi_entity_filter_rules	read/write/delete	read/write/delete	read	read
itsi_entity_relationships	read/write/delete	read/write/delete	read	read
itsi_entity_relationship_rules	read/write/delete	read/write/delete	read	read
itsi_event_management	read/write/delete	read/write/delete	read/write/delete	read/write/delete
itsi_import_objects_cache	read/write/delete	read/write/delete	read	read
itsi_import_objects_cache_lookup	read/write/delete	read/write/delete	read	read
itsi_migration	read/write/delete	read/write/delete	read	read
itsi_notable_event_aggregation_policy	read/write/delete	read/write/delete	read	-
itsi_notable_event_email_template	read/write/delete	read/write/delete	read/write/delete	-
itsi_notable_event_ref_url	read/write/delete	read/write/delete	read/write/delete	read/write/delete
itsi_notable_event_tag	read/write/delete	read/write/delete	read/write/delete	read/write/delete
itsi_notable_event_ticketing	read/write/delete	read/write/delete	read/write/delete	read/write/delete

Collection name	itoa_admin	itoa_team_admin	itoa_analyst	itoa_user
itsi_notable_group_user	read/write/delete	read/write/delete	read/write/delete	read/write/delete
itsi_notable_group_system	read/write/delete	read	read	read
itsi_pages	read/write/delete	read/write/delete	read/write/delete	read/write/delete
itsi_refresh_queue	read/write/delete	read/write/delete	read	read
itsi_services	read/write/delete	read/write/delete	read	read
itsi_service_analyzer	read/write/delete	read/write/delete	read/write/delete	read/write/delete
itsi_team	read/write/delete	read	read	read
itsi_temp_batch_claimed_action_queue	read/write/delete	read/write/delete	read	read
itsi_temporary_storage	read/write/delete	read/write/delete	read	read
itsi_user_realnames	read/write/delete	read/write/delete	read	read
maintenance_calendar	read/write/delete	read/write/delete	read	read
operative_maintenance_log	read/write/delete	read/write/delete	read	read
itoa_entity_exchange_entities*	-	-	-	-
itoa_entity_exchange_entity_hash*	-	-	-	-
itoa_entity_exchange_metadata*	-	-	-	-

* These are entity exchange collections which are used for integrating entities from the Splunk App for Infrastructure with ITSI. Only the Splunk admin role has access to these collections.

Note: As of ITSI 4.9.0, the Splunk App for Infrastructure is no longer packaged with ITSI.

Grant and revoke user permissions in ITSI

You can set read and write permissions for the following types of ITSI objects:

- Service analyzers
- Glass tables
- Deep dives
- Episode review dashboards
- Correlation searches
- Multi-KPI alerts
- Notable event aggregation policies

These permissions determine which user roles have read/write permissions to specific objects that have been created, such as a shared service analyzer view, a shared glass table, or a correlation search.

These permissions apply to shared objects. A user can create a private object that only they have access to. To set permissions to a private service analyzer, glass table, deep dive or episode reviews, clone and save the object with **Shared in App** permissions. The other object types, including correlation searches, multi-KPI alerts, and notable event aggregation policies, are shared by default.

Prerequisites

- You must have the configure_perms capability to set permissions for ITSI roles. The itoa_admin and itoa_team_admin roles have this capability by default.
- Before you can set permissions to ITSI objects for a role, the role must have the proper capabilities assigned. For more information, see the ITSI capabilities reference in this manual.

Steps

1. On the lister page for the object type, such as service analyzer, glass table, deep dive, episode review dashboard, correlation search, or notable event aggregation policy, click **Edit** under the Actions column for the object and select **Edit Permissions**.
2. Assign read/write permissions to ITSI roles for the object.
3. Click **Save.**

Teams

Overview of teams in ITSI

Teams provide presentation-layer security only and not data-level security. It's still possible for a user with access to the Splunk search bar to look up ITSI summary index data.

Teams let you restrict service-level information in IT Service Intelligence (ITSI) to only the departments or organizations that need access to that information. Teams empower domain experts in different areas within an organization to create and monitor the services that pertain to their department. If your organization doesn't need to restrict service visibility to specific areas within your organization, you don't need to implement teams. For information about setting up or viewing the teams in your environment, see Create teams in ITSI.

An ITSI administrator creates teams and assigns write permissions to them by role. When you create a service, you specify the team it belongs to. A service can only belong to one team.

If a user doesn't have read access to a team, they can't see data from the services within that team in the following object types:

- Glass tables
- Service analyzers
- Deep dives
- Correlation searches
- Multi-KPI alerts
- Episode Review

Users with restricted access to service-level information might still be able to edit other information in a glass table, deep dive or other visualization if they have write permissions for that object.

How teams differ from other access controls in ITSI

Teams provide another level of access control on top of those delivered by default with ITSI.

Without teams

ITSI is delivered with ITSI-specific roles. These roles have capabilities that control access to different features in ITSI. You can change the capabilities assigned to roles as needed. For more information, see ITSI capabilities reference.

You can also set read and write permissions for glass tables, deep dives, and other ITSI objects. For more information, see Configure users and roles in ITSI.

With teams

Teams restrict read and write permissions to the underlying objects within ITSI visualizations, such as KPI base searches.

For example, a user might have permission to view a particular glass table. However, if a KPI in that glass table belongs to a service in a team for which the user doesn't have read permission, the KPI value isn't displayed. Only the data related to services for which the user has read access appear on the glass table.

What's in the Global team?

By default, all ITSI objects are contained within the default Global team. If you don't need to restrict service visibility to specific teams in your organization, create all services in the Global team. You can't delete the Global team.

The following objects can only be created in the Global team and can't belong to a specific team:

- Service templates
- Entities
- KPI templates (provided by modules)
- KPI base searches
- KPI threshold templates

The Global team contains common services shared across all departments. The ITSI administrator, using the `itoa_admin` role, configures the deployment and creates services in the Global team that are common and used across departments.

Each team admin, with roles inherited from the `itoa_team_admin` role, creates services for their department with dependencies on Global services, if needed. Members of a specific team can't view services in another team.

The administrator provides support to the admins of the dependent services. The `itoa_admin` role has full access to view and change the basic services in the Global team as well as team-specific services as needed.

Global team permissions

The itoa_admin role has write access to the Global team and can change permissions on the Global team. All other roles have read access. Read access ensures that services in other teams can still use objects in the Global team.

If you have the itoa_admin or itoa_team_admin role, or the capabilities of these roles, you need write access to the Global team to write and delete global objects such as service templates, entities, KPI templates, base searches, and threshold templates.

Team admin role

The `itoa_team_admin` role is delivered with ITSI to help departmental admins manage services for their team. This role has all the capabilities of the `itoa_admin` role, except it can't perform backups and restores, perform bulk imports of entities and services, or create service templates.

The `itoa_team_admin` role can't create new teams. The `itoa_admin` role creates teams and has read and write access to all teams that are created, as well as to the Global team.

The administrator creates custom roles for each departmental admin that will manage a team in ITSI. These team admin roles must inherit from the `itoa_team_admin` role to obtain the appropriate capabilities. For more information about this role's capabilities, see Configure users and roles in ITSI.

Example

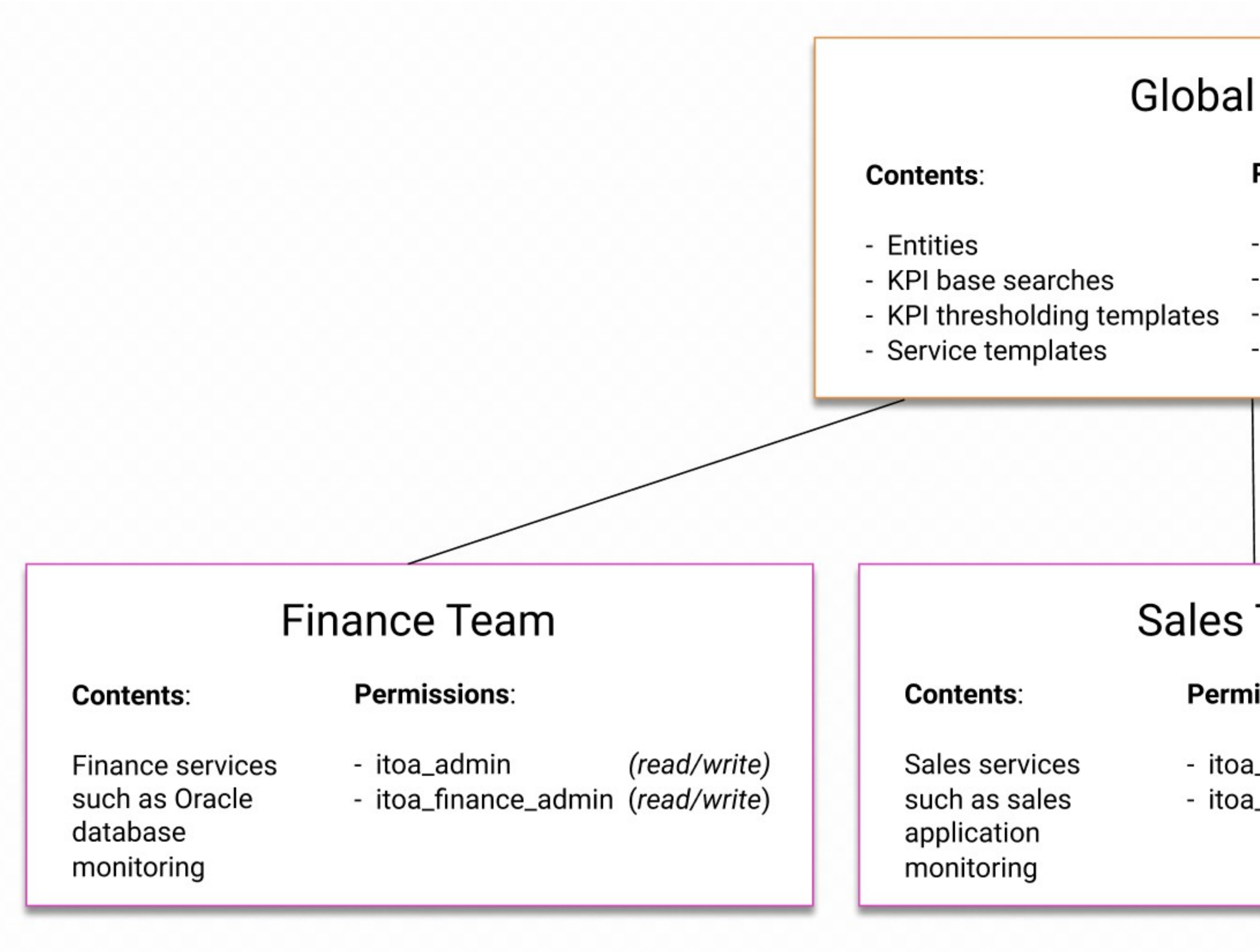
In the following example, an ITSI administrator wants to create three teams within ITSI: a **finance** team, a **sales** team, and an **engineering** team.

The administrator performs the following steps:

1. Creates an `itoa_finance_admin` role for the admin of the Financial department.
2. Creates an `itoa_sales_admin` role for the admin of the Sales department.
3. Creates an `itoa_eng_admin` role for the admin of the Engineering team.
4. Assigns read/write permissions to the `itoa_finance_admin` role for the Finance team.
5. Assigns read/write permissions to the `itoa_sales_admin` role for the Sales team.
6. Assigns read/write permissions to the `itoa_eng_admin` role for the Engineering team.

As a result, the `itoa_finance_admin` role can create services in the Finance team, the `itoa_sales_admin` role can create services in the Sales team, and the `itoa_eng_admin` can create services in the Engineering team.

The following diagram shows how read and write permissions are assigned within the three teams:


Global
Contents:
- Entities
- KPI base searches
- KPI thresholding templates
- Service templates
P
-
-
-
-
Finance Team
Contents:
Finance services such as Oracle database monitoring
Permissions:
- itoa_admin (read/write)
- itoa_finance_admin (read/write)
Sales T
Contents:
Sales services such as sales application monitoring
Permi
- itoa_
- itoa_

Create teams in ITSI

Implement teams in IT Service Intelligence (ITSI) to restrict service-level information to only the departments or organizations that need access to it. Teams empower domain experts in different areas within an organization to create and monitor the services that pertain to their department.

Prerequisites

- See Overview of teams in ITSI to determine whether you need to implement teams for your organization.
- Plan out what teams you need to create in ITSI. You can create teams for technology areas or for different departments within your organization. Create a team for every area that needs a separate view of ITSI service-level data or that needs to be administered independently within ITSI.

High-level steps

1. Create team admin roles to administer each team and assign users to those roles.
2. Create custom analyst and user roles for each team.
3. Create teams and assign read/write permissions to the team admin roles you created.
4. Create services within teams.

Step 1: Create roles to administer your teams

After you determine the teams you are going to create in ITSI, create roles to administer the services in each team.

1. Create a role in the Splunk platform for each ITSI team admin.
2. Configure the roles to inherit from the `itoa_team_admin` role in order to obtain the appropriate capabilities.
3. Assign users to each team admin role you created.

For example, the Splunk administrator creates an `itoa_finance_admin` role that inherits from the `itoa_team_admin` role for the administrator of the Finance team. The Splunk admin then assigns the Finance team administrator to the `itoa_finance_admin` role.

Splunk Cloud Platform administrators need to request Splunk Support to create the custom roles needed for teams.

For information about the `itoa_team_admin` role's capabilities, see Configure users and roles in ITSI. For information about creating custom roles, see About configuring role-based user access.

Step 2: Create custom roles within each team

Create custom roles for the ITSI analysts and users in each team. For example, create an `itoa_finance_analyst` role that inherits from the `itoa_analyst` role for the analysts in the Finance department. Create an `itoa_finance_user` role that inherits from the `itoa_user` role for the users in the Finance department. You can then assign permissions to the Finance team without allowing access to analysts and users from other departments.

> You must configure the itoa_admin role to inherit from the custom roles you create. Otherwise, the itoa_admin role cannot assign permissions to the custom roles. Alternatively, use the admin role to assign permissions.

Step 3: Create teams

Create teams to group services by department, organization, or type of service and control access to the services.

Prerequisites

- You must have the `itoa_admin` role to create a team.
- Before you create a team, you must create the team admin role that will administer the team so that you can assign permissions to the role when creating the team. See Implement teams in ITSI for information.

Steps

1. Click **Configuration** > **Teams**.
2. Click **Create Team**.
3. Provide a team name and description. Duplicate team names are allowed, but be aware of other team names and use naming conventions to avoid confusion.
4. Assign read or write access to the listed roles as appropriate. The `itoa_admin` role has read/write permissions by default. If a role has write permissions for a team, a user with this role can create and modify services in the team. The user can't delete a service in the team unless the role has the delete capability for a service.
5. Click **Create**.

> If you do not see the custom team admin role listed for which you want to assign permissions, make sure the role has been created and inherits from the itoa_team_admin role. If you are logged in using the itoa_admin role, rather than the admin role, also make sure that the itoa_admin role inherits from the custom team admin role and any other custom roles you have created.

Open a team to see the services that belong to it or to review or change team permissions.

Step 4: Create services within each team

After the administrator creates teams, the team admins that are assigned read/write permissions can create services within their teams. When creating a service, a team admin can assign it to any team for which they have read/write permissions. ITSI administrators can also create services in private teams.

Team admins can access all of the KPI base searches, KPI templates, and entities in the Global team when creating services in their private teams. Team admins can also create dependencies on services in the Global team or within the same team. You cannot create service dependencies between services in different private teams. See Overview of creating services in ITSI for more information.

Add existing services to a team

> Adding a service to a team breaks service dependencies if the service is dependent on another service that cannot be accessed from the new team. If a service is dependent on another service within the same team and one of the services is moved to another team, that dependency is broken. A service in a private team cannot have a dependency on a service in another private team.

If you're implementing teams in a previously configured environment, you might already have existing services you want to assign to a team.

1. From the ITSI main menu, click **Configuration** > **Services**.
2. Click the checkboxes next to the services you want to add to a team.
3. Click click **Bulk Action** > **Edit Team**.
4. Select the team to add the services to.
5. Click **Save**.

Move a service to a different team

> Moving a service from one team to another team breaks service dependencies if the service is dependent on another service that cannot be accessed from the new team. If a service is dependent on another service within the same team and one of the services is moved to another team, that dependency is broken. A service in a private team cannot have a dependency on a service in another private team.

Prerequisite

You must have write permissions for both teams to move a service from one team to another.

Steps

1. Click **Configuration** > **Teams**.
2. Select the team containing the service you want to move.
3. Select the name of the service you want to move.
4. Go to the **Settings** tab.
5. Click the **Team** dropdown and select the team to move the service to.
6. Click **Save**.

To move more than one service at a time, click **Configuration** > **Services** and select the checkboxes next to the services you want to move. Click **Bulk Actions** > **Edit Team** and select the team to move the services to.

Maintenance windows

Overview of maintenance windows in ITSI

Maintenance windows allow for IT Service Intelligence (ITSI) knowledge objects to enter a maintenance state. This state is intended to silence alarms about machines that don't require active monitoring.

It's a best practice to schedule maintenance windows with a 15- to 30-minute time buffer before and after you start and stop your maintenance work. This gives the system an opportunity to catch up with the maintenance state and reduces the chances of ITSI generating false positives during maintenance operations.

For example, if a server will be shut down for maintenance at 1:00PM and restarted at 5:00PM, the ideal maintenance window is 12:30PM to 5:30PM. The 15- to 30-minute time buffer is a rough estimate based on 15 minutes being the time period over which most KPIs are configured to search data and identify alert triggers.

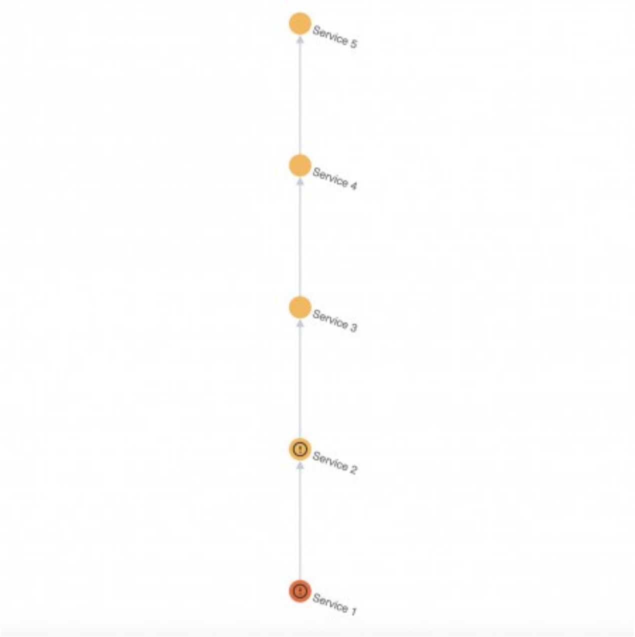
Maintenance windows apply to services and entities. For instructions on putting a service or entity into maintenance mode, see Schedule maintenance downtime in ITSI.

Manage maintenance windows through the REST API

The Maintenance Service Interface encapsulates operations on maintenance windows in ITSI. Use this interface to perform CRUD operations on maintenance windows in your environment. For more information, see Maintenance Services Interface in the IT Service Intelligence *REST API Reference* manual.

Maintenance mode and service dependencies

If you want a service to be in maintenance mode, you need to put all services it depends on in maintenance mode as well, if their `ServiceHealthScore` KPIs are included as dependencies. This rule applies even if you only want to put a single service in maintenance mode. For example, the following topology tree shows that `Service 5` depends on `Service 4`, `Service 4` depends on `Service 3`, and so on:



If you want to put `Service 3` in maintenance mode, and it depends on the `ServiceHealthScore` KPI of services 1 and 2, then those services must be put in maintenance mode as well. The same applies to `Service 5`, which depends on the health scores of all the other services. You need to put services 1, 2, 3, and 4 in maintenance mode in order for `Service 5` to be in maintenance mode.

For an explanation of how dependencies are visually represented in the service tree, see Use the Service Analyzer tree view in ITSI in the *Service Insights Manual*.

For instructions to add dependencies to a service, see Add service dependencies in ITSI in the *Service Insights* manual.

Impact of services in maintenance mode

Maintenance windows can have an impact on associated KPIs, service health score calculations, and other ITSI features.

Consider the following when you put a service into maintenance mode:

- All KPIs within that service are automatically put into maintenance mode.
- ITSI ignores search results from KPIs in maintenance mode for the purpose of service health score calculation for the duration of the maintenance window.
- Maintenance windows don't affect adaptive threshold calculations. Search results from KPIs in maintenance mode don't count when looking back at past data to calculate threshold values.

Impact of entities in maintenance mode

Consider the following when you put an entity into maintenance mode:

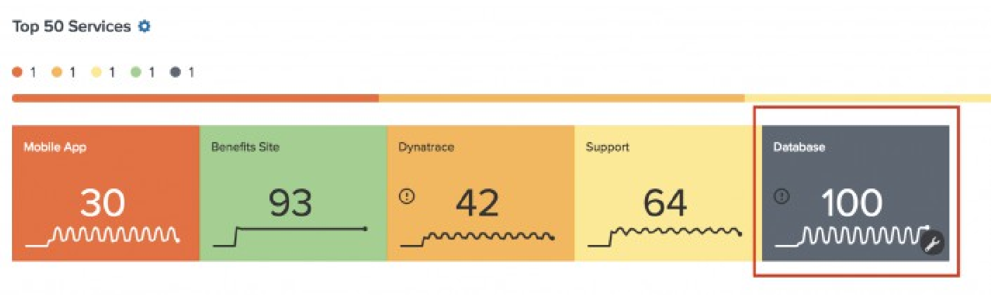
- If the entity has no KPIs running searches against it, there is no impact on service health scores.
- If the entity has one or more KPIs running searches against it, all search results from all KPIs running against that entity are ignored for the purpose of service health score calculation.
- If a KPI is split by entity, for example if the same KPI is running against two different entities, and one entity is in maintenance mode and one is not, search results generated by the KPI running against the

entity in maintenance mode are ignored for the purpose of health score calculation. Search results generated by the same KPI running against the entity that's not in maintenance mode are included as usual in the service health score calculation.

- You can put an entity in full or partial maintenance mode without it being explicitly put into maintenance mode, if a service that contains the entity is put in maintenance mode.

Impact on dashboards

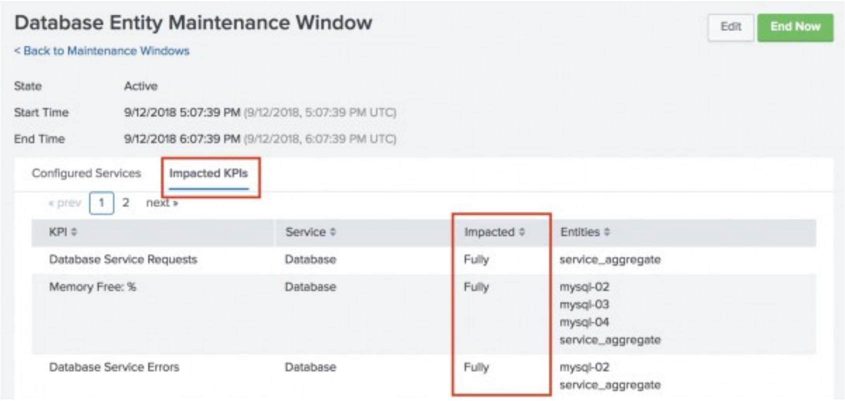
Services, entities, and KPIs that are fully or partially impacted by a maintenance window appear in a dark gray color on pages that display health scores, including service analyzers, service and entity details pages, glass tables, multi-KPI alerts, and deep dives.



View impacted KPIs

You can view the impact of a maintenance window on associated KPIs.

1. Click **Configuration** > **Maintenance Windows**.
2. Select a maintenance window to see the specific services or entities impacted by it.
3. Click **Impacted KPIs** to see a list of KPIs impacted by the maintenance window. KPIs that are split by entity, and thus are currently running searches against other entities that are not in maintenance mode, are listed as partially impacted. KPIs that aren't split by entity are listed as fullyimpacted.



Database Entity Maintenance Window

< Back to Maintenance Windows

State	Active
Start Time	9/12/2018 5:07:39 PM (9/12/2018, 5:07:39 PM UTC)
End Time	9/12/2018 6:07:39 PM (9/12/2018, 6:07:39 PM UTC)

KPI	Service	Impacted	Entities
Database Service Requests	Database	Fully	service_aggregate
Memory Free: %	Database	Fully	mysql-02 mysql-03 mysql-04 service_aggregate
Database Service Errors	Database	Fully	mysql-02 service_aggregate

Schedule maintenance downtime in ITSI

Schedule maintenance downtime in IT Service Intelligence (ITSI) to prevent alerts from triggering from machines and other devices that are undergoing maintenance operations or don't require active monitoring. Defining maintenance windows for scheduled downtimes, such as during server or software upgrades, reduces unnecessary noise. For more information about maintenance windows and their impact on ITSI objects, see Overview of maintenance windows in ITSI.

For example, if your ticketing system experiences a failure and you start to receive a cascade of identical notable events, you can put the service monitoring the ticketing system into maintenance mode to stop ITSI from generating alerts until the issue is resolved.

Maintenance windows apply to services and entities. You can only see maintenance windows that contain at least one service or entity you have read access to. If you have only read permissions for the services or entities in a maintenance window, you can view the maintenance window but not edit it.

If you're bulk deleting maintenance windows, you can only delete the maintenance windows that contain services or entities for which you have write access.

Prerequisites

Only users assigned the `write-maintenance_calendar` capability as well as write access to the Global team can create a maintenance window. By default, the `itoa_admin` and `itoa_team_admin` roles have this capability.

Steps

1. From the ITSI top menu bar, click **Configuration** > **Maintenance Windows**.
2. Click **Create Maintenance Window**.
3. Provide a title for the maintenance window. For example, "DB entity maintenance window."
4. Set the start time, duration, and end time for the maintenance window.
5. Select the **Objects** for which you want to create a maintenance window: **Entities** or **Services**.

   > If you don't have write access to the Global team, you can't put entities into maintenance mode.
6. Click **Next**.
7. Select the specific services or entities that you want to place in maintenance mode for the duration of the maintenance window. You can only select services or entities for which you have write access.
8. Click **Create**. The selected entities or services enter maintenance mode according to the defined schedule.

Backup and restore

Overview of backing up and restoring ITSI KV store data

Regularly backing up the KV store lets you restore your IT Service Intelligence (ITSI) data from a backup in the event of a disaster or if you add a search head to a cluster. You can perform both full backups and partial backups of your data.

When you run a backup job, ITSI saves your data to a set of JSON files compressed into a single ZIP file located in `$SPLUNK_HOME/var/itsi/backups` on the search head. ITSI detects and preserves the application version that it creates a backup from. When you restore from a backup, ITSI detects the correct version of the backup and performs the required migration.

You can perform the following backup and restore operations within ITSI:

- Create a full backup of ITSI
- Create a partial backup of ITSI
- Restore a full or partial backup of ITSI

Splunk Cloud Platform customers must back up and restore their data from the ITSI user interface.

The following table describes the functionality available in each backup and restore method:

Method	Backup/Restore UI	Command line script	Comments
Full backup	X	X	
Partial backup	X	X	If you perform a partial backup using the command line script, the backup does not include dependent objects.
Partial restore		X	
Merge changes during restore	X	X	Merges objects in the backup with existing KV store objects.
Clean restore		X	Replaces existing KV store objects with objects in the backup.

In addition to any custom backup jobs you create, ITSI also takes a default scheduled backup of your KV store data every day at 1:00 AM. For more information, see About the default scheduled backup in ITSI.

Difference between an ITSI backup and a Splunk Enterprise backup

Splunk Enterprise offers an option to back up and restore the KV store. For more information, see Back up and restore KV store in the Splunk Enterprise *Admin Manual*. However, an ITSI backup is specifically formatted to process the content in the ITSI backup files. The Splunk Enterprise backup is not formatted like an ITSI backup, so you cannot use it to back up your ITSI data.

ITSI processes all backup content. ITSI also triggers many other activities, such as saved search generation and object dependency updates. Directly restoring Splunk Enterprise KV store data does not restore the ITSI system completely. Instead, use the processes described in this topic to back up your ITSI data.

What gets backed up

The following table describes the types of data included and not included in an ITSI backup.

Data	Included in backup?	Example
KV store objects	Yes	Services, service templates, entities, KPIs, KPI base searches, teams, glass tables, service analyzers, deep dives
Indexed data	No	ITSI summary index, notable events

To back up indexed data, use the same approach you use to back up other Splunk indexes. For more information, see Back up indexed data in the Splunk Enterprise *Managing Indexers and Clusters of Indexers* manual.

Back up and restore in a search head cluster environment

You can run backup and restore jobs from the Backup/Restore page in search head cluster environments. You can create a backup on any cluster member and then restore data from that backup on any cluster member, regardless of where you initiated the backup.

For example, suppose your search head cluster has three cluster members: `sh-01`, `sh-02`, and `sh-03`. If you create a backup on `sh-01`, you can restore that backup on `sh-01`, `sh-02`, or `sh-03`.

When you create a backup on any search head cluster member, the configuration data from all cluster members is backed up. Likewise, when you restore from a backup on any cluster member, configuration data is restored across all cluster members.

In a search head cluster environment, the scheduled backup runs only on the search head cluster captain. However, you can restore a scheduled backup from any cluster member. If you download the scheduled backup, make sure to download it from the captain as it contains the latest backup.

About the default scheduled backup in ITSI

The default scheduled backup is a full backup of your IT Service Intelligence (ITSI) environment that runs daily at 1:00 AM in the server's local time zone. The time of the backup job is displayed on the Backup/Restore Jobs page. You can't create additional scheduled backup jobs.

The following limits apply to keeping scheduled backup files in the backup history:

- A minimum of 7 days if running a daily scheduled backup
- A minimum of 14 days if running a weekly scheduled backup
- A maximum of 30 days for all scheduled backup files

The oldest backup file is automatically deleted from the backup history after the retention time expires. To set the schedule and retention time period for scheduled backups, edit the settings as defined in the next section.

Edit the default scheduled backup

When you edit the default scheduled backup, you can change the name of the scheduled backup, the frequency and time to run the scheduled backup, how many days to keep the backup file, or you can disable the scheduled backup. You can't delete the scheduled backup job.

Prerequisites

You must have the itoa_admin role to view and edit the settings for the default scheduled backup job.

Steps

1. From the ITSI main menu, click **Configuration** > **Backup/Restore**.
2. Open the **Default Scheduled Backup**.
3. (Optional) Enable **Include .conf files** to back up the following configuration files located in `$SPLUNK_HOME/etc/apps/SA-ITOA/local` and `$SPLUNK_HOME/etc/apps/itsi/local`:
    - alert_actions.conf
    - app_common_flags.conf
    - commands.conf
    - deep_dive_drilldowns.conf
    - glasstable_icon_library.conf
    - inputs.conf
    - itsi_settings.conf
    - limits.conf
    - macros.conf
    - notable_event_actions.conf
    - savedsearches.conf (only in `$SPLUNK_HOME/etc/apps/SA-ITOA/local`)

    > ITSI only backs up these .conf files if they exist in a non-default directory, such as $SPLUNK_HOME/etc/apps/itsi/local. For more information, see About configuration files in ITSI. When restored, the backed up .conf file overrides the existing local version.
4. Select a daily or weekly schedule, including the time, to run the scheduled backup. The default setting runs the backup job daily at 1:00 AM in the server's local time zone.
5. Set the number of days you want to keep the backup file. The oldest backup file automatically disappears from the backup history after the retention time expires.
6. Click **Save**.

Create a full backup of ITSI

Create a full backup of IT Service Intelligence (ITSI) to make a copy of all your configuration information. 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.

Make sure to be familiar with the standard backup and restore tools and procedures used by your organization.

For more information about ITSI backups, including what gets backed up, see Overview of backing up and restoring ITSI KV store data.

Prerequisites

- You must have the itoa_admin role or the write_itsi_backup_restore capability to create a backup job.

- Before creating a backup, make sure no service templates are syncing. Check the sync status of service templates by clicking **Configuration > Service Templates** from the ITSI main menu.

Steps

1. From the ITSI main menu, click **Configuration** > **Backup/Restore**.
2. Click **Create Job > Create Backup Job**.
3. Select **Full Backup**.
4. Provide a name and description of the backup job.
5. (Optional) Enable **Include .conf files** to back up the following configuration files located in `$SPLUNK_HOME/etc/apps/SA-ITOA/local` and `$SPLUNK_HOME/etc/apps/itsi/local`:
    - alert_actions.conf
    - app_common_flags.conf
    - commands.conf
    - deep_dive_drilldowns.conf
    - glasstable_icon_library.conf
    - inputs.conf
    - itsi_settings.conf
    - limits.conf
    - macros.conf
    - notable_event_actions.conf
    - savedsearches.conf (only in `$SPLUNK_HOME/etc/apps/SA-ITOA/local`)

    > ITSI backs up these .conf files only if they exist in a non-default directory, such as $SPLUNK_HOME/etc/apps/itsi/local. For more information, see About configuration files. When restored, the backed up .conf file overrides the existing local version.
6. Click **Create**.

The backup job appears on the Backup/Restore Jobs page with the status "Queued" until the job runs. When the backup job finishes, the status changes to "Completed" and a confirmation message appears in the Messages drop-down list in Splunk Web.

You can run any completed backup job again by clicking **Edit** > **Start Backup** in the Actions column. You can also modify the completed backup job before running it again.

Next steps

To restore the backup you created, see Restore a full or partial backup of ITSI.

Create a partial backup of ITSI

Create a partial backup of IT Service Intelligence (ITSI) if you want to back up a subset of your KV store objects. You can back up services, service templates, teams, glass tables, and configuration files. When selecting one of these object types, dependent objects are automatically selected to preserve the functionality of the objects after they are restored. In some cases, you can choose whether or not to include dependent objects in the backup.

Prerequisites

- You must have the itoa_admin role or the write_itsi_backup_restore capability to create a backup job.

- Before creating a backup, make sure no service templates are syncing. Check the sync status of service templates by clicking **Configuration > Service Templates** from the ITSI main menu.

Steps

1. From the ITSI main menu, clck **Configuration** > **Backup/Restore**.
2. Click **Create Job > Create Backup Job**.
3. Select **Partial Backup**.
4. Provide a name and description of the backup.
5. (Optional) Toggle **Include .conf files** to back up the following configuration files located in `$SPLUNK_HOME/etc/apps/SA-ITOA/local` and `$SPLUNK_HOME/etc/apps/itsi/local`:
    - alert_actions.conf
    - app_common_flags.conf
    - commands.conf
    - deep_dive_drilldowns.conf
    - glasstable_icon_library.conf
    - inputs.conf
    - itsi_settings.conf
    - limits.conf
    - macros.conf
    - notable_event_actions.conf
    - savedsearches.conf (only in `$SPLUNK_HOME/etc/apps/SA-ITOA/local`)

    > ITSI backs up these .conf files only if they exist in a non-default directory, such as $SPLUNK_HOME/etc/apps/itsi/local. For more information, see About configuration files. When restored, the backed up .conf file overrides the existing local version.
6. Click **Next**.
7. On the partial backup page, select the objects to include in the backup. If you select one object type, it can cause other object types to be automatically selected if there are dependencies between the objects.
8. (Optional) Click **Change Settings** to change the objects that are selected when you select a service. By default, dependent services are selected. The KPI base searches, threshold templates, and team associated with a service are always included in the backup.
9. (Optional) Although entities are not listed in the partial backup page, you can include them in the backup file by selecting **Entities** in the Settings dialog box.
10. After making your selections, verify the objects that you selected.
11. Click **Save and Backup**.

The backup job appears on the Backup/Restore Jobs page with the status Queued until the job runs. When the backup job finishes, the status changes to Completed and a confirmation message appears in the Messages drop-down list in Splunk Web.

You can edit any partial backup job before it starts. When the backup job starts, you see a read-only view that lists the objects contained in the partial backup.

You can run any completed backup job again by clicking **Edit** > **Start Backup** in the Actions column. You can also modify the completed backup job before running it again.

What happens when you back up a service

When you back up a service, ITSI also backs up the following objects:

- KPI base searches

- Threshold templates
- Teams

You can also choose whether to back up the following associated objects:

- Dependent services
- Entities that match service entity rules
- A linked service template

If you do not choose to back up an associated object, the dependency between a service and the object breaks when you restore.

What happens when you back up a service template

When you back up a service template, all the services linked to the service template are added to the backup. If you choose to not back up a linked service, it will not exist in the restored environment.

What happens when you back up a team

When you back up a team, all the services associated with that team are added to the backup. You can deselect any services you do not want to back up.

What happens when you back up a glass table

When you back up a glass table, all of the services associated with that glass table are added to the backup. If you choose not to back up a service that the glass table depends on, any visualizations that use KPIs from the service will no longer function.

Glass table images and access control lists (ACLs) are always included in the backup when you back up a glass table.

What happens when you back up a deep dive

When you back up a deep dive, all services associated with that deep dive are also added to the backup. If you choose not to back up a service that the deep dive depends on, any KPI swimlanes from the service will no longer function if the service does not exist in the restored environment.

What happens when you back up a correlation search

When you back up a correlation search, all services associated with that correlation search are also added to the backup. If you don't back up a service the correlation search depends on, the search will still generate events with the associated service ID, but Episode Review won't be able to look up the service information.

What happens when you back up an aggregation policy

Because aggregation policies aren't directly associated with services, no services are backed up with an aggregation policy by default.

Next steps

To restore the backup you created, see Restore a full or partial backup of ITSI.

Restore a full or partial backup of ITSI

Restoring a backup of IT Service Intelligence (ITSI) merges the JSON data contained in the backup ZIP file with your existing KV store data in the following ways:

- If you added new objects since you created the backup, ITSI keeps these objects.
- If an existing object matches an object in the backup file, the existing object is replaced.
- All other existing objects are preserved.

If you restart Splunk software while a backup or restore job is in progress, the job resumes after the restart is complete. Queued jobs automatically time out if they are not completed within twelve hours. You can change the default timeout duration by updating the value of `job_queue_timeout` in the `[backup_restore]` stanza in a local version of itsi_settings.conf.

Version and deployment considerations

The restore modal displays a warning if a backup came from a deployment that's different from the current deployment. A different deployment means the backup came from another instance or search head cluster, depending on the deployment structure. Restoring from a different deployment might cause security issues, so consider rechecking the backup before proceeding.

ITSI supports backups taken from the the current version as well as three versions back. For example, version 4.6.x supports backups taken from version 4.3.x, 4.4.x, and 4.5.x. Restoring a backup to an earlier version isn't supported.

Prerequisites

- You must create a backup before you can restore it. For instructions, see Create a full backup of ITSI and Create a partial backup of ITSI.
- Make sure no service templates are syncing. Check the sync status of service templates by clicking **Configuration > Service Templates** from the ITSI main menu.
- Make sure all technology add-ons (TAs), supporting add-ons (SAs), and domain add-ons (DAs) that exist on the old system are installed on the new system.
- If you've made modifications to any add-ons on the old system, manually copy those add-ons over the new system before restoring.

Restore from a backup

You can restore from a default scheduled backup or a backup that you created.

1. On the ITSI top menu bar, click **Configuration** > **Backup/Restore** and find the backup that you want to restore from.
2. Click **Edit** > **Restore Backup**.
3. If you're restoring a scheduled backup, select a saved backup from the list. If you're restoring a created backup, go to the next step.
4. Click **Start Restore**. "Restore from" is prepended to the backup name in the jobs list. A message stating that the restore job successfully completed appears in the messages dropdown list in Splunk Web.
5. If you restored a backup that contains configuration files, you must restart your Splunk Enterprise instance.

Restore from a backup ZIP file

You can download any backup ZIP file that is created when you run a backup job in the UI and then restore from that backup ZIP file using the Backup/Restore Jobs UI. The maximum file size supported for uploading a backup file is 500 MB.

Perform the following steps to download a backup ZIP file:

1. On the ITSI top menu bar, click **Configuration** > **Backup/Restore** and find the backup file that you want to download.
2. Click **Edit** > **Download Backup**. If you are restoring a scheduled backup, select a saved backup from the list. If you are restoring a created backup, the backup file displays.
3. Save the file. The backup ZIP file downloads to your local machine.

Perform the following steps to restore from a downloaded backup ZIP file:

1. On the Backup/Restore Jobs page, click **Create Job > Create Restore Job**.
2. Provide a name and an optional description of the backup.
3. Click **Choose File** and select the previously downloaded backup ZIP file that you want to restore from.
4. (Optional) Toggle **Include .conf files** to restore any configuration files included in the backup.
5. Click **Create**.
   ITSI uploads the backup ZIP file and the new restore job appears in the Backup/Restore Jobs list. A message stating that the restore job has successfully completed appears in the Message drop-down list in Splunk Web.
6. (Optional) If you restore from a backup that contains .conf files, you must restart Splunk software.

How teams are restored

Team permissions are retained when teams are restored. The roles assigned to the teams must exist on the system that the backup is restored to. For example, suppose a restore creates teams called "HR" and "Finance", which have read/write access for the `hr_admin` and `finance_admin` roles. If the current system doesn't have these roles, only the `itoa_admin` role can access these teams. If the roles assigned to the teams don't exist on the system, you can create them either before or after restoring.

kvstore_to_json.py operations in ITSI

ITSI provides a `kvstore_to_json.py` script that lets you backup/restore ITSI configuration data, perform bulk service KPI operations, apply time zone offsets for ITSI objects, and regenerate KPI search schedules.

Usage options

The `kvstore_to_json.py` script is located in `$SPLUNK_HOME/etc/apps/SA-ITOA/bin/`.

The `kvstore_to_json.py` script has these 4 modes:

Mode 1: Backup and restore operations
Mode 2: Bulk service KPI operations.
Mode 3: Time zone offset operations.
Mode 4: Regenerate KPI search schedules.

To view all `kvstore_to_json.py` usage options, specify the `-h` option.

```
[root@myserver splunk]# ./bin/splunk cmd python etc/apps/SA-ITOA/bin/kvstore_to_json.py -h

Usage: kvstore_to_json.py [options]

Options:
  -h, --help            show this help message and exit
  -s SPLUNKDPORT, --splunkdport=SPLUNKDPORT
                        splunkd port. If no option is provided, we will
                        default to '8089'
  -u USERNAME, --username=USERNAME
                        Splunk username
  -p PASSWORD, --password=PASSWORD
                        Splunk password
  -n, --no-prompt       Use this option when you want to disable the prompt
                        version of this script
  -v, --verbose         Use this option for verbose logging
  -f FILE_PATH, --filepath=FILE_PATH
                        The full path of a directory. Usage depends on mode.
                        When importing backed up data of version 1.2.0, this
                        could be a file or a set of files. When working with
                        service KPIs, this is a directory containing
                        input.json on entry and output.json on exit.
  -m MODE, --mode=MODE  Specify the mode of operation - what kind of
                        operations to perform. Mode is set to:         1 - for
                        backup/restore operations.         2 - for service KPI
                        operations.

  Backup and restore operations. This is mode 1.:
    Use this option when you want to perform backup/restore operations.

    -i, --importData    Use this option when you want to upload data to the KV
                        Store. When importing data from version 1.2.0, you can
                        use filepath as wildcard to upload data from more than
                        one file. However, filepath must be within quotes if
                        it is being used as a wildcard
    -d, --persist-data  Use this option when you want to persist existing
                        configuration in KV Store during import. NOTE:
                        Applicable only if importData option is used
    -y, --dry-run       Use this option when you want only to list objects for
                        import or backup
    -a, --conf-file     Use this option when you want to back up .conf files.
    -b BR_VERSION, --base-version=BR_VERSION
                        The original ITSI application version user intends to
                        backup/restore from.
    -e DUPNAME_TAG, --dupname-tag=DUPNAME_TAG
                        Automatically rename all the duplicated service or
                        entity names from restoring with a tag. If this option
                        is not set, the restoring will halt if duplicate names
                        are detected. The default tag is:
                        _dup_from_restore_<epoch_timestamp>

  Service KPI operations. This is mode 2.:
    Use this option when you want to get/create/update/delete KPIs for
    existing services.

    -g, --get           For input, specify a list of service keys with the
                        keys of KPIs to retrieve. Expected format: [{_key:
                        <service key>, kpis: [{_key: <KPI key>}]]. Specify []
                        to get all KPIs from all services. Specify [{_key:
                        <service key>, kpis: []] to get all KPIs from a
                        service. Assumes input is available in
                        file_path/input.json
    -c, --create        For input, specify a non-empty list of service keys
                        with their KPIs list. Expected format: [{_key:
                        <service key>, kpis: [{_key: <KPI key>, <rest of KPI
                        structure>}]]. Note that only existing services could
                        be updated with new KPIs only with this option.
                        Assumes input is available in file_path/input.json
    -t, --update        For input, specify a non-empty list of service keys
                        with their KPIs list. Expected format: [{_key:
                        <service key>, kpis: [{_key: <KPI key>, <rest of KPI
                        structure>}]]. Note that only existing services and
                        existing KPIs could be updated using this option.
                        Assumes input is available in file_path/input.json
    -r, --delete        For input, specify a list of service keys with the
                        keys for the KPIs to delete.Expected format: [{_key:
                        <service key>, kpis: [{_key: <KPI key>}]]. Assumes
                        input is available in file_path/input.json

Timezone offset operations. This is mode 3.:
    Use this option when you want to adjust timezone settings for time
    sensitive fields on object configuration.

    -q IS_GET, --is_get=IS_GET
                        For input, specify if you are trying to read objects
                        or update their timezone offsets.
    -o OBJECT_TYPE, --object_type=OBJECT_TYPE
                        For input, specify a valid object type that contains
                        time sensitive configuration. This option will apply
                        offset to all objects on this type unless scoped to a
                        specific object using object_key parameter.Supported
                        object types are: "maintenance_calendar" for
                        maintenance windows, "service" for Services/KPIs
                        (threshold policies)
    -k OBJECT_TITLE, --object_title=OBJECT_TITLE
                        For input, specify an optional object title of object
                        type that contains time sensitive configuration. Using
                        this option will cause the offset change to only apply
                        to that object.
    -z OFFSET_IN_SEC, --offset_seconds=OFFSET_IN_SEC
                        For input, specify the offset to apply in seconds as a
                        positive or negative number. This offset should be the
                        number of seconds that you want to add or subtract
                        from the current value.
```

Running the script in a search head cluster environment

When running the `kvstore_to_json.py` script in a replicated KV store environment, the script works on any member of the search head cluster. It does not require execution on the captain.

It is best practice to run the script on the MongoDB captain, which might be different than the captain of the search head cluster.

Backup and restore operations (mode 1)

You can no longer perform partial backups using the kvstore_to_json.py mode 1 option. Use the partial backup workflow in the UI instead. For information, see Create a partial backup in the *Administer Splunk IT Service Intelligence* manual.

Use replacement options

The partial restore rules schema provides replacement options, which let you change the name of an object when you run a partial backup/restore operation. Replacement options are useful for renaming objects when moving from a test environment to a production environment.

For example, to backup a service called test_database_service, but change the name to database_service; and to backup a deep dive called test_database_deep_dive, and change the name to database_deep_dive, you would create a `rules.json` file that contains the following:

```
[
 {
    "object_type": "service",
    "title_list": "^test_database_service$",
    "replacement_rules": [ {
        "replacement_key": "title",
        "replacement_type": "replace",
        "replacement_string": "database_service",
        "replacement_pattern": "^test_database_service$"
    }]
  },
  {
    "object_type": "entity",
    "title_list": ["10.12.*", "*host_database*"]
  },
  {
    "object_type": "deep_dive",
    "title_list": "^test_database_deep_dive$",
    "replacement_rules": [ {
        "replacement_key": "title",
        "replacement_type": "replace",
        "replacement_string": "database_deep_dive",
        "replacement_pattern": "^test_database_deep_dive$"
    }]
  }
]
```

Service KPI operations (mode 2)

`kvstore_to_json.py` mode 2 options let you run bulk operations on KPIs, including get (`-g`), create (`-c`), update (`-t`), and delete (`-r`). Use these options to replicate, edit, and copy KPIs to multiple services, for example, when moving your ITSI deployment from a test environment to a production environment.

All service KPI options require you to specify the mode 2 parameter `-m 2`. You must also specify the file path `-f` parameter as the full path to the directory containing the `input.json` file.

Before you can run service KPI operations, you must create an `input.json` file in the destination directory. The script accepts data input from `input.json` and sends data output to an `output.json` file that the script creates in the same directory.

All service KPI operations, except get `-g`, require you to specify service and/or KPI keys. You can retrieve these keys using the `-g` option in `output.json`.

See the kvstore_to_json.py help - h option for proper input.json and command syntax.

Get service and KPI keys

Use the get `-g` option to retrieve service and KPI data in JSON format, including service and KPI keys.

1. Create an `input.json` file in the destination directory.

   ```
   mkdir <directory_containing_input.json>
   touch input.json
   ```
2. Edit `input.json`: Add `[]` to the file to retrieve JSON data for all services and KPIs, or add specific service and kpi keys to the file to retrieve JSON data for those services and KPIs only. For example:

   ```
   [{"_key": "<service_key>", "kpis": [{"_key": "<kpi_key>"}] } ].
   ```
3. Run the `kvstore_to_json` script using the get `-g` option. Name the full path to the directory containing the `input.json` file as the file path `-f` parameter. For example:

   ```
   cd $SPLUNK_HOME
   bin/splunk cmd python kvstore_to_json.py -u admin -p changeme -m 2 -g -f
   <directory_containing_input.json> -n
   ```
4. Review the contents of `output.json` to identify service and KPI keys. For example:

   ```
   [
       {
           "_key": "669c5cec-a492-419d-8659-95a185b4dc5c",
           "kpis": [
               {
                   "_key": "f017cc7b2e67f2b3b9152146",
   ...
   }
   ]
   }
   ]
   ```

Create KPIs

Use the `-c` option to create new KPIs.

1. Edit `input.json` to specify the service key of the service for which you want to create the KPI.
2. Add KPI keys for the KPIs that you want to add to the service and any key-value pairs belonging to the KPI that you want to include in the KPI definition. Leave the key field for each KPI empty for ITSI to auto generate it. For example:

   ```
   [
       {
           "_key": "<service_key>",
           "kpis": [
               {
                   "title": "<title_of_kpi_to_create>",
   ...
   }
   ]
   }
   ]
   ```

Update KPIs

Use the `-t` option to update KPIs.

In `input.json` specify the service and KPI key for each KPI, and any other key/value pair data that you want to update for the KPI.

Delete KPIs

Use the `-r` option to delete KPIs.

In `input.json`, specify service and kpi keys for all KPIs that you want to delete.

Caution: Make sure to properly validate your JSON input. While the `kvstore_to_json` script does provide some schema validation, incorrect JSON formatting can cause errors.

Time zone offset operations (mode 3)

The `kvstore_to_json.py` mode 3 option lets you apply a time offset for time-sensitive fields in object configurations. You can use this option to correct time zone discrepancies for the following object types:

- `maintenance_calendar`: Sets an offset for maintenance window start and end times.
- `service`: Sets an offset for the KPI threshold time policies within a service.
- `kpi_threshold_template`: Sets an offset for a KPI threshold template. After running the command to set an offset for a KPI threshold template, you must run the command again for each service that uses the KPI threshold template and set the same offset so that they are in sync.

Apply time zone offset

Run the following command to set an offset for one of the supported object types.

> If you use the command to set an offset for a KPI threshold template, you must run the command again for each service that uses the KPI threshold template and set the same offset so that they are in sync.

1. Run `kvstore_to_json.py`, where `-o` is the object type, `-k` is the title of the specific object, and `-z` is the specific time zone offset in seconds. For example:

   ```
   cd $SPLUNK_HOME
   bin/splunk cmd python etc/apps/SA-ITOA/bin/kvstore_to_json.py -m 3 -o service -k "Database Service"
   -z 1800
   ```
2. Enter the requested information at the prompts (default interactive mode only). For example:

   ```
   >> Enter the splunkd port number OR press the enter key to use [8089] > 8089
   >> Enter splunk username or press enter to use "admin" > admin
   >> Enter splunk password for "admin" >
   ```
3. The script applies the time zone offset to the specified object. For example:

   ```
   1 object(s) match request
   Applying timezone change on requested object(s): [u'Database Service']
   Timezone offset has been applied on the objects requested.
   ```

ITSI time-sensitive configurations are normalized to UTC.

Regenerate KPI search schedules (mode 4)

The `kvstore_to_json.py` mode 4 option regenerates the search schedules for your KPIs. Use this command if you have set your KPI saved search schedules to run at the same time in itsi_settings.conf. Run this command to reset the search schedules of all your KPIs to use the new search schedule. See Synchronize KPI searches in ITSI for more information.

1. Run `kvstore_to_json.py` in mode 4.
   For example:

   ```
   cd $SPLUNK_HOME
   bin/splunk cmd python etc/apps/SA-ITOA/bin/kvstore_to_json.py -m 4
   ```
2. Enter the requested information at the prompts (default interactive mode only).
3. You'll see the following message after the KPI search schedules have been reset:
   ```
   Retrieving KPIs to reset their saved search scheduling
   Saving updated KPI scheduling
   Done.
   ```

ITSI indexes

Overview of ITSI indexes

IT Service Intelligence (ITSI) implements custom indexes for event storage. All ITSI indexes are listed in `$SPLUNK_HOME/etc/apps/SA-IndexCreation/default/indexes.conf`.

- In a single instance deployment, the installation of ITSI creates the indexes in the default path for data storage.
- In a Splunk Cloud Platform deployment, customers work with Splunk Support to set up, manage, and maintain their cloud index parameters. See Manage Splunk Cloud Platform indexes in the *Splunk Cloud Platform Admin Manual*.
- In a distributed deployment, create the indexes on all Splunk platform indexers or search peers.

For detailed examples of configuring indexes, see indexes.conf.example in the Splunk Enterprise *Admin Manual*.

The following table describes the indexes available in `$SPLUNK_HOME/etc/apps/SA-IndexCreation/default/indexes.conf`:

Index	Description
itsi_summary	An events index that stores the results of scheduled KPI searches. Summary indexing lets you run fast searches over large data sets by spreading out the cost of a computationally expensive report over time.
itsi_summary_metrics	A metrics index that stores the results of scheduled KPI searches. Every KPI is summarized in both the itsi_summary events index and the metrics index. This index improves the performance of the searches dispatched by ITSI, particularly for very large environments.
anomaly_detection	An internal index used to support trending and cohesive anomaly detection in ITSI.
itsi_tracked_alerts	Stores active raw notable event data.
itsi_notable_audit	Stores all audit events for episodes, including actions, comments, status change, and owner change.
itsi_notable_archive	Stores episode metadata (tags and comments) that has been moved from the KV store after a default 6 month retention period, which begins when you close an episode in the UI. Moving data from the KV store removes extraneous data and helps improve performance.
itsi_grouped_alerts	Stores active episode data.
snmptrapd	Stores events coming in from SNMP traps. For more information, see Ingest SNMP traps into ITSI.
itsi_import_objects	Stores events indexed from a manual entity or service import from a CSV file.
itsi_im_meta	Optional index that stores Kubernetes metadata.
itsi_im_metrics	Stores entity data for entity discovery in ITSI.

ITSI summary index reference

Fields in the IT Service Intelligence summary index (itsi_summary) are generally defined in families. For example, all fields that begin with `indexed_*` are defined as indexed extractions and thus can be filtered more quickly. `alert_*` fields are generally properties replicated from the KPI as well as the information related to the time series. `is_*` fields are boolean fields that only ever have the values of 0 or 1.

KPI data points have evolved extensively throughout the history of ITSI. As a result, a lot of extraneous fields have been carried forward to avoid upgrade issues. These fields are marked deprecated and should not ever be referenced in any search.

Classes of fields in the summary index

Type of field	Description
Service aggregate	Represents the value of the KPI for the service at a given time along with its evaluated severity. This field exists for every time, even if there is no data. There is exactly 1 for each period of the KPI.
Entity-level	Represents the value of the KPI for a particular entity at a given point in time along with its evaluated severity. There are 0 to n of these data points. If there is no data, there are no entity-level data points even if the KPI is split by entity.
Max severity	Represents the most severe KPI data point among service aggregate and all entity-level data points for a given time. Its value is random if multiple data points have the same severity. This data point exists solely for the purpose of evaluating score events. It always exists for every time, even if there's no data. There is exactly 1 event for each period of the KPI.
Health score	Represents the health score of a given service at a given time. There is exactly 1 event for each service every minute regardless of the number or period of KPIs within the service.
Composite multi-KPI alert score	Represents the health score of a composite multi-KPI alert. The fields only exist to support multi-KPI alerting. They exist in the summary index because multi-KPI alert scores are calculated at the same time as health scores.

Summary index fields

The following table provides descriptions and sample values for each field in the summary index.

Field	Sample value	Description
alert_color	#99D18B	A hex code for the color of the severity of the data point.
alert_level	2	An integer indicating the severity of the data point. This is the main property for severity and should be the one used for filtering and grouping. Other properties related to the severity are only there for convenience and may be deprecated in a future release.
alert_period	5	The period, in minutes, at which the data point is expected in the summary index. For example, if "5", there should be 1 event every 5 minutes. This field translates to the cron schedule of the KPI.
alert_severity	normal	The text label for the severity of alert_level.
alert_value	1	The actual aggregated numeric value of the KPI for this data point. This field is used for all graphing and display of the KPI value.
color	#99D18B	Duplicate of alert_color.
entity_key	service_aggregate	The key in the entity database of the entity to which this data point belongs, if defined. If "N/A" then the value refers to a pseudo entity. If "service_aggregate" then the value refers to the Service Aggregate data point for the KPI. On a maximum severity event this field and the entity_title can tell you which KPI data point was selected as the Max Severity data point.
entity_title	service_aggregate	The title of the entity in the entity database. In the case of pseudo entities, the title of the entity as found in the data. Will be "service_aggregate" if the value refers to the Service Aggregate data point for the KPI. On a maximum severity event this field and the entity_key can tell you which KPI data point was selected as the Max Severity data point.

Field	Sample value	Description
gs_kpi_id	efd9c9eeb482a9cfde9a8e2d	Duplicate of itsi_kpi_id. Deprecated.
gs_service_id	b5946968-dfa8-4aa2-a393-7163d2576c6e	Duplicate of itsi_service_id. Deprecated.
health_score	100.0	Duplicate of severity_value.
host	ip-10-202-0-160.ec2.splunkit.io	The originating hostname or IP address the KPI saved search was dispatched from.
index	itsi_summary	The index the data point is stored in.
indexed_is_service_aggregate	1	Indexed field of is_service_aggregate. Always filter against this field instead of the non-indexed version. You must use "::" and not "=" for it to make a difference in filtering.
indexed_is_service_max_severity_event	0	Indexed field of is_service_max_severity_event. Always filter against this field instead of the non-indexed version. You must use "::" and not "=" for it to make a difference in filtering.
indexed_itsi_kpi_id	efd9c9eeb482a9cfde9a8e2d	Indexed field of itsi_kpi_id. Always filter against this field instead of the non-indexed version. You must use "::" and not "=" for it to make a difference in filtering.
indexed_itsi_service_id	b5946968-dfa8-4aa2-a393-7163d2576c6e	Indexed field of itsi_service_id. Always filter against this field instead of the non-indexed version. You must use "::" and not "=" for it to make a difference in filtering.
info_max_time	1572460080.000	The latest time bound of the dispatched saved search that resulted in this data point. Added by the summary indexing process, mainly useful for forensics.
info_min_time	1572460020.000	The earliest time bound of the dispatched saved search that resulted in this data point. Added by the summary indexing process, mainly useful for forensics.
info_search_time	1572460080.844	The actual time the saved search that resulted in this data point was dispatched. Added by the summary indexing process, mainly useful for forensics.
is_entity_defined	0	"0" if the entity described in entity_title and entity_key is a pseudo entity, "1" if it's a defined entity. This field is better to filter against than entity_key!="N/A", though it is still 0 for service-level data.
is_entity_in_maintenance	0	"0" if the entity described in entity_key is not in maintenance at the time the data point was taken, "1" if it was. A pseudo entity can never be in maintenance. If an entity is maintenance, its value is not included in the Service Aggregate calculation unless every entity in the service is in maintenance. For more information, see Schedule maintenance downtime in ITSI.
is_filled_gap_event	0	An indication of whether there were any data gaps that were filled with an specified value. If "1", there was a gap that was filled. For more information about filling data gaps, see Configure KPI monitoring calculations in ITSI.
is_service_aggregate	1	"0" if the data point is from an entity, "1" if it's from the Service Aggregate calculation. Never filter against this field. Use the indexed version instead.

Field	Sample value	Description
is_service_in_maintenance	0	"0" if the service described in itsi_service_id is not in maintenance at the time the data point was taken, "1" if it was.
is_service_max_severity_event	0	"0" if this is a normal KPI data point, "1" if it's the Max Severity event. Never filter against this field. Use the indexed version instead.
itsi_kpi_id	efd9c9eeb482a9cfde9a8e2d	The ID or key of the KPI to which this KPI data point belongs. Never filter against this field. Use the indexed version instead.
itsi_service_id	b5946968-dfa8-4aa2-a393-7163d2576c6e	The ID or key of the service to which this KPI data point belongs. Never filter against this field. Use the indexed version instead.
kpi	Network Txmt KBps	The name of the KPI at the time the data point was taken. For display purposes only.
kpi_name	Network Txmt KBps	The name of the KPI at the time the data point was taken.
kpi_urgency	11	The importance value configured for the KPI at the time the data point was taken.
kpibasesearch	5d75b61e6e651456557ab604	Only defined on data points generated by a shared base search. This is the key of the shared base search that made this KPI data point.
kpiid	efd9c9eeb482a9cfde9a8e2d	Deprecated. Duplicate of itsi_kpi_id. Never use.
linecount	1	The number of lines an event contains before it's indexed.
python.version	python3	The current Python version.
qf	--	Quick filter. This field is only populated when ITSI needs to put something in maintenance to make maintenance searches perform better.
scoretype	service_health	The type of health score the event contributes to. Used to distinguish service health score events from composite health score events. This field is only present in Health Score type KPI data points. • For a composite multi-KPI event the value is compositekpi_health. • For a service health score event the value is service_health.
search_name	disabled_kpis_healthscore_generator	The name of the saved search that made the KPI data point. Added by the summary indexing process, mainly useful for forensics.
search_now	1572460080.000	The effective "now" used when the saved search was dispatched. Added by the summary indexing process, mainly useful for forensics.
sec_grp	default_itsi_security_group	The team the service belongs to. For more information, see Overview of teams in ITSI.
service	Middleware	The name of the service.

Field	Sample value	Description
service_name	Middleware	Duplicate of service.
serviceid	b5946968-dfa8-4aa2-a393-7163d2576c6e	Deprecated. Duplicate of itsi_service_id.
severity_label	normal	The text label for the severity of severity_value.
severity_value	100.0	The numeric value of the service health score. This field is used for all graphing and display of the service health score value.
source	disabled_kpis_healthscore_generator	The search that populates the summary index with state values for the KPI.
sourcetype	stash	Specifies the format of the data input from which the event originates. Set by the summary indexing process to "stash" for licensing purposes.
splunk_server	ip-10-202-0-198.ec2.splunkit.io	The name of the Splunk server containing the event. Useful in a distributed Splunk environment.
timeendpos	25	The position in the raw event string at which the timestamp ends.
timestartpos	0	The position in the raw event string at which the timestamp starts.
urgency	11	The importance value configured for the KPI at the time the data point was taken. Duplicate of kpi_urgency.

ITSI metrics summary index reference

The IT Service Intelligence (ITSI) metrics summary index, `itsi_summary_metrics`, is a metrics-based summary index that stores KPI data. The index is a metrics version of the ITSI events summary index. For more information, see ITSI summary index reference.

As of ITSI version 4.6.0, each KPI is summarized into both the events summary index and the metrics summary index. Service health score values are calculated using metrics, and KPI and service health score tiles on the Service Analyzer are rendered using metrics. The metrics summary index creates a more responsive UI experience by increasing the performance of the searches dispatched by ITSI. In future releases, additional UI elements will be converted to use the `mstats` syntax.

The metrics summary index provides the following performance improvements:

- Service Analyzer rendering is 28% faster
- Service topology rendering is 18% faster

For more information about metrics indexes, see Metrics indexes in the Splunk Enterprise *Metrics Manual*.

Classes of fields in the metrics summary index

Type of field	Description
Service aggregate	Represents the value of the KPI for the service at a given time along with its evaluated severity. This field exists for every time, even if there is no data. There is exactly 1 for each period of the KPI.
Entity-level	

Type of field	Description
	Represents the value of the KPI for a particular entity at a given point in time along with its evaluated severity. There are 0 to n of these data points. If there is no data, there are no entity-level data points even if the KPI is split by entity.
Max severity	Represents the most severe KPI data point among service aggregate and all entity-level data points for a given time. Its value is random if multiple data points have the same severity. This data point exists solely for the purpose of evaluating score events. It always exists for every time, even if there's no data. There is exactly 1 event for each period of the KPI.
Health score	Represents the health score of a given service at a given time. There is exactly 1 event for each service every minute regardless of the number or period of KPIs within the service.

Metrics summary index fields

The following table provides descriptions and sample values for each field in the summary index.

Field	Sample value	Description
alert_period	5	The period, in minutes, at which the data point is expected in the summary index. For example, if "5", there should be 1 event every 5 minutes. This field translates to the cron schedule of the KPI.
entity_key	service_aggregate	The key in the entity database of the entity to which this data point belongs, if defined. If "N/A" then the value refers to a pseudo entity. If "service_aggregate" then the value refers to the Service Aggregate data point for the KPI. On a maximum severity event this field and the entity_title can tell you which KPI data point was selected as the Max Severity data point.
entity_title	service_aggregate	The title in the entity database of the entity to which this data point belongs. In the case of pseudo entities, the title of the entity as found in the data.
host	ip-10-202-0-160.ec2.splunkit.io	The originating hostname or IP address the KPI saved search was dispatched from.
index	itsi_summary_metrics	Stores the name of the index, which will always be itsi_summary_metrics. A standard field in Splunk software.
info_max_time	1572460080.000	The latest time bound of the dispatched saved search that resulted in this data point. Added by the summary indexing process, mainly useful for forensics.
info_min_time	1572460020.000	The earliest time bound of the dispatched saved search that resulted in this data point. Added by the summary indexing process, mainly useful for forensics.
info_search_time	1572460080.844	The actual time the saved search that resulted in this data point was dispatched. Added by the summary indexing process, mainly useful for forensics.
is_backfilled_event	1	Indicates whether a result is is from a backfill operation.
is_entity_defined	0	"0" if the entity described in entity_title and entity_key is a pseudo entity, "1" if it's a defined entity. This field is better to filter against than entity_key!="N/A", though it is still 0 for service-level data.
is_entity_in_maintenance	0	"0" if the entity described in entity_key is not in maintenance at the time the data point was taken, "1" if it was. A pseudo entity can never be in maintenance. If an entity is in maintenance, its value is not included in the Service

Field	Sample value	Description
		Aggregate calculation unless every entity in the service is in maintenance. For more information, see Schedule maintenance downtime in ITSI.
is_filled_gap_event	0	An indication of whether there were any data gaps that were filled with an specified value. If "1", there was a gap that was filled. For more information about filling data gaps, see Configure KPI monitoring calculations in ITSI.
is_null_alert_value	1	Indicates if the the KPI score or service health score value was actually N/A previously. This field exists because values stored under a metric_name can only be integers. Mainly for forensics to better reflect the true alert_value.
is_service_aggregate	1	"0" if the data point is from an entity, "1" if it's from the Service Aggregate calculation. Never filter against this field.
is_service_disabled	0	Indicates if the service was in a disabled state at into_max_time OR has been disabled at the time the data point was taken.
is_service_in_maintenance	0	"0" if the service described in itsi_service_id is not in maintenance at the time the data point was taken, "1" if it was.
is_service_max_severity_event	0	"0" if this is a normal KPI data point, "1" if it's the Max Severity event. Never filter against this field.
itsi_kpi_id	efd9c9eeb482a9cfde9a8e2d	The ID or key of the KPI to which this KPI data point belongs. Never filter against this field.
itsi_service_id	b5946968-dfa8-4aa2-a393-7163d2576c6e	The ID or key of the service to which this KPI data point belongs. Never filter against this field.
itsi_team_id	default_itsi_security_group	The team the service belongs to. For more information, see Overview of teams in ITSI.
kpi	Network Txmt KBps	The name of the KPI at the time the data point was taken. This is for display purposes only.
kpi_importance	11	The importance value configured for the KPI at the time the data point was taken.
kpi_base_search	5d75b61e6e651456557ab604	Only defined on data points generated by a shared base search. This is the key of the shared base search that made this KPI data point.
metric_name:alert_level	2	An integer indicating the severity of the data point. This is the main property for severity and should be the one used for filtering and grouping. Other properties related to the severity are only there for convenience and may be deprecated in a future release.
metric_name:alert_value	1	The actual aggregated numeric value of the KPI for this data point. This field is used for all graphing and display of the KPI value.
metric_name:service_health_score	100.0	The numeric value of the service health score. This field is used for all graphing and display of the service health score value.
scoretype	service_health	The type of health score the event contributes to. Used to distinguish service health score events from composite health score events. This field is only present in Health Score type

Field	Sample value	Description
		KPI data points. • For a composite multi-KPI event the value is compositekpi_health. • For a service health score event the value is service_health.
search_name	disabled_kpis_healthscore_generator	The name of the saved search that made the KPI data point. Added by the summary indexing process, mainly useful for forensics.
search_now	1572460080.000	The effective "now" used when the saved search was dispatched. Added by the summary indexing process, mainly useful for forensics.
source	disabled_kpis_healthscore_generator	The search that populates the summary index with state values for the KPI.
sourcetype	stash	Specifies the format of the data input from which the event originates. Set by the summary indexing process to "stash" for licensing purposes.

Troubleshooting

Troubleshoot ITSI permissions, teams, backups, and restores

Here are some common issues related to ITSI permissions and capabilities, backups, and restores and how to resolve them.

User assigned a custom role can't view objects

A user is assigned a custom role can't view objects in ITSI

Resolution

Make sure you've fully completed steps 1-4 in Create a custom role in ITSI.

User has itoa_admin role but can't view objects

A user is assigned the itoa_admin role but is unable to read services or any other objects on their corresponding lister pages.

Resolution

By default, the itoa_admin role ships with the itoa_analyst and itoa_user roles. The itoa_user ships with read capabilities for ITOA objects like services, entities, glass tables, and deep dives. Make sure these capabilities haven't changed.

Unable to create an external ticket

A user is assigned the itoa_analyst role with the create_external_ticket capability. However, they're unable to create an external ticket.

Resolution

A restriction in Splunk Enterprise means the user needs the itoa_admin role, which inherits from the admin role.

"Access denied. You do not have permission to create this object."

You see access denied errors when attempting to create objects.

Cause

ITSI relies on the fact that your admin role inherit from the roles defined in $SPLUNK_HOME/etc/apps/itsi/default/authorize.conf:

```
[role_admin]
importRoles = itoa_admin;itoa_analyst;itoa_user;power;user
```

Resolution

Use btool to check system/local/authorize.conf:

```
 $SPLUNK_HOME/bin/splunk btool authorize list role_admin --debug
```

You might have redefined the admin role inheritance in system/local/authorize.conf, or in other apps. If this is the case, add the inheritances added from the UI or through the configuration file.

Default scheduled backup not running

After a fresh install or migration, the default scheduled backup isn't running at 1:00 am.

Cause

The backup runs at 1:00 am in the timezone of the server. If your local timezone is different than the server's, it might appear to run at a different time.

Alternatively, the modular input for the default scheduled backup runs at every restart, and every hour after that. It's possible to see a maximum of one-hour delays. For example, if the next scheduled time is 1:00am, the modular input runs at 12:45am and 1:45am, the backup will start at 1:45am.

Failed to fetch backup information preview

ITSI fails to fetch backup information preview with ID: <backup_id>

Resolution

Check `https://localhost:8089/servicesNS/nobody/SA-ITOA/backup_restore_interface/backup_restore/preview/<backup_id>` and see if the information exists for the given backup ID.

Failed to upload a backup file

ITSI fails to upload the selected backup file.

Resolution

- Check the network tab of the browser to see if there's a failed request. Check if you can create a restore job by clicking **Create**.
- Make sure the file is valid and not corrupted.
- Get a new backup file from the backup job. Download this file and try to upload it for restore.

Global team is gone after upgrade

The global team is no longer present after an ITSI upgrade.

Resolution

All services in ITSI must be assigned to a team. If migration fails with the error `Failed to import Team settings`, you can manually run the Python script called `itsi_reset_default_team.py`. The script manually creates the Global team in the

KV store which completes the migration.

To run the script, perform the following steps:

1. Run the following commands on any search head in your ITSI deployment:

   ```
   cd $SPLUNK_HOME/etc/apps/SA-ITOA/bin
   $SPLUNK_HOME/bin/splunk cmd python itsi_reset_default_team.py
   ```
2. Provide the splunkd port number and your Splunk username and password when prompted. After the script finishes successfully, the Global team is created in the KV store.
3. Restart your Splunk software.

Check the ITSI logs

IT Service Intelligence log files have a prefix of `itsi_`.

- IT Service Intelligence search command logs are located in `$SPLUNK_HOME/var/run/splunk/dispatch/<session_id>/itsi_search.log`.
- All other ITSI logs are located in `$SPLUNK_HOME/var/log/splunk`.

All ITSI logs have a source type of `itsi_internal_log` to make them easy to search.

Steps

1. Run the following Splunk search to search ITSI logs:

   ```
   index = _internal sourcetype=itsi_internal_log
   ```
2. Click the **source** field under Selected Fields to see specific log files.

> For Windows deployments, the ITSI search command log, itsi_search.log, cannot be searched in Splunk Web. You must open the file on the Windows host using a text editor.

Dashboards

Use the ITSI Health Check dashboard

The **ITSI Health Check** dashboard provides basic statistics about your ITSI environment.

Dashboard panels

Panel	Description
Splunk Server Information	Basic server information for each host.
ITSI Migration Status	The current version of ITSI and the ITSI KV store. These versions should be the same.
ITSI Upgrade Readiness	Checks whether any service templates are currently syncing. If so, it is not safe to upgrade. Click **Configure** > **Service Templates** to see the current sync status of your service templates.
Basic ITSI Information	For each host, lists the number of services, searches, and entities, as well as KV store and HEC information.
KPI Base Search Usage Summary	The number of KPIs using each base search.
KV Store Collections	All ITSI KV store collections, the number of objects in each collection, acceleration information, and the collection size. If a collection is approaching the limit, consider trimming it to retain three months or less of metadata. For instructions, see Trim down notable event KV store collections in the *Event Analytics* manual.
Concurrent Searches	All ITSI searches currently running.
Interesting Indexes	All ITSI indexes and their statistics.
Interesting Searches	Real-time searches that ITSI runs. `itsi_event_grouping` handles event grouping for notable event aggregation policies and is stored in savesearches.conf. `itsi_mad_context` and `itsi_mad_cohesive_context` handle metric anomaly detection and are stored in /SA-ITSI-MetricAD/local/savedsearches.conf once KPI anomaly detection is turned on. If any search jobs are failed or not running, this could indicate a problem.
KPI Performance	Basic performance information for each KPI in your ITSI instance. Any failed or skipped searches indicate a problem. The runtime headroom percentage indicates how much time has been used up out of the search's frequency. A headroom percentage close to 100 is best, and a value closer to 0 indicates a problem.
Saved search Error Messages	Lists names of saved search with error messages that include details about count, average run time, message key, and error messages.
Not Executed Searches (In last 1 hour)	The number of searches that were not executed in the last hour.
Refresh Queue Runtimes	Statistics for the refresh queue. The refresh queue ensures data integrity and eventual consistency of your ITSI configuration. It runs as a single instance.

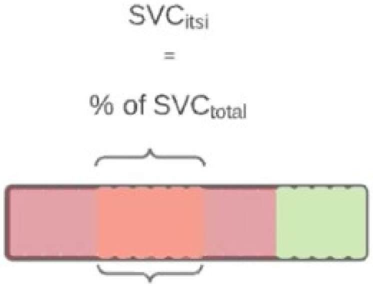
Panel	Description
Refresh Queue Failed Jobs	The number of failed jobs in the refresh queue. Click a failed job to drill down to the logs.
ITSI Log Messages (deduplicated)	Warning and error messages in the ITSI logs. The messages are deduplicated so you won't see the same error multiple times.
Check for Duplicate Entity Aliases	Lists the entity aliases (field-value pairs) identifying more than one entity.

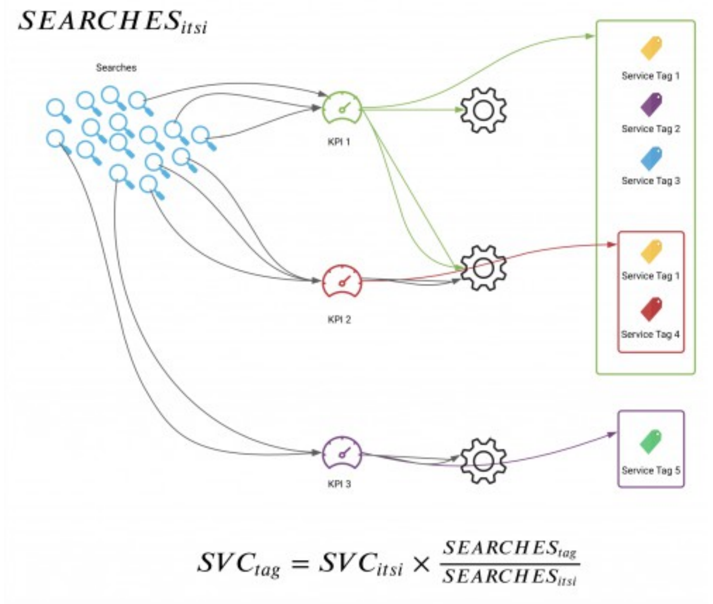
Use the ITSI SVC Statistics dashboard

The **ITSI SVC Statistics** dashboard enables you to view the total Splunk Virtual Core (SVC) usage for ITSI. An SVC is a unit of capabilities in Splunk Cloud Platform that includes compute, memory, and I/O resources. To learn more about SVCs, see Monitor current SVC usage of your workload-based subscription in the Splunk Cloud Platform *Admin* manual.

The dashboard can be used as a reference to view the SVC breakdown for your ITSI instance by service tags. Clone and configure this dashboard to generate other breakdowns based on your organization's specific requirements.

Dashboard panels

Panel	Description
ITSI SVC Entitlement	Displays the number of SVCs assigned to your subscription per your license entitlement that are specific to ITSI. To display the ITSI SVC entitlement, duplicate the dashboard and edit the following search string in **Source** mode: `itsi_svc_entitled=svc_total_entitled*0.8` Replace `svc_total_entitled*0.8` with the total SVC entitlement displayed in the Workload dashboard on the Splunk Cloud Platform Monitoring Console. To learn more about the Workload dashboard, see the Workload dashboard.
ITSI SVC Usage In Last 7 Days	The ITSI SVC is calculated as a portion of the total SVC entitlements. The percentage is calculated by dividing the number of scheduled ITSI searches by the total number of searches. The diagram below shows a visual representation of the breakdown calculation: .

Panel	Description
ITSI SVC Usage Breakdown by Service Tags In Last 24 Hours	Displays a breakdown of SVC usage in the last 24 hours based on service tags. The breakdown is calculated by dividing the number of searches for a KPI by relevant service tags. Each KPI base search is linked to one or more services. For KPIs that share a service, the searches are divided by all tags on associated services. The **GLOBAL** service tag represents KPI searches that are not tagged. For more information about service tags, see Add tags to a service in ITSI. The diagram below shows a visual representation of the breakdown calculation:  $SEARCHES_{itsi}$ Searches KPI 1 KPI 2 KPI 3 Service Tag 1 Service Tag 2 Service Tag 3 Service Tag 1 Service Tag 4 Service Tag 5 $SVC_{tag} = SVC_{itsi} \times \frac{SEARCHES_{tag}}{SEARCHES_{itsi}}$
ITSI Search Breakdown By Service Tags In Last 7 Days	Displays a breakdown of the number of ITSI searches in the last 7 days based on service tags. The breakdown is calculated by dividing the number of searches for a KPI by relevant service tags. Each KPI base search is linked to one or more services. For KPIs that share a service, the searches are divided by all tags on associated services.
ITSI SVC Breakdown By Service Tags In Last 7 Days	Displays a breakdown of SVC counts in the last 7 days based on service tags. The breakdown is calculated by dividing the number of searches for a KPI by relevant service tags.

Use the ITSI Upgrade Readiness Dashboard

The **ITSI Upgrade Readiness Dashboard** displays the results of a nightly upgrade readiness check that detects common ITSI issues in your environment before you upgrade to the next ITSI version. The dashboard surfaces and provides steps to resolve these issues so you can successfully upgrade your environment. For more information about upgrading ITSI, see Install Splunk IT Service Intelligence on a single instance.

ITSI upgrades might fail due to issues outside of those checked by the Upgrade Readiness dashboard. For more information, see Troubleshooting ITSI upgrades and Rolling back an upgrade of ITSI.

You must have the itoa_admin role to access this dashboard. The dashboard populates after the upgrade readiness job runs every 24 hours.

Dashboard panels

Panel	Description
No issues	Displays the status of each specific job that checks for issues that may interfere with the ITSI upgrade. If no issues are detected, the dashboard displays the green checkmark icon ✓ . For example, if the job doesn't detect any KPIs with missing threshold levels, the check for **Missing KPI Threshold Levels** displays this green checkmark icon.
Upgrade precheck name	Select each check name to view a more detailed description and remediation steps. Drill down to view the specific objects affected by each check by selecting the **Open in Search** button on the dashboard. Additionally, the dashboard lists extra details about each check: • **Severity**: describes the level of impact that the error might have on the upgrade. There are 3 severity levels: **Minor**, **Moderate**, and **Major**. A minor or moderate severity error will not stop an upgrade, but might cause issues with functionality. However, though you'll still be able to proceed with upgrading to the next version, an error categorized with major severity will likely cause a failure during the upgrade process. • **Blocks Upgrade**: if this is **True**, it is highly recommended you address errors before upgrading your ITSI environment to prevent potential failures or errors. However, you can still proceed with upgrading to the next version of ITSI even if errors are identified. • **Category**: displays the type of objects affected by the error, such as Services or Service Templates. • **Description**: A summary of the issue. • **Resolution**: Steps you can take to resolve the issue. • **Documentation**: A link to related documentation for the affected objects. For more information about each check, see the Upgrade error reference.
Number of prechecks with issues detected	Displays the number of upgrade checks that did not pass and require attention before you upgrade your environment.

Manually start a check

Select the **Start new precheck** button on the dashboard page to manually start a new upgrade readiness check. Refresh the page after the precheck runs to view the most up to date data on the dashboard. For example, if you remediate an error and select this button to run a precheck again, the error should no longer surface on the dashboard.

Run automatic fixes before upgrading ITSI

Prechecks that do not pass the upgrade readiness check can be automatically resolved by selecting the **Run fix** button. This fix will address the issues blocking an upgrade. For example, when you run a fix for the **Linked services missing** precheck, a job will run to link the missing service dependencies to the correct service templates.

> Do not run a backup or restore operation at the same time an automatic fix is running in the background in order to avoid conflicts with the data being fixed and the backup and restore operation.

Upgrade error reference

The table below lists the types of upgrade errors that may be identified and displayed on the dashboard.

Error name	Description	Resolution	Severity	Run fix available? (Yes/No)
Missing KPI threshold levels	One or more of your KPIs are missing threshold levels.	Add the missing KPI threshold field to the KPI. View a list of KPIs to edit by selecting the **Open in Search** button.	Moderate	No
Incorrect KPI base search reference	Services or service templates display an inaccurate dependency with one or more shared KPI base searches. For example, one of your services shows that it is dependent on a shared KPI base search that doesn't exist in your system.	For the KPIs of services, update the KPI search type from base search to adhoc search. View a list of KPIs to edit by selecting the **Open in Search** button. For the KPIs of service templates, update the current base search to a different base search. View a list of KPIs to edit by selecting the **Open in Search** button.	Moderate	No
KPIs with missing KPI threshold template	Services or service templates display an inaccurate dependency with one or more KPI threshold templates.	Update the KPI threshold configuration. View a list of KPIs to edit by selecting the **Open in Search** button.	Moderate	Yes
Objects having KPIs with incorrect search type	KPIs aren't configured with the correct search type.	Update the KPI to the correct search type. View a list of KPIs to edit by selecting the **Open in Search** button.	Moderate	No
Missing KPI base search	KPIs linked to services or service templates are missing a base search ID field.	For the KPIs of services, update the KPI search type from base search to adhoc search. View a list of KPIs to edit by selecting the **Open in Search** button. For the KPIs of service templates, update the current base search to a different base search. View a list of KPIs to edit by selecting the **Open in Search** button.	Moderate	Yes **Note:** Note: An automatic fix is only available for KPIs of services. You must manually fix the missing base searches for KPIs of service templates.
Incorrect linked services	Service templates are linked to services that don't exist.	Add a tag to the service to remove the incorrect linked services.	Moderate	Yes
Linked services missing	Service templates don't display the correct linked services.	From the service configuration, link the service to the correct service template.	Minor	Yes
Incorrect linked services number	The number of linked services does not match the total linked services field for a service template.	Add a tag to the service template to resolve the discrepancy.	Minor	Yes
Sync status field error	The sync status field is not set to 'synced'.	Add a tag to the service template in order for the 'synced' status to display or resolve the issue using the last_sync_error message.	Major	No
			Minor	Yes

Error name	Description	Resolution	Severity	Run fix available? (Yes/No)
Incorrect service dependencies	A service is incorrectly displayed as the dependent of another service that doesn't exist.	This precheck error won't affect your upgrade.		
Main service dependencies missing	A service doesn't display the other services that it depends on.	This precheck error won't affect your upgrade.	Minor	Yes
Dependent services missing	A service doesn't display the correct dependent services.	Add the correct dependent services to the main service.	Minor	Yes
Duplicate Service names	One or more services has the same name as an existing service.	Change or update one of the duplicate service names.	Moderate	No
Incorrect dependent services	A service has inaccurate service dependencies. The service displays dependencies on other services that don't exist.	Add a tag to the service to remove the incorrect dependent services.	Moderate	Yes

Administer Splunk ITSI with configuration files

About configuration files in ITSI

Splunk IT Service Intelligence configuration information is stored in **configuration files**. These files are identified by the `.conf` extension and hold the information for different aspects of your ITSI configurations. These aspects include:

- System settings
- Authentication and authorization information
- KPI, glass table, and deep dive configurations
- Notable event configurations
- Module settings

A single Splunk instance typically has multiple versions of configuration files across several directories. You can have configuration files with the same name in your default, local, and app directories. This creates a layering effect that allows Splunk to determine configuration priorities based on factors such as the current user and the current app.

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

Most configuration files come packaged with your ITSI software in the `$SPLUNK_HOME/etc/apps/` directory.

Editing a configuration file

Never change, copy, or move the configuration files in the default directory. Default files must remain intact and in their original location. To change settings for a particular ITSI 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.

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. See Configuration file precedence.
- Consult the .spec and .example files for the configuration file. These files reside in the file system in `$SPLUNK_HOME/etc/apps/SA-ITOA/README` or `$SPLUNK_HOME/etc/apps/itsi/README`

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

List of ITSI configuration files

The following is a list of ITSI configuration files. All files are located under `$SPLUNK_HOME/etc/apps/`. Most .conf files have accompanying spec and example files located in the README folder that list all supporting attributes. Contact Support before editing a conf file that does not have an accompanying spec or example file.

If you are using Splunk Cloud, you can't edit a .conf file directly. For any task that requires editing a .conf file, submit a ticket using the Support Portal and Splunk Support will work with you to arrange a maintenance window.

Caution: Never change or copy the configuration files in the default directory. Default files must remain intact and in their original location. The upgrade process overwrites the default directory, so any changes that you make in the default directory are lost on upgrade. Create and edit your files in a local directory, for example `$SPLUNK_HOME/etc/apps/<app_name>/local`. Local directories are not overwritten during upgrades. For more information, see Configuration file directories.

File	Purpose	ITSI Location
alert_actions.conf	Generate ITSI notable events and configure episode actions.	/SA-ITOA/default
alert_actions.conf	Summarize KPI searches into the ITSI summary index.	/itsi/default
app_common_flags.conf	Enable or disable certain ITSI features. CAUTION: Do not edit this file.	/itsi/default
authorize.conf	Configure ITSI-specific roles and capabilities, including role-based access controls. Always use `/itsi/default`. For more information, see Grant and revoke user permissions in ITSI.	/itsi/default
collections.conf	Configure KV store collections for ITSI.	/SA-ITOA/default
commands.conf	Connect search commands to any custom search script.	/SA-ITOA/default
datamodels.conf	Attribute/value pairs for configuring data models.	/DA-ITSI-APPSERVER/default /DA-ITSI-LB/default /DA-ITSI-VIRTUALIZATION/default
deep_dive_drilldowns.conf	Configure deep dive drilldowns, add new drilldowns.	/itsi/default
itsi_entity_type.conf		/SA-ITOA/default

File	Purpose	ITSI Location
	Upload sample entity types to the KV store. For more information, see Create entity types in ITSI.	
distsearch.conf	Specify behavior for distributed search. Group search peers to facilitate searching on a subset of peers.	/SA-ITOA/default
drilldownsearch_offset.conf	Configure time range picker presets for correlation search drilldown offsets.	/itsi/default
fields.conf	Create multi-value fields and add search capability for indexed fields.	/itsi/default
glasstable_icon_library.conf	Add and remove icons from the glass table icon library.	/itsi/default
inputs.conf	Set up data inputs.	/SA-ITOA/default /itsi/default
itsi_da.conf	(Deprecated) Configure an app to export entity searches and service templates for use within ITSI.	/SA-ITOA/default
itsi_data_integrations.conf	See the available chicklets listed on the Data Integrations page. For more information, see Overview of entity integrations in ITSI.	/itsi/default
itsi_deep_dive.conf	Upload deep dives to the KV store.	/SA-ITOA/default
itsi_event_management.conf	Configure Episode Review default settings.	/SA-ITOA/default

File	Purpose	ITSI Location
itsi_glass_table.conf	Upload glass tables to the KV store.	/SA-ITOA/default
itsi_kpi_base_search.conf	Upload KPI base searches to the KV store.	/SA-ITOA/default
itsi_kpi_template.conf	Upload KPI templates to the KV store.	/SA-ITOA/default
itsi_kpi_threshold_template.conf	Upload KPI threshold templates to the KV store.	/SA-ITOA/default
itsi_module_settings.conf	Define whether a module is editable in the module lister page. Default is false.	/DA-ITSI-EUEM/default /DA-ITSI-WEBSERVER/default /DA-ITSI-OS/default /DA-ITSI-VIRTUALIZATION/default /DA-ITSI-APPSERVER/default /DA-ITSI-LB/default /DA-ITSI-APM/default /DA-ITSI-DATABASE/default /DA-ITSI-STORAGE/default /DA-ITSI-CLOUD/default
itsi_module_viz.conf	Change tab names and panel titles in a module details dashboard.	/DA-ITSI-EUEM/default /DA-ITSI-WEBSERVER/default /DA-ITSI-OS/default /DA-ITSI-VIRTUALIZATION/default /DA-ITSI-APPSERVER/default /DA-ITSI-LB/default /DA-ITSI-APM/default /DA-ITSI-DATABASE/default /DA-ITSI-STORAGE/default /DA-ITSI-CLOUD/default
itsi_notable_event_retention.conf	Define how long notable events are retained before they move to the index. Default is 6 months.	/SA-ITOA/default
itsi_notable_event_severity.conf	Configure the colors associated with different severity levels in Episode Review.	/SA-ITOA/default
itsi_notable_event_status.conf	Configure label descriptions and	/SA-ITOA/default

File	Purpose	ITSI Location
	event status in Episode Review.	
itsi_service.conf	Upload services to the KV store.	/SA-ITOA/default
itsi_service_analyzer.conf	Configure auto-refresh interval, or disable auto-refresh.	/SA-ITOA/default
itsi_service_template.conf	Configure an app to export service templates for use within ITSI.	/SA-ITOA/default
itsi_settings.conf	Configure ITSI.	/SA-ITOA/default
itsi_team.conf	Upload sample ITSI teams to the KV store.	/SA-ITOA/default
limits.conf	Set various limits (such as maximum result size or concurrent real-time searches) for search commands.	/SA-ITOA/default /itsi/default
macros.conf	Define search macros in Settings.	/SA-ITOA/default /itsi/default
mad.conf	Configure anomaly detection.	/SA-ITSI-MetricAD/default
notable_event_actions.conf	Configure actions to take on groups in Episode Review.	/SA-ITOA/default
notable_event_commonality.conf	Define fields to include or exclude from the Common Fields tab of Episode Review.	/SA-ITOA/default
notable_event_correlation.conf	Set threshold values and limits for Smart Mode event correlation.	/SA-ITOA/default
props.conf	Set indexing property configurations, including timezone offset,	/SA-ITOA/default

File	Purpose	ITSI Location
	custom source type rules, and pattern collision priorities. Also, map transforms to event properties.	
restmap.conf	Create custom REST endpoints.	/SA-ITOA/default
savedsearches.conf	Define ordinary reports, scheduled reports, and alerts.	/SA-ITOA/default
searchbnf.conf	Configure the search assistant.	/SA-ITOA/default
threshold_labels.conf	Change the label, color, threshold level, health weight, minimum and maximum health score, and score contribution. Changes to this file won't be reflected on the service analyzer.	/itsi/default
threshold_periods.conf	Deprecated. Do not edit.	/itsi/default
transforms.conf	Configure regex transformations to perform on data inputs. Use in tandem with props.conf.	/SA-ITOA/default
visualizations.conf	Declare common visualizations that other modules can use.	/SA-ITSI-CustomModuleViz/default
web.conf	Configure Splunk Web, enable HTTPS.	/SA-ITOA/default

Configuration file reference

alert_actions.conf

The following are the spec and example files for `alert_actions.conf`.

alert_actions.conf.spec

```
# This file contains possible attributes and values for generating ITSI
# notable events, configuring episode actions, and executing
# post-search processing actions.
#
# There is an alert_actions.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an alert_actions.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.

ttl = <integer> [p]
* The minimum time to live (TTL), in seconds, of the search artifacts
  if this action is triggered.
* If p follows the integer, then the integer is the number of scheduled periods.
* Default: 600 (10 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.
* If you do not include a suffix, the time defaults to seconds.
* Default: 600 (10 minutes)

maxresults = <integer>
* The maximum number of search results sent via the alert.
* Default: 10000

is_custom = <boolean>
```

```
* Specifies whether the alert action is based on the custom alert
  actions framework and is supposed to be listed in the search UI.
* Default: 1

label = <string>
* Defines the label shown in the UI.
* If not specified, the stanza name is used instead.

description = <string>
* Defines the description shown in the UI.

payload_format = [xml|json]
* The format in which the alert script receives
  the configuration via STDIN.
* Default: json
```

[itsi_event_generator]

```
* Generate notable events under this stanza name.
* ITSI sends notable events to the ITSI summary index.
* Follow this stanza name with any number of the following
  attribute/value pairs.
* If you do not specify an entry for each attribute, Splunk will
  use the default value.

param.http_token_name = <string>
* The HTTP token name.
* Optional.
* If you do not provide a token name, ITSI obtains one
  token using the index and sourcetype parameters below.

param.index = <string>
* The index name.
* This setting is required if you do not provide an HTTP
  token for the 'param.http_token_name' setting.
* Default: itsi_tracked_alerts

param.sourcetype = <string>
* The sourcetype.
* This setting is used if you do not provide an HTTP
  token for the 'param.http_token_name' setting.
* Default: itsi_notable:event

param.event_identifier_fields = <comma-separated list>
* A list of fields that are used to identify event duplication.
* Default: source

param.is_use_event_time = <boolean>
* If "1", ITSI uses the actual event time.
* If "0", ITSI uses the time the event was indexed.
* Default: 0

param.event_field_max_length = <integer>
* The maximum field length.
* Default: 10000

param.title = <string>
* The title of the notable event in Episode Review.
* Optional. If a title is not provided the search name
  becomes the title.
```

```
param.description = <string>
* A description of the notable event.
* Optional. If a description is not provided the search
  description becomes the event description.

param.owner = <string>
* The initial owner of the notable event.
* Optional.
* Default: unassigned

param.status = <string>
* The triage status of the event in Episode Review.
* Optional. If a status is not provided then default_status is assigned.
* Values must match an integer specified in the default version of
  itsi_notable_event_status.conf, or the local version if you created one.

param.severity = <string>
* The level of importance of the event.
* Optional. If a severity is not provided then default_severity is assigned.
* Values must match an integer specified in the default version of
  itsi_notable_event_severity.conf, or the local version if you created one.

param.itsi_instruction = <string>
* Instructions for how to address the notable event.
* Optional.
* Must use tokens such as %fieldname% to map the field name from an external event.
  Static instructions are not supported.
* You can use an aggregation policy to aggregate individual instructions into an episode.
  By default, episodes display the instructions for the first event in an episode.

param.drilldown_search_title = <string>
* You can drill down to a specific Splunk search from an event or episode. This setting
  specifies the text of the drilldown link. Provide the actual search string in the
  'param.drilldown_search_search' setting.
* Optional.

param.drilldown_search_search = <string>
* The Splunk search string to drill down to from an event or episode.
* Optional.

param.drilldown_search_latest_offset = <seconds>
* Defines how far ahead from the time of the event, in seconds,
  to look for related events.
* This offset is added to the event time.
* Optional.

param.drilldown_search_earliest_offset = <string>
* Defines how far back from the time of the event, in seconds,
  to start looking for related events.
* This offset is subtracted from the event time.
* Optional.

param.drilldown_title = <string>
* You can drill down to a specific URI from an event or episode. This setting
  specifies the text of the drilldown link. Provide the actual URI string in
  the 'param.drilldown_uri' setting.
* Optional.

param.drilldown_uri = <string>
* The URI to drill down to from an event or episode.
* Optional.
```

```
param.service_ids = <comma-separated list>
* A list of service IDs representing one or more ITSI services to
  which this correlation search applies.
* Optional.

param.entity_lookup_field = <string>
* The field in the data retrieved by the correlation search that
  is used to look up corresponding entities. For example, "host".
* Optional.

param.search_type = <string>
* The search type.
* Optional.
* Default: custom

param.meta_data = <string>
* The search type of any stored metadata.
* Optional.

param.is_ad_at =  <boolean>
* Whether this correlation is created by enabling adaptive
  thresholding or anomaly detection (AT/AD) for KPIs or services.
* Optional.
* If "1", the correlation is created by adaptive thresholding or anomaly detection.
* If "0", the correlation is not created by adaptive thresholding or anomaly detection.

param.ad_at_kpi_ids = <comma-separated list>
* A list of KPIs where adaptive thresholding or anomaly detection is enabled.
* Optional.

param.editor = <string>
* The type of editor used to create the correlation search.
* Can be either "advance_correlation_builder_editor", which is the correlation
  search editor in ITSI, or "multi_kpi_alert_editor", which is the multi-KPI
  alert builder.
* Default: advance_correlation_builder_editor
```

[itsi_sample_event_action_ping]

```
* Ping a host in one or more ITSI episodes under this stanza name.
* Follow this stanza name with any number of the following
  attribute/value pairs.
* If you don't specify an entry for each attribute, Splunk uses
  the default value.

param.host_to_ping = <string>
* The field from the episode representing the host to ping.
* If your event contains the field 'server', set to '%server%'.
* When ITSI executes the alert action, it extracts the value corresponding
  to the token value from event data and tries to ping it.
* If you set a value that does not begin and end with '%', ITSI
  considers this to be the value to ping. No extractions are done in this case.
* Default: %orig_host%
```

[itsi_event_action_link_ticket]

```
* Set options to associate an episode with a ticket from an
  external ticketing system under this stanza name.
* Follow this stanza name with any number of the following
```

```
  attribute/value pairs.
* If you do not specify an entry for each attribute, Splunk will
  use the default value.

param.ticket_system = <string>
* The name of the external ticketing system.
* This setting is required to create/update/delete a ticket.
* There is no default.

param.ticket_id = <string>
* The ID of the specific ticket to link to.
* This setting is required to create/update/delete a ticket.
* There is no default.

param.ticket_url = <string>
* The drilldown link to the ticket in the external ticketing system.
* This setting is required to create/update a ticket.
* There is no default.

param.operation = <upsert|delete>
* Specifies the type of action to take on the ticket.
* If "upsert", ITSI inserts or updates existing fields.
* If "delete", ITSI deletes the ticket.
* There is no default.

param.kwargs = <dict>
* A dictionary of additional fields to pass to the ticket.
* Optional.
* There is no default.
```

[itsi_event_action_link_url]

```
* Set options to associate an episode with an external URL.
* Follow this stanza name with any number of the following
  attribute/value pairs.
* If you do not specify an entry for each attribute, Splunk will
  use the default value.

param.url = <string>
* A URL to an external document or incident

param.url_description = <string>
* The label or description of the document to link to.
* This setting is required to create/update/delete a URL.
* There is no default.

param.operation = <upsert|delete>
* Specifies the type of action to take on the URL.
* If "upsert", ITSI inserts or updates existing fields.
* If "delete", ITSI deletes the URL.
* There is no default.

param.kwargs = <dict>
* A dictionary of additional fields to pass to the URL.
* Optional.
* There is no default.
```

[itsi_event_action_snow_wrapper]

```
param.account = <list>
* The name of the account in which the incident is created.
* Required.

param.state = <string>
* The state of the incident.
* Optional.

param.configuration_item = <string>
* Configuration item.
* Optional.

param.contact_type = <string>
* The method by which the incident was reported.
* Optional.

param.assignment_group = <string>
* The name of the assignment group associated with the incident.
* Optional.

param.category = <string>
* The category of the incident.
* Required.

param.subcategory = <string>
* The subcategory of the incident.
* Optional.

param.impact = <number>
* The impact value of the incident.
* Optional.

param.urgency = <number>
* The urgency of the incident.
* Optional.

param.priority = <number>
* The priority of the incident, determined by the impact and urgency values.
* Optional.

param.short_description = <string>
* A brief description of the ITSI episode.
* Required.

param.correlation_id = <string>
* A brief description of the ServiceNow incident.
* Optional.

param.splunk_url = <link>
* An external drilldown link from the ServiceNow incident.
* You can use this setting to link back to the corresponding episode in ITSI.
* Optional.

param.custom_fields = <string>
* Custom fields.
* Optional.
```

[itsi_event_action_clear_sim_incidents]

```
* Clear all Splunk Infrastructure Monitoring incidents within an ITSI episode. An incident
  in Splunk Infrastructure Monitoring is the combination of an alert event and a clear event.
```

[itsi_import_objects]

```
* Import entity and service object data.

param.backfill_enabled = <boolean>
* Whether to enable backfill on all KPIs in linked service templates.
* Optional.
* Default: 0

param.entity_description_fields = <string>
* A list of fields that represents the description of an entity.
* Optional.

param.entity_field_mapping = <string>
* A key-value mapping of fields to re-map to other fields in the data.
* Follows a <field> = <Splunk search field> format.
* For example, ip1 = dest, ip2 = dest, storage_type = volume
* Use this setting to rename a field or column to an alias or info value.
* Optional.

param.entity_identifier_fields = <string>
* A list of fields that represent identifier data of an entity.
* Optional.

param.entity_informational_fields = <string>
* A list of fields that represent the informational data of an entity.
* Optional.

param.entity_merge_field = <string>
* The field that should be used when resolving conflicts between entities.
* Optional.

param.entity_title_field = <string>
* The field that represents the title of an entity.
* Optional.

param.entity_type_field = <string>
* The field that matches the title for the entity type that is associated with an entity.
* Optional.

param.service_dependents_fields = <string>
* A list of fields that indicate service dependencies.
* Optional.

param.service_description_fields = <string>
* A list of fields that represents the description of a service.
* Optional.

param.service_tags_field = <string>
* A list of fields that represents one or more tags to be added to a service.
* Optional.

param.service_enabled = <boolean>
* Whether or not imported services should be enabled.
* Optional.
* Default: 0

param.service_team = <string>
* The ITSI team that the imported services belong to.
* Optional.
* Default: default_itsi_security_group

param.service_templates_config = <string>
* A dictionary of key-value pairs that maps entity rules to service templates.
* Optional.

param.service_template_field = <string>
* Determines which service template a service is linked to.
* Optional.

param.service_title_field = <string>
* The field that represents the title of a service.
* Optional.

param.update_type = <string>
* The update/insertion method when uploading entities.
* APPEND: ITSI makes no attempt to identify commonalities between entities.
*   All information is appended to the table.
* UPSERT: ITSI appends new entries.  Existing entries (based on the value
*   found in the title_field) have additional information appended
*   to the existing record.
* REPLACE: ITSI appends new entries. Existing entries (based on the value
*   found in the title_field) are replaced by the new record value.
* Optional.
* Default: UPSERT
```

[itsi_summary_metrics_collect]

```
* Wraps the mcollect macro for converting event data into metrics and pushing it into the ITSI metrics
summary index.
```

alert_actions.conf.example

```
No example
```

app_common_flags.conf

The following are the spec and example files for `app_common_flags.conf`.

app_common_flags.conf.spec

```
# This file contains attributes and values for disabling (feature flagging)
# certain ITSI features.
#
# There is an app_common_flags.conf in $SPLUNK_HOME/etc/apps/itsi/default.
# To set custom configurations, place an app_common_flags.conf in
# $SPLUNK_HOME/etc/apps/itsi/local/. 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# CAUTION: This is an internal configuration file used to turn off certain ITSI
# features that are incomplete. Do NOT edit or remove 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, 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.
```

[<app_common_flag>]

```
* Each stanza represents a feature within Splunk IT Service Intelligence (ITSI).
* If the feature is disabled, it is currently incomplete and should NOT be enabled.

feature = <string>
* The name of the feature.

description = <string>
* A description of what the feature does.

disabled = <boolean>
* Whether the feature is enabled or disabled.
* If "1", the feature is disabled.
* If "0", the feature is enabled.
```

app_common_flags.conf.example

```
No example
```

authorize.conf

The following are the spec and example files for `authorize.conf`.

authorize.conf.spec

```
#   Version 8.1.0
#
```

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

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

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.
* 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.
* If you make any changes in the "Indexes searched by default" Settings panel
  for a role in Splunk Web, those values take precedence, and any wildcards
  you specify in this setting are lost.
* 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.
* 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.
* 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 'deleteIndexesAlowed'. 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.
* 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 historical 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

federatedProviders = <semicolon-separated list>
* List of federated providers that the role can access.
* Allows a user to run federated searches defined in the savedsearches.conf file. This
* setting must be used in conjunction with fsh_search capability.
* Defaults to none.

####
# 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.

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: never

disabled = <boolean>
* Disables and enables Splunk token authorization.
* Default: true
```

[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::run_multi_phased_searches]

```
* Lets a user in a distributed search environment run searches with
  three or more map-reduce phases.
* Lets users take advantage of the search performance gains
  related to parallel reduce functionality.
* Multi-phased searches can lead to higher resource utilization on
  indexers, but they can also reduce resource utilization on search heads.
```

[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_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_own_password]

```
* Lets a user change their own password. You can remove this capability
  to control the password for a user.
```

[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::delete_messages]

```
* Lets a user delete system messages that appear in the UI 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_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_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.
```

[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_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_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 limit this ability, 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]

```
* Lets a user access and change the license.
```

[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_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 access the /storage/passwords endpoint.
* Lets the user perform GET operations.
* The 'admin_all_objects' capability must be added to the role in order 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_mcollect]

```
* Lets a user run the 'mcollect' and 'meventcollect' commands.
```

[capability::run_msearch]

```
* Lets a user run the 'mpreview' and 'msearch' commands.
```

[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::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::web_debug]

```
* Lets a user access /_bump and /debug/** web debug endpoints.
```

[capability::fsh_manage]

```
* Lets a user in Splunk platform implementations that have enabled Data
  Fabric Search (DFS) functionality manage the federated search settings.
* With the federated search settings, users with this role can add federated
  providers to federated.conf and manage user access to those federated
  providers through the maintenance of authentication settings.
* The 'admin' role has this capability enabled by default.
```

[capability::fsh_search]

```
* Lets a user in Splunk platform implementations that have enabled Data Fabric
  Search (DFS) functionality run federated searches.
* Lets a user create federated searches in the savedsearches.conf.
* The 'admin' role has this capability enabled by default.
```

[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.
```

authorize.conf.example

```
#   Version 8.1.0
#
# 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

# 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.
```

commands.conf

The following are the spec and example files for `commands.conf`.

commands.conf.spec

```
#   Version 8.1.0
```

OVERVIEW

```
# This file contains descriptions for the setting/value pairs that you can
# use for creating search commands for custom search scripts.
#
# You can add your custom search script to one of these paths:
# * If you add your custom search script to the $SPLUNK_HOME/etc/searchscripts/
#   path, put a custom commands.conf file in the $SPLUNK_HOME/etc/system/local/
#   directory.
# * 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]
#   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>
* Whether or not to run this command if generating results just for preview
  rather than for final output.
* Default: true

enableheader = <boolean>
* Whether or not your script expects header information.
* Currently, the only thing in the header information is an authentication token.
* 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

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 8.1.0
#
# 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
```

collections.conf

The following are the spec and example files for `collections.conf`.

collections.conf.spec

```
#   Version 8.1.0
#
# 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.
* 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 8.1.0
```

```
#
# 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}
```

datamodels.conf

The following are the spec and example files for `datamodels.conf`.

datamodels.conf.spec

```
#   Version 8.1.0
#
# 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 8.1.0
#
# 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
```

deep_dive_drilldowns.conf

The following are the spec and example files for `deep_dive_drilldowns.conf`.

deep_dive_drilldowns.conf.spec

```
# Copyright (C) 2005-2020 Splunk Inc. All Rights Reserved.
# This file contains all possible attribute/value pairs for configuring
# drilldown options for deep dive lanes.
#
# A unique drilldown options is represented by a stanza in this file.
# The name of the stanza is the name that will appear in the UI.
# ITSI currently supports a maximum of 22 drilldown stanzas in this file.
# Default values are provided for most settings and are defined in
# the [default] stanza of the configuration file.
#
# Other more complex drilldown options are not defined in this file
# because they are only represented in the deep dive code and cannot
# be disabled.
#
# There is a deep_dive_drilldowns.conf in $SPLUNK_HOME/etc/apps/itsi/default.
# To set custom configurations, place a deep_dive_drilldowns.conf in
# $SPLUNK_HOME/etc/apps/itsi/local/. You must restart Splunk software to
# enable configurations.
#
# To learn more about configuration files (including precedence) please
# see the documentation located at
# https://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[<name>]

```
* Each stanza represents a unique drilldown option. Use these settings to
  configure properties for all types of drilldowns.

type = uri|search
* Represents whether this drilldown is meant to redirect to a new
  URI or open a Splunk search.
* Required.

replace_tokens = true|false
* Enables token replacement in the search string or URI.
* Optional.
* If "true", the search or URI is token replaced by properties of the drilldown.
* Token replacement is similar to token replacement in simpleXML. Tokens are
  represented in tokenized strings as a sub-string key surrounded by '$'.
    * For example, search=index=_internal | stats count | where count>$value$
* The following tokens are available for replacement by default:
  * lane_title - the title of the lane
  * lane_subtitle - the subtitle of the lane
  * lane_search - the search that powered the primary graph in the lane
  * earliest - the earliest epoch time stamp of the entire lane
  * latest - the latest epoch time stamp of the entire lane
  * bucket_earliest - the earliest epoch time stamp of the time bucket clicked
  * bucket_latest - the latest epoch time stamp of the time bucket clicked
* The following tokens are available for KPI lanes only:
  * kpi.service_id - the ID of the service to which the KPI belongs
  * kpi.service_title - the tite of the service to which the KPI belongs
  * kpi.kpi_id - the ID of the KPI represented in the lane
  * kpi.kpi_title - the title of the KPI represented in the lane
  * kpi.single_value_search - the raw data alert search for the KPI
  * kpi.timeseries_search - the raw data time series search for the KPI
  * kpi.base_search - the event gathering/filtering search for the KPI
* Default: false

metric_lane_enabled = true|false
* Whether to enable drilldowns on metric lanes.
* Optional.
* If "true", drilldown is available on metric lanes.
* If "false", drilldown is unavailable on metric lanes.
* Default: false

kpi_lane_enabled = true|false
* Whether to enable drilldowns on KPI lanes.
* Optional.
* If "true", drilldown is available on KPI lanes.
* If "false", drilldown is unavailable on KPI lanes.
* Default: false

event_lane_enabled = true|false
* Whether to enable drilldowns on event lanes.
* Optional.
* If "true", drilldown is available on event lanes.
* If "false", drilldown is unavailable on event lanes.
* Default: false

####
# Entity-based features
####
# Entity-based features are only available on KPI lanes because KPI lanes are the only
# lanes that understand entities. Note that KPIs must have 'Split by Entity' enabled.

entity_level_only = true|false
* Whether to enable drilldowns only on lanes that surface entity-level information.
* Optional.
* If "true", drilldown is only available on lanes that surface entity-level information.
* If "false", drilldown is available on all lanes.
* Entity-level drilldowns make additional tokens and information available based
  on the entities clicked. See the 'entity_tokens' setting for more details.
* Default: false

entity_tokens = <csv>
* A CSV file of entity attributes to include on a drilldown.
* Optional.
* Only defiend entities will be available on entity-level
  drilldowns. Pseudo-entities are ignored.
* If the 'replace_tokens' setting is "true", this setting will generate
  additional token replacements.
* Attributes can be either info fields or aliases.
* If the 'uri_payload_type' setting is set to "json", these entity attributes
  are added to the JSON payload per entity.
* Tokens from the first entity are replaced. If there are multiple entities,
  they all appear in a JSON payload.
* Tokens have the format "entity.<attribute name>".
* If any entity tokens are set to "all" (required to make drilldown work),
  entity.id and entity.title will always be available as tokens.

entity_activation_rules = <JSON blob of entity rules>|all|kpi_title_match
* Determines which entities to consider for drilldown.
* Optional.
* If "all", all entities are considered valid for drilldown.
* If "kpi_title_match", no entity rule-based matching is performed. Instead,
  for the KPIs listed in the 'kpi_titles_with_drilldown' setting,
  their associated entity lanes include a custom drilldown for that KPI.
  The drilldown redirects to the URI you provide, after token replacement.
* If set to a JSON blob of entity rules, entities are tested for
  compliance with those rules. If no entities match, the drilldown
  isn't available. If some or all all entities match, only those
  matching are passed to the drilldown.
* Default: "all"

kpi_titles_with_drilldown = <comma-separated list of KPI titles>
* Configure custom drilldowns for specific KPIs. This setting lets you drill down
  to a specified URI when viewing the entity overlays for that KPI in a deep dive.
* Optional.
* This setting is only consumed if the 'entity_activation_rules' setting
  is set to "kpi_title_match".
```

```
####
# Properties for search type drilldowns
####
search = <tokenized search string>
* The search to use in the new lane or on the search page.
* Required for search type drilldowns.
* If the 'replace_tokens' setting is "true", the search is token replaced
  by properties from the drilldown itself.

add_lane_enabled = true|false
* Whether users can activate the drilldown as a search.
* Required for search type drilldowns
* If "true", users can activate the drilldown as a search.
* If "false', users cannot activate the drilldown as a search.
* Default: false

use_bucket_timerange = true|false
* Whether to use only the time range of the selected bucket
  when redirected to a Splunk search.
* Optional.
* If "true", the drilldown search uses only the time range from which
  the user clicked in the deep dive.
* If "false, the drilldown search uses the entire search timerange.
* Default: true

new_lane_settings = <tokenized JSON for lane settings properties>
* A tokenized JSON string that represents a model to use for new lanes.
* Required for search type drilldowns with the 'add_lane_enabled' setting
  set to "true".
* The "search" setting is overridden by the search property in this stanza.
* If the 'replace_tokens' setting is "true", the string is token replaced
  by properties from the drilldown itself.
* Default lane settings are applied if you do not specify any values.

####
# Properties for URI type drilldowns
####
uri = <str>
* The URI to redirect to on the drilldown.
* Required for URI type drilldowns.
* If the 'replace_tokens' setting is "true" and the 'uri_payload_type'
  setting is "simple", the URI string is replaced by tokens.
* Follows the format of an href:
  * A leading protocol allows a change in domain.
  * A leading slash changes the full path on the same domain.
  * Any other string only replaces the last segment of the URI with that string.

uri_payload_type = simple|json
* If "simple", token replacement is performed on the URI as if it were a search.
* If "json", no token replacement is performed and a query string parameter
  'drilldown_payload' is appended to the URI with a JSON representation of
  the context of a drilldown. This payload will always contain
  the context portion of the JSON blob, which contains the basic properties.
* If it is entity level and the entity properties of the drilldown are specified,
  the entities portion will exist and consist of the entity ID and title
  as well as all attributes specified in as 'entity_tokens'. A JSON payload
  format will look like the following (assumes 'entity_tokens' was host,family):
    {
      "context": {
        "earliest": <earliest time of full lane>,
        "latest": <latest time of full lane>,
        "bucket_earliest": <earliest time of bucket clicked>,
        "bucket_latest": <latest time of the bucket clicked>,
        "return_url": <URI of the current deep dive>,
        "service_id": "158bdaf4-6b0c-433e-9c24-c3a36c0e8eea",
        "kpi_id": "65ec30c5e1dd5046ac5416f5",
        "service_title": "Production Webservers",
        "kpi_title": "Total Request Latency (ms)"
      },
      "entities": [
        {
          "id": "5303377f-162c-45cc-809a-d1e3254ea4a1",
          "title": "Host Title 1",
          "host": "Host1",
          "family": "Linux"
        },
        {
          "id": "7aefd044-0f46-4ba4-ab13-f31e5797a3bf",
          "title": "Host Title 2",
          "host": "Host2",
          "family": "Linux"
        }
      ]
    }
* Default: simple
```

deep_dive_drilldowns.conf.example

```
# This is an example deep_dive_drilldowns.conf. Use this file to
# configure custom drilldowns.
#
# To use one or more of these configurations, copy the configuration block
# into deep_dive_drilldowns.conf in $SPLUNK_HOME/etc/apps/itsi/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# This example alert includes showing raw events at selected time buckets,
# showing raw events from a displayed time range, showing KPI events for
# a host, and showing all events for a host.

[Show raw events at selected time bucket]
type = uri
kpi_lane_enabled = true
entity_level_only = false
uri = /app/itsi/search?q=search $kpi.base_search$&earliest=$bucket_earliest$&latest=$bucket
_latest$&display.page.search.mode=smart&dispatch.sample_ratio=1
replace_tokens = false

[Show raw events from displayed time range]
type = uri
kpi_lane_enabled = true
entity_level_only = false
uri = /app/itsi/search?q=search $kpi.base
_search$&earliest=$earliest$&latest=$latest$&display.page.search.mode=smart&dispatch.sample_ratio=1
replace_tokens = false

[Show kpi events for this host]
type = uri
```

```
kpi_lane_enabled = true
entity_level_only = true
replace_tokens = true
entity_tokens = host
uri = /app/itsi/search?q=search $kpi.base_search$ AND
host=$entity.host$&earliest=$earliest$&latest=$latest$&display.page.search.mode=smart&dispatch
.sample_ratio=1
entity_activation_rules = [ \
    { \
        "rule_condition": "AND",  \
        "rule_items": [ \
            { \
                "field": "host",  \
                "field_type": "alias",  \
                "rule_type": "not",  \
                "value": "" \
            } \
        ] \
    } \
]


[Show ALL events for this host]
type = uri
kpi_lane_enabled = true
entity_level_only = true
replace_tokens = true
entity_tokens = host
uri = /app/itsi/search?q=search index=*
host=$entity.host$&earliest=$earliest$&latest=$latest$&display.page.search.mode=smart&dispatch
.sample_ratio=1
entity_activation_rules = [ \
    { \
        "rule_condition": "AND",  \
        "rule_items": [ \
            { \
                "field": "host",  \
                "field_type": "alias",  \
                "rule_type": "not",  \
                "value": "" \
            } \
        ] \
    } \
]
```

distsearch.conf

The following are the spec and example files for `distsearch.conf`.

distsearch.conf.spec

```
#   Version 8.1.0
#
# 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

shareBundles = <boolean>
* DEPRECATED.

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.

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.

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.

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
```

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 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
  Splunk logs an error message.
* The maximum value is 102400 (100 GB).
* If the bundle exceed '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.
* 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

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
```

RFS (AKA S3/REMOTE FILE SYSTEM) REPLICATION-SPECIFIC SETTINGS

```
enableRFSReplication = <boolean>
* DEPRECATED.

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 'replicationWhitelist' 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>
* 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>
* 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>
* 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 [bundleEnforcerBlacklist] stanza.
  Hence, if both [bundleEnforcerWhitelist] and [bundleEnforcerBlacklist] 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
  [bundleEnforcerBlacklist] stanza.
* All rules are regular expressions.
* No default.
```

BUNDLE ENFORCER DENY LIST OPTIONS

```
[bundleEnforcerBlacklist]

<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 [bundleEnforceWhitelist] 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 [bundleEnforcerWhitelist].
* If this stanza is empty, then only [bundleEnforcerWhitelist] 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>
* Whether or not this group is the default group of peers against which all
  searches are run, unless a server group is not explicitly specified.
```

distsearch.conf.example

```
#   Version 8.1.0
#
# 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
[replicationWhitelist]
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
```

drilldownsearch_offset.conf

The following are the spec and example files for `drilldownsearch_offset.conf`.

drilldownsearch_offset.conf.spec

```
# This file contains attributes and values for configuring time range picker
# presets for correlation search drilldown offsets.
#
# There is a drilldownsearch_offset.conf in $SPLUNK_HOME/etc/apps/itsi/default/.
# To set custom configurations, place a drilldownsearch_offset.conf in
# $SPLUNK_HOME/etc/apps/itsi/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[<offset-period-number>]

```
timeInSecs = <integer>
* The offset time, in seconds.
* Required.

description = <string>
* The description that is shown in the UI for the earliest and latest offset
  dropdown. The earliest offset prepends "Last" to the description and the
  latest offset prepends "Next" to the description.
* Required if the 'earliest_description' and 'latest_description' settings
  are not defined below.

earliest_description = <string>
* A description for the earliest offset dropdown.
* Optional.

latest_description = <string>
* A description for the latest offset dropdown.
* Optional.
```

drilldownsearch_offset.conf.example

```
No example
```

fields.conf

The following are the spec and example files for `fields.conf`.

fields.conf.spec

```
#   Version 8.1.0
#
```

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 indexed.
* Set to "true" if the field is indexed.
* 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 8.1.0
#
# 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)
```

glasstable_icon_library.conf

The following are the spec and example files for `glasstable_icon_library.conf`.

glasstable_icon_library.conf.spec

```
# This file contains possible attributes and values for adding
# and removing icons from the glass table icon library.
#
# There is a glasstable_icon_library.conf in $SPLUNK_HOME/etc/apps/itsi/default/.
# To set custom configurations, place a glasstable_icon_library.conf in
# $SPLUNK_HOME/etc/apps/itsi/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[default]

```
iconThumbnailSrc = <string>
* The file path of the icon.
* Required.
```

[<stanza name>]

```
iconId = <string>
* An internal unique identifier for the icon.
* Required.

iconLabel = <string>
* The name or label for the icon that appears in the UI.

iconThumbnailSrc = <string>
* The file path of the icon.
* Required.

iconCategory = ['Application'|'Splunk'|'Network'|'General']
* The assigned category for the icon.
* Required.

svgPath = <string>
* The SVG path for the icon.
* The same path used for the icon library thumbnail.

defaultWidth = <positive integer>
* The initial width of the icon.

defaultHeight = <positive integer>
* The initial height of the icon.
```

glasstable_icon_library.conf.example

```
No example
```

inputs.conf

The following are the spec and example files for `inputs.conf`.

inputs.conf.spec

```
# This file contains possible settings you can use to configure ITSI inputs, register
# user access roles, and import services and entities from CSV files or search strings.
#
# There is an inputs.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default. To set custom
# configurations, place an inputs.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
# You must restart ITSI to enable new configurations.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.

# log_level = <DEBUG|INFO|WARN|ERROR>
# * This setting sets the logging level of each modular input.
# * Logging levels are in order of most to least verbose.
# * The logging level describes the type and/or quantity of output
#   that an application writes to a log file.
# * Set the logging verbosity of each modular input to specify how
#   much and what kind of information it writes to the log file.
# * Setting a log level gets you messages at that level and higher,
#   so default settings are typically INFO or WARN.

[itsi_user_access_init]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_user_access_init://<name>]
* A modular input that runs once during startup (or at the user's request)
  to register user access roles and capabilities with the SA-UserAccess module.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: WARN

app_name = <name>
* The Splunk application that has the user access roles and capabilities.
* Default: itsi
```

```
registered_capabilities = [true|false]
* Indicates whether or not capabilities have already been registered with ITSI.
* If true, the 'itsi_user_access_init' input does not re-register capabilities.
* If false, 'itsi_user_access_init' registers ITSI capabilities again.
* Default: false

[configure_itsi]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[configure_itsi://<name>]
* A configuration input that runs once (or at the user's request) to pull
  entities from the configuration file system into the App Key Value (KV) Store.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: WARN

is_configured = ""
* Left it for backwards compatibility.

[itsi_csv_import]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_csv_import://<string>]
* A modular input that periodically uploads CSV data into the KV Store.
* The CSV file must contain headers for the import to work properly.
* This input runs every 4 hours or after a Splunk software restart.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: WARN

import_from_search = <boolean>
* Indicates whether to import data from a CSV file or a Splunk search.
* If "true", this input imports data from the search specified by 'search_string'.
* If "false", this input imports CSV data from the path specified by  'csv_location'.
* This setting is required, and the input does not run if the setting is
  not present.
* There is no default.

csv_location = <path>
* The location on disk of the CSV file to import.
* NOTE: The disk must be local to the search head. Cloud storage is unacceptable.
* This setting is required if you import data from a CSV file
  (if you set 'import_from_search' to "false").
* There is no default.

search_string = <string>
* The Splunk search string that generates the data to import.
* This setting is required if you import from a search string
  (if you set 'import_from_search' to "true").
* There is no default.

service_security_group = <string>
* The ITSI team that the imported services belong to.
* Use teams to group services by department, organization, or
  type of service and control access to the services.
* This setting is required, and the input does not run if the setting is
```

```
  not present.
* There is no default.

index_earliest = <integer>
* Specify the earliest _indextime, in minutes, for the time range of your search.
* This setting is required if you import from a search string
  (if you set 'import_from_search' to "true").
* Default: -15m

index_latest = <integer>
* Specify the latest _indextime, in minutes, for the time range of your search.
* This setting is required if you import from a search string
  (if you set 'import_from_search' to "true").
* Default: now

entity_title_field = <string>
* The column name in the CSV file, or the field in the search, to import
  the entity title from.
* This field serves as the informal identifier of the entity.
* There is no default.

entity_merge_field = <string>
* The column name in the CSV file, or the field in the search, to import
  the entity merge field from.
* There is no default.

entity_relationship_spec = <dict>
* A dictionary of key:value pairs that specifies how
  'entity_title_field' associates with other fields and in what relationship.
* NOTE: This setting is unused.
* For example,
  {"hosts": "vm1, vm2", "hostedBy": "host_id"}, or
  {"hosts": ["vm1", "vm2"], "hostedBy": "host_id"}.
* For a record that has values for fields: vm1, vm2, host_id,
  <'entity_title_field' value>, three relationships are extracted:
  <value for 'entity_title_field'> hosts <value for vm1>
  <value for 'entity_title_field'> hosts <value for vm2>
  <value for 'entity_title_field'> hostedBy <value for host_id>
* There is no default.

selected_services = <comma-separated list>
* A list of existing services to associate the imported entities with.
* DEPRECATED.
* There is no default.

service_rel = <comma-separated list>
* A list of existing service relationships.
* DEPRECATED.
* Use this setting to represent service dependencies in ITSI.
* There is no default.

service_dependents = <comma-separated list>
* A list of child columns in the CSV file, or child fields in the search,
  that indicate service dependencies.
* There is no default.

entity_service_columns = <comma-separated list>
* A list of services found in the CSV file or search that are to be
  associated with the entity for the row.
* DEPRECATED.
* There is no default.
```

```
entity_identifier_fields = <comma-separated list>
* A list of columns found in the CSV file or fields in the search
  that identify the entities (entity aliases).
* There is no default.

entity_description_column = <comma-separated list>
* A list of columns found in the CSV file or fields in the search
  that describe the entities.
* There is no default.

entity_informational_fields = <comma-separated list>
* A list of informational columns in the CSV file or fields in the search.
* These are non-identifying fields for the entities.
* There is no default.

entity_field_mapping = <key-value pairs>
* A key-value mapping of fields to re-map to other fields in your data.
* Follows a <CSV field> = <Splunk search field> format.
* For example, ip1 = dest, ip2 = dest, storage_type = volume
* Use this setting to rename a field or column to an alias or info value.
* There is no default.

service_title_field = <string>
* The field to import the service title from.
* This field is the informal identifier of the service.
* There is no default.
* This setting is required if you import services.

service_description_column = <comma-separated list>
* A list of columns in the CSV file or fields in the search
  that describe the services.
* There is no default.

service_tags_field = <comma-separated list>
* A list of columns in the CSV file or fields in the search
  that add descriptor tags to the services.
* There is no default.

service_enabled = <boolean>
* Whether or not imported services are enabled.
* Default: false

service_template_field = <string>
* This setting determines which service template a service is linked to.
* There is no default.

template = <dict>
* A dictionary of key:value pairs that maps entity rules to service templates.
* For example,
  {"test_template_2":{"entity_rules":[{"rule_items":
  [{"rule_type":"matches","field_type":"alias","field":"whoa","value":"doe"}],
  "rule_condition":"AND"}]},"test_template_1":{"entity_rules":[{"rule_items":
  [{"rule_type":"matches","field_type":"alias","field":"blah","value":"da"}],
  "rule_condition":"AND"}]}}
* CAUTION: Do not change this setting.
* There is no default.

backfill_enabled = <boolean>
* This setting determines whether to enable backfill on all
  Key Performance Indicators (KPIs) in linked service templates.
* Backfill is the process of getting historical KPI data.
* ITSI backfills the KPI summary index (itsi_summary). You must have
```

```
  indexed adequate raw data for the backfill period.
* There is no default.

update_type = <APPEND|UPSERT|REPLACE>
* The update/insertion method when uploading entities.
* This setting is required, and the input will not run if the setting is
  not present.
* APPEND: ITSI makes no attempt to identify commonalities between entities.
  All information is appended to the table.
* UPSERT: ITSI appends new entries.  Existing entries (based on the value
  found in the title_field) have additional information appended
  to the existing record.
* REPLACE: ITSI appends new entries. Existing entries (based on the value
  found in the title_field) are replaced by the new record value.
* There is no default.

interval = <integer>
* The interval, in seconds, that determines how often this input runs.
* There is no default.

[itsi_async_csv_loader]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_async_csv_loader://<name>]
* A modular input that periodically uploads CSV data into the KV store.
* The file must contain headers for the import to work properly.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: WARN

import_from_search = <boolean>
* Indicates whether to import data from a CSV file or a Splunk search.
* If "true", this input imports data from the search specified by 'search_string'.
* If "false", this input imports CSV data from the path specified by  'csv_location'.
* This setting is required, and the input does not run if the setting is
  not present.
* There is no default.

csv_location = <path>
* The location on disk of the CSV file to import.
* NOTE: The disk must be local to the search head. Cloud storage is unacceptable.
* This setting is required if you import data from a CSV file
  (if you set 'import_from_search' to "false").
* There is no default.

search_string = <string>
* The Splunk search string that generates the data to import.
* This setting is required if you import from a search string
  (if you set 'import_from_search' to "true").
* There is no default.

index_earliest = <integer>
* Specify the earliest _indextime, in minutes, for the time range of your search.
* This setting is required if you import from a search string
  (if you set 'import_from_search' to "true").
* Default: -15m

index_latest = <integer>
* Specify the latest _indextime, in minutes, for the time range of your search.
* This setting is required if you import from a search string
  (if you set 'import_from_search' to "true").
* Default: now

entity_title_field = <string>
* The column name in the CSV file, or the field in the search, to import
  the entity title from.
* This field serves as the informal identifier of the entity.
* There is no default.

entity_merge_field = <string>
* The column name in the CSV file, or the field in the search, to import
  the entity merge field from.
* There is no default.

entity_relationship_spec = <dict>
* A dictionary of key:value pairs that specifies how
  'entity_title_field' associates with other fields and in what relationship.
* NOTE: This setting is unused.
* For example,
  {"hosts": "vm1, vm2", "hostedBy": "host_id"}, or
  {"hosts": ["vm1", "vm2"], "hostedBy": "host_id"}.
* For a record that has values for fields: vm1, vm2, host_id,
  <'entity_title_field' value>, three relationships are extracted:
  <value for 'entity_title_field'> hosts <value for vm1>
  <value for 'entity_title_field'> hosts <value for vm2>
  <value for 'entity_title_field'> hostedBy <value for host_id>
* There is no default.

selected_services = <comma-separated list>
* A list of existing services to associate the imported entities with.
* DEPRECATED.
* There is no default.

service_rel = <comma-separated list>
* A list of existing service relationships.
* DEPRECATED.
* Use this setting to represent service dependencies in ITSI.
* There is no default.

service_dependents = <comma-separated list>
* A list of child columns in the CSV file, or child fields in the search,
  that indicate service dependencies.
* There is no default.

entity_service_columns = <comma-separated list>
* A list of services found in the CSV file or search that are to be
  associated with the entity for the row.
* DEPRECATED.
* There is no default.

entity_identifier_fields = <comma-separated list>
* A list of columns found in the CSV file or fields in the search
  that identify the entities (entity aliases).
* There is no default.

entity_description_column = <comma-separated list>
* A list of columns found in the CSV file or fields in the search
  that describe the entities.
* There is no default.

entity_informational_fields = <comma-separated list>
```

```
* A list of informational columns in the CSV file or fields in the search.
* These are non-identifying fields for the entities.
* There is no default.

entity_field_mapping = <key-value pairs>
* A key-value mapping of fields to re-map to other fields in your data.
* Follows a <CSV field> = <Splunk search field> format.
* For example, ip1 = dest, ip2 = dest, storage_type = volume
* Use this setting to rename a field or column to an alias or info value.
* There is no default.

service_title_field = <string>
* The field to import the service title from.
* This field is the informal identifier of the service.
* There is no default.
* This setting is required if you import services.

service_description_column = <comma-separated list>
* A list of columns in the CSV file or fields in the search
  that describe the services.
* There is no default.

service_tags_field = <comma-separated list>
* A list of columns in the CSV file or fields in the search
  that add descriptor tags to the services.
* There is no default.

update_type = <APPEND|UPSERT|REPLACE>
* The update/insertion method when uploading entities.
* This setting is required, and the input will not run if the setting is
  not present.
* APPEND: ITSI makes no attempt to identify commonalities between entities.
  All information is appended to the table.
* UPSERT: ITSI appends new entries.  Existing entries (based on the value
  found in the title_field) have additional information appended
  to the existing record.
* REPLACE: ITSI appends new entries. Existing entries (based on the value
  found in the title_field) are replaced by the new record value.
* There is no default.

[itsi_migration_queue]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_migration_queue://<name>]
* A modular input that checks the ITSI migration queue
* If the queue is not empty, start a migration with params stored in the queue.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_refresher]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_refresher://<name>]
* A modular input that processes deferred methods using a single queue processor.
* Tracks relational objects and dependencies.
* This input detects conflicts and ensures consistency across ITSI.
```

```
log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_consumer]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_consumer://<name>]
* A modular input that processes deferred methods using multiple queues
  across the Splunk environment.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

number_of_thread = <integer>
* Number of threads enabled for certain refresh queue jobs.
* 0 or 1 means a single thread.
* Default: 8

[itsi_backup_restore]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_backup_restore://<name>]
* A modular input that performs backup and restore operations by
  managing backup/restore jobs.
* If you restore ITSI from a backup of an older version of ITSI,
  migration begins during the restore process.
* The input runs runs every 5 seconds to check for the scheduled job.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_scheduled_backup_caller]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_scheduled_backup_caller://<name>]
* A modular input that manages ITSI backup schedules.
* For example, you might use this input if you want to back up ITSI
  every night at 1 am.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_service_template_update_scheduler]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_service_template_update_scheduler://<name>]
* A modular input that performs a scheduled sync from
  service templates to services every 15 minutes.
```

```
log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_backfill]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_backfill://<name>]
* A modular input that manages KPI backfill jobs.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_notable_event_archive]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_notable_event_archive://<name>]
* A modular input that moves notable events from the KV store
  to the index every hour.

owner = <string>
* Splunk cannot read the modular name unless a parameter is specified.
  Therefore, ITSI passes 'owner = <string>'.

[maintenance_minder]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[maintenance_minder://<name>]
* A modular input that runs every 60 seconds and populates
  the operative maintenance log based on configured maintenance windows.
* This input is responsible for putting services into maintenance mode.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_default_aggregation_policy_loader]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_default_aggregation_policy_loader://<name>]
* A modular input that loads the default aggregation policy.
* The default aggregation policy receives notable events that do
  not match the filtering criteria of any other aggregation policies.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_default_correlation_search_acl_loader]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.
```

```
[itsi_default_correlation_search_acl_loader://<name>]
* A modular input that loads the Access Control List (ACL)
  for the default correlation searches provided with ITSI:
  "Monitor Critical Services Based on Health Score",
  "Splunk App for Infrastructure Alerts", and
  "Normalized Correlation Search".
* This input pulls ACL information from the KV store.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_notable_event_hec_init]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_notable_event_hec_init://<name>]
* A modular input that initializes HEC client on a search head by creating and
  showing pertinent HEC tokens.
* A new HEC token is acquired during a Splunk restart.
* The internal system populates the new HEC token automatically.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_notable_event_actions_queue_consumer]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_notable_event_actions_queue_consumer://name]
* A modular input that acts as a consumer of the queue for executing
  notable event actions, such as pinging a host or running a script.
* This setting is primarily used by the rules engine.

exec_delay_time = <integer>
* The amount of time, in seconds, to delay execution of a notable event action.
* Default: 0

batch_size = <integer>
* The number of jobs to pick up in a single request from the
  notable event actions queue.
* Default: 5

timeout = <integer>
* The timeout period, in seconds, that ITSI uses when a
  user reclaims an expired job.
* Default: 7200 (2 hours)

system_user_name = <string>
* The username of the system.
* Default: splunk-system-user

[itsi_entity_exchange_consumer]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_entity_exchange_consumer://name]
* A modular input that consumes entities from the entity exchange module.
```

```
log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of the modular input.
* Default: DEBUG

interval = <value>
* The interval, in seconds, at which the modular input should run.
* Optional
* Default: 300 (5 minutes)

[itsi_age_kpi_alert_value_cache]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_age_kpi_alert_value_cache://<name>]
* A modular input that cleans up the aged entries in the KPI summary cache.

retentionTimeInSec = <integer>
* Aging/retention time for entries present in the KPI summary cache.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

[itsi_summary_metrics_backfill]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_summary_metrics_backfill://<name>]
* A modular input that migrates data from the itsi_summary index to the
  itsi_summary_metrics index by checking the metrics_backfill queue.

disabled = <boolean>
* Whether or not the modular input for metrics backfill is disabled
* Default : 1

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

metrics_backfill_throttle = <integer>
* The amount of time, in seconds, that the backfill function pauses between executing metrics backfill
searches.
* Default: 10

metrics_backfill_length = <integer>
* The amount of time, in days, that the metrics backfill searches look back to migrate data
  into the itsi_summary_metrics index.
* Default: 3

metrics_backfill_concurrent_searches = <integer>
* The number of concurrent searches the backfill function runs at the same time. Having more
  concurrent searches allows backfill searches to complete faster but puts more load on the indexers.

[itsi_suite_enforcer]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_suite_enforcer://<name>]
```

```
* A modular input that enforces suite editions.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

interval = <integer>
* The interval, in seconds, that determines how often this input runs.
* There is no default.

[itsi_backfill_record_cleanup]
python.version = {default|python|python2|python3}
* In Splunk Enterprise version 8.0 and later, this attribute lets you select
  which Python version to use.

[itsi_backfill_record_cleanup://<name>]
* A modular input that enforces suite editions.

log_level = <DEBUG|INFO|WARN|ERROR>
* The logging level of this input.
* Default: INFO

interval = <integer>
* The interval, in seconds, that determines how often this input runs.
* There is no default.
```

inputs.conf.example

```
No example
```

itsi_base_service_template.conf

The following are the spec and example files for `itsi_base_service_template.conf`.

itsi_base_service_template.conf.spec

```
# This file contains possible settings you can use to upload sample
# base service templates to the KV store.
#
# There is an itsi_base_service_template.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default. To set custom
# configurations, place an itsi_base_service_template.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
# You must restart ITSI to enable new configurations.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles.
```

[<name>]

```
title = <string>
* The title of the service template.

description = <string>
* A description of the service template.

_owner = <string>
```

```
* The owner of the service template.
* Default: itsi

_immutable = <boolean>
* Whether the service template can be edited or deleted.
* If "true", the service template cannot be edited or deleted.
* If "false", the service template can be edited or deleted.
* Default: false

entity_rules = <json>
* A list of entity rules (rules specification) used to associate entities
  to services created from this service template.
* This setting is the same as the 'entity_rules' setting in itsi_service.conf.spec.
* Example:
        [\
            {\
                "rule_condition": "AND", \
                "rule_items": [\
                    {\
                        "field": "app_title", \
                        "field_type": "alias", \
                        "rule_type": "not", \
                        "value": ""\
                    }, \
                    {\
                        "field": "itsi_role", \
                        "field_type": "info", \
                        "rule_type": "matches", \
                        "value": "apm"\
                    }, \
                    {\
                        "field": "type", \
                        "field_type": "info", \
                        "rule_type": "matches", \
                        "value": "application"\
                    }\
                ]\
            }\
        ]

kpis = <json>
* A JSON blob that specifies the array of KPI definitions.
* For an example, see itsi_base_service_template.conf.
```

itsi_base_service_template.conf.example

```
No example
```

itsi_da.conf

The following are the spec and example files for `itsi_da.conf`.

itsi_da.conf.spec

```
# Copyright (C) 2005-2020 Splunk Inc. All Rights Reserved.
#
# This configuration file is DEPRECATED.
# For [entity_source_template://<string>], use inputs.conf/[itsi_csv_import://<name>] instead.
# For [service_template://<string>], use itsi_service_template.conf/[string] instead.
#
# This file contains possible settings you can use to configure an itsi_da.conf file. Use this
# file to configure an app to export entity searches and service templates for use within the
# IT Service Intelligence (ITSI) app.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# 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.
```

[entity_source_template://<string>]

```
title = <string>
* The display name of the search.

description = <string>
* A human-readable description of this search.

saved_search = <string>
* The actual Splunk saved search that outputs a table. This will be enforced by
  client-side code.

title_field = <string>
* A single field that acts as the title for the entity.

description_fields = <comma-separated list>
* A list of fields that describe the entity.

identifier_fields = <comma-separated list>
* A list of fields that identify the entity.

informational_fields = <comma-separated list>
* A list of fields that act as additional entity metadata.
```

[service_template://<string>]

```
title = <string>
* A title for the service template.

description = <string>
* The full description of the service being created.

entity_source_templates = <comma-separated list>
* The list of entity searches that create entities that can be used with this service.
* The list is used to populate the list of entity searches in the combined
  entity-service creation workflow.

entity_rules = <string>
* A list of entity rules (rules specification) used to associate entities to service
  created from this template.
* This field is the same as the entity_rules field in itsi_service.conf.spec.

recommended_kpis = <comma-separated list>
* A list of KPIs that are automatically added when a service is created with this template.
```

```
informational_kpis = <comma-separated list>
* A list of informational (no threshold) KPIs that are automatically added when a
  service is created with this template.

optional_kpis = <comma-separated list>
* A list of KPIs that are available for this service (but not added automatically).
```

itsi_da.conf.example

```
No example
```

itsi_data_integrations.conf

The following are the spec and example files for `itsi_data_integrations.conf`.

itsi_data_integrations.conf.spec

```
# This file contains the list of data integrations that will be presented
# as chiclets on the data integrations page.

# To set custom configurations, place a itsi_data_integrations.conf.spec in
# $SPLUNK_HOME/etc/apps/itsi/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

[<data_integration_name>]

```
title = <string>
* The title for the data integration chiclet.
* Required.

description = <string>
* The description for the data integration chiclet.
* Required.

icon_path = <string>
* The icon for the data integration chiclet.
* Required.
```

itsi_data_integrations.conf.example

```
No example
```

itsi_deep_dive.conf

The following are the spec and example files for `itsi_deep_dive.conf`.

itsi_deep_dive.conf.spec

```
# This file contains possible attributes and values for uploading sample
# deep dives to the KV store.
#
# To upload deep dives to the KV store, place an
# itsi_deep_dive.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
#
# You must restart Splunk software to enable configurations, unless you are
# 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# WARNING: Manual editing of this file is not recommended. Contact Support before proceeding.
```

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>]

```
* A name or primary identifier for the deep dive.

focus_id = <string>
* The ID of the entity or service that is in focus in the deep dive.
* When an entity or service has focus, you see a list of metrics
 (performance metrics, event counts) for that entity/service.
* Any particular deep dive can have a particular IT context
  in focus at any given time.
* You can change the IT context in focus at any time. However,
  changing the focus has implications for historical tracking
  if not in a named deep dive.

title = <string>
* The title of the deep dive that is displayed in the UI.

lane_settings_collection = <array>
* An array of lane settings specifying each lane's configuration.

acl = <value>
* The team Access Control List (ACL) settings.

mod_time = <value>
* The last time the 'acl' setting was modified.
```

```
description = <value>
* Optional. The description of the deep dive.

is_named = <true|false>
* Whether or not this deep dive is named.
* A deep dive is considered "named" if you save it in the UI
  and give it a name. You might name a deep dive if you find it
  particularly useful and want to save it for future use.
* A deep dive is considered "unnamed" if you dynamically generated
  it (for example, from a drilldown) and did not save it.

_owner = <string>
* The user's KV store account in which to store the deep dive.
* In nearly all cases this value is "nobody".

source_itsi_da = <string>
* Optional. The ITSI module that acts as the source to define
  the deep dive.
* This attribute is used by the domain add-ons.
```

itsi_deep_dive.conf.example

```
No example
```

itsi_entity_type.conf

The following are the spec and example files for `itsi_entity_type.conf`.

itsi_entity_type.conf.spec

```
# This file contains possible settings you can use to upload sample
# entity types to the KV store.
#
# An entity type defines how to classify a type of data source.
# For example, you can create a Windows, Kubernetes, or VMware vCenter Server entity type.
# An entity type can include zero or more entity data drilldowns and zero or more entity data dashboards.
#
# There is an itsi_entity_type.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default. To set custom
# configurations, place an itsi_entity_type.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
# You must restart ITSI to enable new configurations.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles.
```

[<name>]

```
title = <string>
* Required
* Title of the entity type.

description = <string>
* Description of the entity type.

dashboard_drilldowns = <json array>
* Required. If no value empty list
* A list of dashboard drilldowns that entities of this class can use to associate with raw data.
* A single dashbobard drilldown JSON object contains the following fields
{
    "title": <string>
    * Usage:
        * Required
        * The title of the dashboard.

    "id" = <string>
    * Usage:
        * Required
        * A unique ID for the dashboard drilldown.

    "is_splunk_dashboard" = <boolean>
    * Usage:
        * Required
        * A flag to determine whether the dashboard drilldown is saved as a navigation or a splunk
dashboard.

    "base_url": <string>
    * Usage:
        * An internal or external URL pointing to the dashboard.

    "params": <json>
    * Usage:
        * Contains two fields: 'alias_param_map' and 'static_params'.
        * 'alias_param_map' is a mapping of a URL parameter and its alias.
        * 'static_params' are parameters with a defined value.
        * Example:
            {
                "static_params": {
                    "start_time": "-12h",
                },
                "alias_param_map": [
                    {
                        "alias": "host",
                        "param": "node"
                    }
                ]
            }
}

data_drilldowns = <json array>
* A list of data drilldowns that entities of this class can use to populate pre-built dashboards.
* A single data drilldown JSON object contains the following fields
{
    "title": <string>
    * Usage:
        * Required
        * The title of the entity data drilldown.

    "type": <metrics|events>
    * Usage:
        * Required
        * The type of indexed data that this drilldown is associated with.
        * Must be either "metrics" or "events".

    "static_filter": <json>
    * Usage:
        * An SPL filter represented by a JSON structure following a defined schema.
        * The static filter finds a subset of indexed data that is associated with
          this entity data drilldown.
        * There are two types of filters for a static_filter:
          1. Basic filter - fields including:
            - type: One of "include" or "exclude"
            - field: The field name in raw data
            - values: A list of values for "field" to filter on
          2. Boolean filter - fields including:
            - type: One of "or" or "and"
            - filters: A list of filters in the shape of a basic filter or boolean filter

        * The following example filter is equivalent to "sourcetype=access_logs AND index=main":
        { \
            "type": "and", \
            "filters": [ \
                { \
                    "type": "include", \
                    "field": "sourcetype", \
                    "values": ["access_logs"] \
                }, \
                { \
                    "type": "include", \
                    "field": "index", \
                    "values": ["main"] \
                } \
            ] \
        }

    "entity_field_filter": <json>
    * Usage:
        * Specifies what field (info or alias) of an entity to apply
          to further filter down the indexed data.
        * There are two types of filters for an entity_field_filter:
          1. Entity field filter - fields including:
            - type: Must be "entity"
            - data_field: The field name in raw data
            - entity_field: The field of an entity whose value will be used to filter on raw data with
"data_field"
          2. Boolean filter - fields including:
            - type: One of "or" or "and"
            - filters: A list of filters in the shape of a entity field filter or boolean filter

        * Example:
        { \
            "type": "or", \
            "filters": [ \
                { \
                    "type": "entity", \
                    "data_field": "src", \
                    "entity_field": "ip" \
                }, \
                { \
                    "type": "entity", \
                    "data_field": "dest", \
                    "entity_field": "ip" \
                } \
            ] \
        }
        * For an entity with "ip=1.2.3.4", this is equivalent to "src=1.2.3.4 OR dest=1.2.3.4".
        * Combined with the static filter example above, the final filter of this entity data drilldown
          is equivalent to "(sourcetype=access_logs AND index=main) AND (src=1.2.3.4 OR dest=1.2.3.4)"
}

vital_metrics = <json array>
```

```
* Optional
* A list of vital metrics that entities of this class are associated with.
{
    "metric_name": <string>
    * Usage:
        * Required
        * The name of the metric.

    "search" = <string>
    * Usage:
        * Required
        * SPL to find this metric.

    "split_by_fields": <array>
    * Usage:
        * Required
        * An array of fields used to split the results to entities.

    "matching_entity_fields": <array>
    * Usage:
        * Required
        * The fields used to look up entities from the KV store.
        * Example: split_by_fields=[id,name], matching_entity_fields=[id,host]
        * Raw event "id" field maps to "id" field of entity, and "name" field maps to "host" field

    "is_key": <boolean>
    * Usage:
        * Optional
        * If "true", this metric is used as a key metric for this entity type in the Infrastructure
Overview.
        * Default: false

    "unit": <string>
    * Usage:
        * Optional
        * The unit for the metric.
}

_immutable = <boolean>
* Required
* Whether you can edit or delete the entity data drilldown.
* If "true", you can't edit or delete the entity data drilldown.
* If "false", you can edit or delete the entity data drilldown.
* Default: false
```

itsi_entity_type.conf.example

```
No example
```

itsi_event_management.conf

The following are the spec and example files for `itsi_event_management.conf`.

itsi_event_management.conf.spec

```
# This file contains attributes and values for configuring different ITSI
# event management features.
```

```
#
# There is an itsi_event_management.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an itsi_event_management.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[<stanza_name>]

```
* A setting that you want to enable for Episode Review.
* Supported settings (stanzas) are 'similar_episodes' and 'common_fields'
```

[similar_episodes]

```
default_fields = <comma-seperated list>
* The list of field names selected by default in Similar Episodes pane
* For example, ["title","description","host"]
* Default: ["title"]
```

[common_fields]

```
number_of_fields = <integer|all>
* The number of common fields to display on the Common Fields tab of an episode.
* Can be a positive integer or the word "all" to display all common fields.
* For example, "50" displays 50 common fields.
* Default: 50
```

[migration]

```
The settings in this stanza apply to upgrades from pre-4.6.0 ITSI versions to
version 4.6.0 or later. The settings support the addition of the following
fields to the itsi_notable_group_system KV store collection: parent_group_id,
split_by_hash, first_event_id, and group_template_id. If you are upgrading from
ITSI version 4.6.0 or later, these settings no longer apply.

kv_store_batch_size = <integer>
* The maximum batch size of fetch requests to the itsi_notable_group_system
```

```
  KV store collection.
* For example, if set to "10000", 10,000 objects are fetched
  from the KV store in a single fetch request.
* Default: 10000

cluster_manager_check_required = <integer>
* Whether a cluster manager check is required before migration starts.
* If set to "1", a cluster manager check is required.
* If set to "0", migration proceeds without a cluster manager check.
* Default: 1

itsi_grouped_alerts_index_lookback = <integer>
* The amount of time, in days, to look back to fetch old active groups from the itsi_grouped_alerts index.
* For example, if set to "60", active groups from last two months are fetched from the index.
* Default: 90

itsi_grouped_alerts_index_search_wait_time = <integer>
* The amount of time, in seconds, to wait for the search job to return results from the itsi_grouped_alerts
index.
* For example, if set to "900", the search job will wait for 15 minutes to return results from the index.
* Default: 7200
```

[precheck]

```
The settings in this stanza apply to upgrades from pre-4.6.0 ITSI versions to
version 4.6.0 or later. The settings suppport the prechecks that runs before
the migration happens.

kv_store_collection_size_limit = <integer>
* The maximum number of a single object type allowed in any KV store collection.
* For example, if set to "1000000", 1000000 objects of a single type are allowed in a KV store collection.
* Default: 1000000
```

itsi_event_management.conf.example

```
No example
```

itsi_glass_table.conf

The following are the spec and example files for `itsi_glass_table.conf`.

itsi_glass_table.conf.spec

```
# This file contains possible attributes and values for uploading sample
# glass tables to the KV store.
#
# To upload glass tables to the KV store, place an
# itsi_glass_table.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
#
# You must restart Splunk software to enable configurations, unless you are
# 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
```

```
# WARNING: Manual editing of this file is not recommended. Contact Support before proceeding.
```

[<name>]

```
svg_content = <value>
* The SVG content settings.

latest = <value>
* The latest time in the time range.

earliest = <value>
* The earliest time in the time range.

svg_coordinates = <value>
* The SVG coordinate settings.

title = <string>
* The user-defined title of the glass table.

description = <string>
* The user defined description of the glass table.

mod_time = <value>
* Last modified time.

acl = <value>
* Access control information.

_owner = <string>
* The user account this deep dive belongs to.

source_itsi_da = <string>
* The ITSI module which is the source defining this glass table.
```

itsi_glass_table.conf.example

```
No example
```

itsi_kpi_base_search.conf

The following are the spec and example files for `itsi_kpi_base_search.conf`.

itsi_kpi_base_search.conf.spec

```
# This file contains possible settings you can use to upload sample
# KPI base searches to the KV store.
#
# There is an itsi_kpi_base_search.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default. To set custom
# configurations, place an itsi_kpi_base_search.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
# You must restart ITSI to enable new configurations.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles.
```

[<name>]

```
description = <string>
* A description of the KPI base search.

title = <string>
* The title of the KPI base search

_owner = <string>
* The owner of this KPI base search
* Default: itsi

base_search = <search>
* The search to execute in the KPI base search. This search is the source
  for the fields defined in metrics.
* Set the search to "*" if the 'is_metric' setting is set to "true".

metrics = <json>
* A JSON blob that specifies the array of metrics to be collected.
* Example item in the blob:
* {
*     "unit": "%",
*     "title": "CPU Utilization: %",
*     "entity_statop": "avg",
*     "aggregate_statop": "avg",
*     "_key": "620b26a6f286a508fd356d94",
*     "threshold_field": "cpu_load_percent"
*  }
* The threshold_field in the item corresponds to a field from the base search.

is_metric = <boolean>
* Whether the KPI base search is a metric search.

metric = <json>
* A JSON blob that specifies the metric index and metric name to search on.
* Example item in the blob:
* {
*     "metric_index": "em_metrics",
*     "metric_name": "cpu.user"
* }

is_entity_breakdown = <boolean>
* Whether the metrics should be broken down by entities for
  threshold calculations.
* If "1", metrics are broken down by entities.
* If "0", metrics are not broken down by entities.

is_service_entity_filter = <boolean>
* Whether metrics should filter out entities not in the service.
* If "1", entities that don't belong to the service are filtered out.
* IF "0", entities that don't belong to the service are still included.

entity_id_fields = <string>
* The field in the base search used to look up the corresponding
  entity to filter KPIs.
* For example, host, ip, and so on.
* This field is required if the 'is_service_entity_filter' setting is set to "true".

entity_breakdown_id_fields = <string>
* The field in the base search used to look up the corresponding entity
  to split KPIs.
```

```
* For example, host, ip, and so on.
* This field is required if the 'is_entity_breakdown' setting is set to "true".

entity_alias_filtering_fields = <comma-separated list>
* A list of alias attributes to be used to filter out entities not in the service.
* Optional.
* This field is required if the 'is_service_entity_filter' setting is set to "true".

alert_period = <integer>
* The frequency, in minutes, at which to run the search.

search_alert_earliest = <integer>
* The time window, in minutes, over which to evaluate the metrics.

alert_lag = <integer>
* The amount of time, in seconds, to push back the metric evaluation.
* This setting corresponds to the data indexing lag.
* Default: 30

metric_qualifier = <string>
* The field in the base search used to further split metrics.
* CAUTION: You cannot modify this setting in the UI.

source_itsi_da = <string>
* The ITSI module that is the source defining this KPI base search.
```

itsi_kpi_base_search.conf.example

```
No example
```

itsi_kpi_template.conf

The following are the spec and example files for `itsi_kpi_template.conf`.

itsi_kpi_template.conf.spec

```
# This file contains possible settings you can use to upload sample
# KPI templates to the KV store.
#
# There is an itsi_kpi_template.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default. To set custom
# configurations, place an itsi_kpi_template.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
# You must restart ITSI to enable new configurations.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles.
```

[<name>]

```
description = <string>
* The description of the KPI template bundle.

title = <string>
* The title of the bundle.

_owner = <string>
```

```
* The owner of the bundle.

kpis = <json>
* A JSON blob that specifies the array of KPI definitions.

source_itsi_da = <string>
* The ITSI module that is the source defining this KPI template.
```

itsi_kpi_template.conf.example

```
No example
```

itsi_kpi_threshold_template.conf

The following are the spec and example files for `itsi_kpi_threshold_template.conf`.

itsi_kpi_threshold_template.conf.spec

```
# This file contains possible settings you can use to upload sample
# KPI threshold templates to the KV store.
#
# There is an itsi_kpi_threshold_template.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default. To set custom
# configurations, place an itsi_kpi_threshold_template.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/local.
# You must restart ITSI to enable new configurations.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles.
```

[<kpi_threshold_template>]

```
* Each stanza represents a KPI threshold template.
* Use threshold templates to build your time policies.
* You can create different policies with different time block
  combinations, such as work hours, off hours, or weekends.

title = <string>
* The title of the KPI threshold template.

description = <string>
* A description of the KPI threshold template.

time_variate_thresholds_specification = <JSON>
* A JSON blob containing the detailed time variant threshold object.

acl = <JSON>
* A JSON blob containing the ACL information for the KPI threshold template.
* Use the following format:
     {
        "perms":{
           "read":[
               <LIST_OF_ROLES>
           ],
           "write":[
               <LIST_OF_ROLES>
           ]
        },
        "can_share_user":[true|false],
        "can_share_app":[true|false],
        "modifiable":[true|false],
        "sharing":["app"|"global"],
        "can_change_perms":[true|false],
        "can_share_global":[true|false],
        "owner": <OWNER_NAME_STRING>,
        "can_write":[true|false]
     }

time_variate_thresholds = [True|False]
* Whether to enable time-variate thresholds.
* Time-variate thresholds accommodate normal variations in usage across
  your services and improve the accuracy of KPI and service health scores.
* For example, a time-variate threshold might take into account higher levels
  of usage during work hours, and lower levels of usage during off-hours
  and weekends.
* Default: True

adaptive_thresholding_training_window = <-7|-14|-30|-60>[d]
* The time window over which historical KPI data is analyzed for
  adaptive threshold updates.
* You must have 7 days of summary data in the summary index for
  adaptive thresholding to work properly.
* Default: -7 days

adaptive_thresholds_is_enabled = [True|False]
* Whether to enable adaptive thresholding for policies in time-variate thresholds.
* Adaptive thresholding lets you create time polices that generate thresholds
  dynamically and update daily based on changes in your data.
* If you set this value to "true", the 'time_variate_thresholds' setting must
  also be set to "true".
* Default: False
```

itsi_kpi_threshold_template.conf.example

```
No example
```

itsi_module_settings.conf

The following are the spec and example files for `itsi_module_settings.conf`.

itsi_module_settings.conf.spec

```
# This file contains a setting for determining whether a module is editable
# in the module lister page.
#
# There is an itsi_module_settings.conf in each individual module directory (for example,
# $SPLUNK_HOME/etc/apps/DA-ITSI-OS/default for the Operating System module). To change this
# setting for a specific module, place an itsi_module_settings.conf in
$SPLUNK_HOME/etc/apps/<module>/local.
# 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

[settings://<app>]

```
* "app" is the ID for the app that contains this configuration file.

is_read_only = <boolean>
* Whether the module shows as editable in the module lister page.
* If "1", the module is not editable in the module lister page.
* If "0", the module is editable in the module lister page.
* Default: 1
```

itsi_module_settings.conf.example

```
No example
```

itsi_module_viz.conf

The following are the spec and example files for `itsi_module_viz.conf`.

itsi_module_viz.conf.spec

```
# This file contains possible attributes and values for changing tab names and panel
# titles in a module details dashboard.
#
# There is an itsi_module_viz.conf in each module-specific directory within ITSI (for example,
# $SPLUNK_HOME/etc/apps/DA-ITSI-OS/default for the Operating System module). To edit these
# configurations, place an itsi_module_viz.conf in $SPLUNK_HOME/etc/apps/DA-ITSI-OS/local.
# 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# WARNING: Manual editing of this file is not recommended. Contact Support before proceeding.
```

[<view_name>]

```
* The name of the deep dive drilldown view within the ITSI module.

tabs = <comma-separated list>
* A list of tab IDs that will be included in this drilldown view.

<tabId>.control_token = <string>
* Used to run all the panel searches in a given tab.
* When the tab is shown, a list of search tokens are retrieved, the search tokens for
  all inactive tabs are removed from the list, and the search token for the active tab
  is added to the list. This guarantees that only the shown tab's panels are displayed.

<tabId>.title = <string>
* The title of the tab that is displayed in the UI.

<tabId>.row.<int> = <comma-separated list>
* A list of panels that are displayed on each row on a tab.
* The panels are formatted as follows: <module_name>:<panel_name>.
* These settings start at 'row.0' and go up to any number of rows that is needed for a tab.
* Example:
        row.0 = DA-ITSI-OS:panel1,DA-ITSI-LB:panel2

<tabId>.extendable_tab = <boolean>
* Whether the tab is considered an extendable tab.
* This setting is for user-created tabs so that a delete button appears on the tab
  in the UI.
* Any tabs that ship with the module default to "false".

<tabId>.activation_rule = <comma-separated list>
* A list of KPI elements that are associated with a given tab so that
  context-aware drilldown is enabled based on the selected KPI from the deep dive.
* Each element here is defined as the content from the "target_field" parameter from each
  selected KPI from the file itsi_kpi_template.conf.

entity_search_filter = <JSON>
* A JSON blob of entity rules to use to filter entities for entity dropdown.

requested_entity_tokens = <comma-separated list>
* A list of entity attributes that are submitted as tokens.
```

itsi_module_viz.conf.example

```
No example
```

itsi_notable_event_retention.conf

The following are the spec and example files for `itsi_notable_event_retention.conf`.

itsi_notable_event_retention.conf.spec

```
# This file contains attributes and values for defining how long notable event metadata remains
# in the KV store before it moves to the 'itsi_notable_archive' index.
#
# There is an itsi_notable_event_retention.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an itsi_notable_event_retention.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[<collection_name>]

```
retentionTimeInSec = <seconds>
* The amount of time, in seconds, to retain the notable event object type.
* Default: 15768000 (6 months)

retentionObjectCount = <number of objects>
* The maximum number of a single object type (for example, notable_event_comment) to retain in the KV store.
* The retention policy runs every hour. If the retention count is 500,000 and a KV store collection exceeds
500,000 objects,
* the oldest objects are moved to the archive index so the collection only has 500,000 objects.
* Default: 500000

disabled = 0|1
* Whether this stanza is enabled or disabled.
* This setting only applies when the 'smart_recycling' setting is disabled.
  If smart recycling is enabled, this parameter is ignored.
* If "1", the stanza is disabled.
* If "0", the stanza is enabled.

object_type = <string>
* The notable event object type to retain.
* For example, comments, tags, external tickets, and so on.
* Required.
* If 'object_type' is not specified, the entire stanza is ignored.

batch_size = <integer>
* The size of each batch of KV store objects recycled to the archive index at a time.
* Default: 1000

event_push_iterations = <integer>
* The maximum number of attempts to push events from a single KV store collection into the archive index.
* If this limit is reached, the retention policy moves on to the next collection.
* Only set this property in the global [default] stanza. It cannot be set on a per-collection basis.
* Default: 20

warning_percentage = <integer>
* The percentage of retentionObjectCount needed to trigger a capacity warning message.
* Default: 90

smart_recycling = <boolean>
* Enable or disable the smart retention policy, which chooses to recycle inactive episodes and
  related objects first before recycling other objects. Smart retention begins when the limits for the
  'retentionTimeInSec' setting or the 'retentionObjectCount' settings are exceeded.
* If "1", smart recycling is enabled.
* If "0", smart recycling is disabled.
* Only set this property in the global [default] stanza. It cannot be set on a per-collection basis.
* Default: 1 (enabled)

recycle_remaining = <boolean>
* When smart_recycling is enabled, this setting determines whether to continue recycling objects
  after all inactive and closed episodes (and their related objects) have been recycled.
* If "1", the archiver continues to recycle remaining objects until the retentionObjectCount limit is
reached.
* If "0", the archiver stops the archival process after all inactive and closed episodes have been archived.
* Enabling this setting might archive some old active episodes and their related objects, but it can prevent
  over-exhausting system resources by making sure the KV store collection sizes don't exceed their limits.
* Default: 1 for itsi_notable_group_system and itsi_notable_group_user, 0 for other KV store collections.
```

itsi_notable_event_retention.conf.example

```
No example
```

itsi_notable_event_severity.conf

The following are the spec and example files for `itsi_notable_event_severity.conf`.

itsi_notable_event_severity.conf.spec

```
# This file contains attributes and values for defining the colors associated with
# different episode and event severity levels in Episode Review.
#
# There is an itsi_notable_event_severity.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an itsi_notable_event_severity.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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>]

```
color = <string>
* A valid color code to represent an episode or event with this severity.
* The color determines the episode's color in Episode Review.
* Required.

lightcolor = <string>
* A valid color code to represent an episode's severity in prominent mode.
* Required.

label = <string>
* The severity label displayed in Episode Review.
* For example, Info, Medium, Critical.

default = 0|1
* Set this flag to indicate the default severity of an event or episode.
```

itsi_notable_event_severity.conf.example

```
No example
```

itsi_notable_event_status.conf

The following are the spec and example files for `itsi_notable_event_status.conf`.

itsi_notable_event_status.conf.spec

```
# This file contains attributes and values for configuring label descriptions
# and episode status in Episode Review.
#
# There is an itsi_notable_event_status.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an itsi_notable_event_status.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[<id>]

```
label = <string>
* A valid label for the episode status.
* Required.

default = <boolean>
* Indicates the initial status of an episode when it is generated in
  Episode Review.
* Set this value to "1" if this label is the default label.

description = <string>
* A description of the episode label.

end = <boolean>
* Indicates the last status in the Episode Review workflow.
* Set this value to "1" if this label is the end of the
  episode management workflow.
* If a status has an end flag enabled, any episode with that status is automatically
```

```
  broken. This means that no more events will flow into that episode. This rule
  applies to status changes in Episode Review as well as through aggregation
  policy action rules.
* CAUTION: If you remove the "end" tag from the "Closed" status, you will no
  longer be able to close episodes through the Episode Review UI. It is
  recommended that you do not remove or change the location of this tag.
```

itsi_notable_event_status.conf.example

```
[default]
disabled = 0
label =
description =
default = 0
end = 0

[0]
label = Unknown
description = An error is preventing the issue from having a valid status assignment

## Enable status "new"
## Enable selected (automatically selects status element in applicable UI pulldowns)
[1]
disabled = 0
default = 1
label = New
description = Event has not been reviewed

## Enable status "in progress"
[2]
disabled = 0
label = In Progress
description = Investigation or response is in-process

## Enable status "pending"
[3]
disabled = 0
label = Pending
description = Event closure is pending some action

## Enable status "resolved"
[4]
disabled = 0
label = Resolved
description = The issue has been resolved and awaits verification

## Enable status "closed"
[5]
disabled = 0
label = Closed
description = Issue has been resolved and verified
end = 1
```

itsi_service.conf

The following are the spec and example files for `itsi_service.conf`.

itsi_service.conf.spec

```
# This file contains attributes and values for uploading services to the KV store.
#
# There is an itsi_service.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an itsi_service.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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>]

```
description = <string>
* A description of the service.

title = <string>
* The title of the service.

services_depends_on = <value>
* Any services that this service depends upon.

services_depending_on_me = <value>
* The fields to be represented in the entity.

_owner = <string>
* The owner of the service.

tags = <value>
* Some tags for the service.

kpis = <value>
* Entity rules for the servce.

entity_rules = <value>
* A list of entity rules used to associate entities to a service.
```

```
identifying_name = <value>
* A field to contain the unique name for the service.

mod_source = <value>
* A field only used by logging, where the edit came from.

source_itsi_da = <value>
* The ITSI module which is the source defining this deep dive.
```

itsi_service.conf.example

```
No example
```

itsi_service_analyzer.conf

The following are the spec and example files for `itsi_service_analyzer.conf`.

itsi_service_analyzer.conf.spec

```
# This file contains attributes and values for configuring the
# auto-refresh interval, or disabling auto-refresh.
#
# There is an itsi_service_analyzer.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an itsi_service_analyzer.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[<stanza_name>]

```
* A setting that you want to enable for the Service Analyzer.
* Currently, 'auto_refresh' is the only supported setting.
* NOTE: Auto-refresh is automatically disabled in real-time search mode
  for the Service Analyzer.

disabled = 0|1
```

```
* Whether this setting is disabled for the Service Analyzer.
* Required.
* If "1", the setting is disabled.
* If "0", the setting is enabled.
* Default: 1

interval = <seconds>
* The interval, in seconds, at which auto-refresh occurs for Service Analyzer.
* Required.
* Default: 120 (2 minutes)
```

itsi_service_analyzer.conf.example

```
No example
```

itsi_service_template.conf

The following are the spec and example files for `itsi_service_template.conf`.

itsi_service_template.conf.spec

```
# Copyright (C) 2005-2020 Splunk Inc. All Rights Reserved.
#
# This file contains attributes and values for creating and uploading modules
# in Splunk IT Service Intelligence (ITSI).
#
# To set custom configurations, place an itsi_service_template.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/local/. 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# 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, 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.
```

[<stanza_name>]

```
title = <string>
* A title of the module.

description = <string>
* The full description of the module being created.

entity_rules = <json>
* A JSON blob of entity rules (rules specification) used to associate entities
  to services created from this module.
* This setting is the same as the 'entity_rules' setting in itsi_service.conf.spec.
* Example:
        [\
            {\
                "rule_condition": "AND", \
                "rule_items": [\
                    {\
                        "field": "app_title", \
                        "field_type": "alias", \
                        "rule_type": "not", \
                        "value": ""\
                    }, \
                    {\
                        "field": "itsi_role", \
                        "field_type": "info", \
                        "rule_type": "matches", \
                        "value": "apm"\
                    }, \
                    {\
                        "field": "type", \
                        "field_type": "info", \
                        "rule_type": "matches", \
                        "value": "application"\
                    }\
                ]\
            }\
        ]

recommended_kpis = <json>
* A list of KPIs that are automatically added when a service
  is created with this module.

optional_kpis = <json>
* A list of KPIs that are available with this module but
  not added automatically when a service is created with it.
```

itsi_service_template.conf.example

```
No example
```

itsi_settings.conf

The following are the spec and example files for `itsi_settings.conf`.

itsi_settings.conf.spec

```
# Copyright (C) 2005-2020 Splunk Inc. All Rights Reserved.
# This file contains attributes and values for configuring the IT Service
# Intelligence (ITSI) app.
#
# There is an itsi_settings.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place an itsi_settings.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/local/. 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# 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, 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.
```

[datamodels://<app>]

```
* 'app' is the ID for the app containing the datamodel.

blacklist = <datamodel_names_list>
* A pipe-separated list of data model external authentication interface
  (EAI) names (IDs) to blacklist.
* NOTE: Data model names do not contain pipe characters.
* The blacklisted data models will not be supported and remain hidden
  from the ITSI UI.
```

[cloud]

```
show_migration_message  = <boolean>
* Removes Cloud migration messages about deprecated files or apps from
  the logs because this process is done internally.
```

[backup_restore]

```
* Defines settings related to ITSI backup/restore.
```

```
job_queue_timeout = <seconds>
* The amount of time, in seconds, before the backup/restore job queue
  times out if the node owning the job has been down for too long to
  allow other jobs to proceed.
* The minimum supported timeout period is 3600 seconds (1 hour). The system
  sets the timeout to 3600 seconds when a value lower than this is set.
* Default: 43200 (12 hours)
```

[import]

```
* Defines limits for import behavior.

import_batch_size = <integer>
* The minimum number of objects the importer should analyze before
  attempting a save to the KV store.
* Default: 1000

preview_sample_limit = <integer>
* The maximum number of rows that are returned from a preview request
  for a pending import.
* Default: 100

asynchronous_processing_threshold = <integer>
* The number of rows after which the bulk importer reads and stores the
  inbound content so that it can be processed at a more convenient time,
  rather than processing it immediately.
```

[metric_backfill]

```
* Defines backfill settings.

pre_calculation_window = <seconds>
* The size, in seconds, of the pre-calculation window for metric backfill.
* The smallest accepted value is 1. Increasing this value makes the
  backfill search faster, but less accurate.
* Default: 1
```

[sai_integration]

```
* Defines Splunk App for Infrastructure (SAI) settings.

show_detection_modal = <boolean>
* Whether or not to show the Splunk App for Infrastructure integration
  modal when the Service Analyzer loads.
* If "1", ITSI displays the integration modal.
* If "0", ITSI does not display the integration modal.
* Default: 1
```

[synced_kpi_scheduling]

```
disabled = <boolean>
* Indicates whether KPI saved searches have a randomized schedule or the same schedule.
* If "1", KPI saved searches run at staggered times throughout the scheduled interval.
* If "0", KPI saved searches all run at the same time during each scheduled interval.
* CAUTION: Changing this value to "0" can have a significant performance impact. KPI saved
* searches are designed to run at different times to prevent the search scheduler
```

```
  from becoming overloaded.
* Default = 1
```

[customsearch]

```
timeout_read = <seconds>
* The maximum number of seconds that an ITSI custom search command will attempt to
  read a chunk from the "chunked" custom search command protocol.
* Default: 3600
```

[episode_action_dispatch]

```
* Enables the ability to dispatch actions from a Splunk instance to be executed on
  another instance.
* Configure these settings in this stanza if you want to specify whether this Splunk
  instance will read actions and execute actions from another instance or dispatch
  actions to another Splunk instance.
* The settings in this stanza define the host's role. If configured as an 'executor' they
  also define the URI and username of the host for consuming Event Analytics episode actions.

role = <executor|manager|both>
* Whether the machine is executing actions, running core event analytics
  services, or both.
* If "executor", the host is only executing actions.
* If "manager", the host is only running core event analytics services.
* Default: both

remote_ea_mgmt_uri = <string>
* The Splunkd management URI from which to pull action jobs, in addition
  to other core event analytics services.
* The URI must include a scheme, host, and port.
* If an empty string, ITSI uses the local Splunk address to avoid the
  necessity of an update if a custom port or scheme is in use on the local
  Splunk instance.
* This setting is only required if 'role' is set to "executor".
* Default: empty string

remote_ea_username = <string>
* The username to use when communicating with the remote host for
  actions and updates.
* If you're on localhost, ITSI always uses the past session from Splunkd
  (the provided username is ignored in this case).
* This setting is only required if 'role' is set to "executor".
* Default: empty string
```

[lock]

```
service_template_sync_in_progress = <boolean>
* Whether a service template is currently syncing.
* If "1", at least one service template is syncing and it is not safe to upgrade.
* If "0", no service templates are syncing and it is safe to upgrade.
* CAUTION: Do not change this setting. It is updated dynamically by ITSI.
* Default: 0
```

[suite_configuration]

```
suite_level = <string>
* ITSI Cloud Suite level.
* CAUTION: This is not user changeable setting.
```

itsi_settings.conf.example

```
No example
```

rest

```
# Time out value for the rest calls.
rest_timeout = 300
```

itsi_team.conf

The following are the spec and example files for `itsi_team.conf`.

itsi_team.conf.spec

```
# This file contains attributes and values for uploading ITSI teams
# to the KV store. By default, only the Global team ships with ITSI.
#
# There is an itsi_team.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default.
# To set custom configurations, place an itsi_team.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/local/. 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# 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, 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.
```

[default_itsi_security_group]

```
title = <value>
* The name of the team.
* Duplicate team names are allowed, but be aware of other team names
  and use naming conventions to avoid confusion.

description = <value>
* A meaningful description of the team.

_immutable = <boolean>
* Whether users can edit the team.
* If "1", the team cannot be edited.
* If "0", the team can be edited.
* Default: 0

acl = <dictionary>
* An Access Control List (ACL) associating ITOA roles with permissions
  within that team.
* Assign read or write access to the listed ITOA roles as appropriate.
  If a role has write permissions for a team, a user with this role can
  create and modify services in the team. The user can't delete a service
  in the team unless the role has the delete capability for a service.
```

[notable_event_review_security_group]

```
disabled = <boolean>
* Use this setting to turn off Role-Based Access Control (RBAC) for Episode
  Review only.
* If you set this flag to "1", all users will be able to see all events
  within Episode Review, regardless of their team.
* If "1", RBAC is disabled for Episode Review.
* If "0", RBAC is enabled for Episode Review.
* Default: 0
```

itsi_team.conf.example

```
No example
```

limits.conf

The following are the spec and example files for `limits.conf`.

limits.conf.spec

```
#   Version 8.1.0
#
```

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

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 maxresults, 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 'maxresults'.
    * If the memory limit is exceeded, output is truncated, not spilled to disk.
  * The stats command processor uses the 'max_mem_usage_mb' value in the
    following way.
    * If the estimated memory usage exceeds the specified limit, the results are
      spilled to disk.
    * If 0 is specified, the results are spilled to the disk when the number of
      results exceed 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 parameter, 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: false

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: false

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: false

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: false

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: false

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
```

[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 are 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
  below.
* This limit should not exceed 50000.
* 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.

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.

max_infocsv_messages  = <positive integer>
* 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.

show_warn_on_filtered_indexes = <boolean>
* Log warnings if search returns no results because user has
  no permissions to search on queried indexes.
```

[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>
* Controls the number of search pipelines that are
  launched at the indexer during batch search.
* Increasing the number of search pipelines should help improve search
  performance, however there will be an increase in thread and memory usage.
* This setting applies only to searches that run on remote indexers.
* 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]
* A flag that 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.

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
```

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 MB, 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.
```

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.
* Default: 1000
```

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.

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_datamodel_meval = <boolean>
* Enable concatenation of successively occurring evals into a single
  comma-separated eval during the generation of datamodel searches.
* default: true
```

```
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

max_id_length = <integer>
* Maximum length of the custom search job ID when spawned by using
  REST API argument "id".

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 retries parts of a search within a
  currently-running search process when there are indexer failures in the
  indexer clustering environment.
* Indexers can fail during rolling restart or indexer upgrade when indexer
  clustering is enabled. Indexer reboots can also result in failures.
* This setting applies only to historical search in batch mode, real-time
  search, and indexed real-time search.
* When set to true, the Splunk software attempts to rerun searches on indexer
  cluster nodes that go down and come back up again. The search process on the
  search head maintains state information about the indexers and buckets.
* NOTE: Search retry is on a best-effort basis, and it is possible
  for Splunk software to return partial results for searches
  without warning when you enable this setting.
* When set to false, the search process will stop returning results from
  a specific indexer when that indexer undergoes a failure.
* Default: false

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>
* Controls whether to use bloom filters to rule out buckets.
* 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: 30

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

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
```

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]
* 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 <ratio>
* Minimum amount of time between results preview updates.
* If specified as a number, between > 0 and  < 1, the minimum time between
  previews is computed as a ratio of the amount of time that the search
  has been running, or as a ratio of the length of the time window for
  real-time windowed searches.
* Default: a ratio of 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.
* 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

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 "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 "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.
* Normally the "idle_process_cache_search count" and the
  "idle_process_cache_timeout" settings will keep the regex cache a
  reasonable size.  This setting is to prevent 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 "auto" or
  if "max_searches_per_process" is set to 0 or 1.
* Default: 2500

idle_process_reaper_period = <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 "auto" or
  if "max_searches_per_process" is set to 0 or 1.
* Default: 30

launcher_max_idle_checks = <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 "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 "auto" or
  if "max_searches_per_process" is set to 0 or 1.
* Default: -1 (a value is selected automatically)

max_old_bundle_idle_time = <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 a 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 "auto" or
  if "max_searches_per_process" is set to 0 or 1.
* Default: 5

max_searches_per_process = <integer>
* On UNIX, 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 "auto".
* Default: 500

max_time_per_process = <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 "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: 300 (5 minutes)

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 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 "auto" or
  if "max_searches_per_process" is set to 0 or 1.
* NOTE: A search can run longer than the 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 = <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 "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: false

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.
```

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: 100000

maxvaluesize = <integer>
* Maximum size, in bytes, of any single value (truncated to this size if
  larger).
* Default: 1000
```

[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: 10000

maxvaluesize = <integer>
* Maximum length of a single value to consider.
* Default: 1000
```

[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
```

[concurrency]

```
batch_search_max_pipeline = <integer>
* Controls the number of search pipelines launched at the indexer during
  batch search.
* Increasing the number of search pipelines should help improve search
  performance but there will be an increase in thread and memory usage.
* This value applies only to searches that run on remote indexers.
* Default: 1

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
```

[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
```

[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
```

[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: 100000

maxvaluesize = <integer>
* Maximum length of a single value to consider.
* Default: 1000
```

[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

list_maxsize = <integer>
* Maximum number of list items to emit when using the list() function
  stats/sistats
* Default: 100

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

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.
```

[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

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
```

[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
```

[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

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
```

[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.
```

[authtokens]

```
expiration_time = <integer>
* Expiration time, in seconds, of auth tokens.
* Default: 3600 (60 minutes)
```

[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"  parameter 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
```

[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.
* The summary fields 'host', 'index', 'source', 'sourcetype', 'eventtype',
  'linecount', 'splunk_server', and 'splunk_server_group' do not count against
  this limit and will always be returned.
* Increase this setting if, for example, you have data with a large
  number of columns and want to ensure that searches display all fields extracted
  from an automatic key-value field (auto kv) configuration.
* Set this value to 0 if you do not want to limit the number of fields
  that can be extracted at index time and search time.
* Default: 100

indexed_kv_limit = <integer>
* The maximum number of fields that can be extracted at index time from a data source.
* 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 searches display all fields from
  the data.
* 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>
* 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: 1000

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 the status of KV Store migration is polled
  on each search head cluster member after the start of the migration.
* 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 (interval length being determined
  by the "periodic_timer_interval" setting) that a search head cluster member's
  status can remain in failed state during KV Store migration before retrying
  migration on the member. If the trial number has hit the max retry limit,
  then the member is marked as aborted.
* Once this limit is reached, the migration is aborted on the member.
* Default: 10
```

[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: 30

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
```

[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 parameters for time_format and output_time_format which match
  this regex will be rejected.
* The regex will be satisfied by a substring match anywhere in the parameter.
* 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 parameter strings will be 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
```

[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 10.
* Default: 2

actions_queue_size = <integer>
* The number of alert notifications to queue before the scheduler starts
  blocking, set to 0 for infinite size.
* Default: 100

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

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>
* 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.
* NOTE: event signing and verification is not supported in distributed mode
* 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: 60
```

```
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

sleep_seconds = <integer>
* The amount of time, in seconds, to sleep between polling the summarization
  complete status.
* 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::pr_job_extractor]

```
enabled = <boolean>
* Enables a search language optimization that converts a search string with a
  'prjob' command into a search string with 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::dfs_job_extractor]

```
enabled = <boolean>
* Enables Splunk software to identify portions of searches and send them to
  the DFS cluster for fast processing.
* Can only be used by Splunk platform implementations that have enabled Data
  Fabric Search (DFS) functionality.
* Default: true

commands = <Command List>
* A comma-separated list of search commands that are affected by DFS
  job extraction.
* Default: The full list of commands supported by DFS.

commands_add = <Command List>
* A comma-separated list of search commands to be added to the list of commands supported by DFS
  Note: This setting is always processed after the 'commands' setting.
* Default: None

commands_rm = <Command List>
* A comma-separated list of search commands to be removed from the list of commands supported by DFS
  Note: This setting is always processed after the 'commands' and 'commands_add' settings.
* Default: None
```

[search_optimization::projection_elimination]

```
enabled = <boolean>
* Enables projection elimination optimization
* Default: true

cmds_black_list = <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 setting if you are dependent on summary indexes. When it
  is enabled, searches that perform stats operations on summary indexes and
  which only reference indexed fields will return incorrect results. This
  occurs because the 'tstats' command does not respect the fields created by
  summary indexing commands. If you are using summary indexing but still choose
  to enable this optimization globally, this optimization can be disabled on
  a per-search basis by appending
  '| noop search_optimization.replace_stats_cmds_with_tstats=f' to the search
  string.
* Default: false
```

[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
```

[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: true

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: true

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: true
```

[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: 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 that has been configured as an intermediate reducer.
* If you specify 0, there is no limit. The indexer runs as many parallel
  reduce searches as the indexer hardware permits.
* Default: 1

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

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

rdinPairingTimeout = <positive integer>
* The amount of time (in seconds) to wait so that indexers and intermediate indexers may get paired
* Default: 300
```

[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 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
```

Data Fabric Search

[dfs]

```
* The settings in this stanza specify aspects of the Data Fabric Search
  (DFS) cluster.

dfc_control_port = <port>
* Sets the listening port for data fabric coordinator (DFC) processes. Enables
  communication between a DFC process and a corresponding search process (SP).
* The port number is internally auto-incremented by Splunk software when the
  default port is unavailable. If this happens, limits.conf is not updated with
  the selected port number.
* The maximum number of DFC control ports that can be used for data fabric
  search at any given time is set by dfc_num_slots.
* Default: 17000

dfc_num_slots = <integer>
* Sets the maximum number of data fabric coordinator (DFC) processes that can run
  concurrently on each search head. Each process uses a search head 'slot'.
* Default: 4

dfs_max_num_keepalives = <integer>
* Sets the maximum number of keepalive packets to run the DFS search.
* Default: 10

dfs_max_reduce_partition_size = <integer>
* Sets the maximum number of partition size to receive data to run the DFS search.
* Recommended setting for executor node with 5 Cores and 12GB memory: 150000.
* Default: 500000

dfs_max_search_result_size = <integer>
* Sets the maximum number of results which a DFS search returns.
* When this value is zero (0), a DFS search returns all the results.
* Default: 1000000

dfw_num_slots = <integer>
* This setting applies only when 'dfw_num_slots_enabled' is set to "true" or
  when search head clustering is enabled in your Splunk implementation.
  * If you have enabled search head clustering, this setting sets the maximum
    number of data fabric coordinator (DFC) processes that can run concurrently
    across the search head cluster.
  * If you have disabled search head clustering, the value of 'dfw_num_slots'
    is equal to 'dfc_num_slots'.
* When multiple deployments are utilizing the same DFS cluster, this setting
  can help resolve concurrent search issues.
* Default : 10

dfw_num_slots_enabled = <boolean>
* Set this to "true" to enable the use of 'dfw_num_slots'.
* Default: false

dfs_resource_awareness = <boolean>
* Available Spark resources are continuously monitored to provide admission control for
  data fabric searches.
* Default: true

dfs_post_proc_speedup = <boolean>
* The post processing on the indexers is sped up by parallelizing post processing
* Default: false

dfs_num_post_proc_speedup_threads = <integer>
* The number of threads dedicated to speed up post process on remote pipelines.
* Default: 1

dfs_post_proc_input_queue_size = <integer>
* The size of queue that holds the chunks that need to be post processed
* Default: 400

dfs_post_proc_output_queue_size = <integer>
* The size of queue that holds the post processed results that need to go through rest of the pipeline
* Default: 400

dfs_estimation_time = <integer>
* The amount of time (in seconds) prior to a Data Fabric Search getting scheduled that the
  event count estimation is calculated.
* Default: 300

dfw_receiving_data_port = <port>
* Sets the listening port for data fabric worker (DFW) nodes. Receives
  redistributed data from Splunk indexers.
* The port number is internally auto-incremented by Splunk software when the
  default port is unavailable. If this happens, limits.conf is not updated with
  the selected port number.
* Default: 17500

dfw_receiving_data_port_count = <integer>
* Maximum number of ports that Splunk software checks for availability, starting from
  the default port set in the parameter 'dfw_receiving_data_port'.
* If the 'dfw_receiving_data_port_count' is set to 0, Splunk software checks for any
  available port without any upper limit.
* Default: 0

dfs_remote_search_timeout = <integer>
* The amount of time (in seconds) to wait because the search run on the
  DFS worker has not received the new results from any of the indexers.
```

```
* Default: 600

dfs_max_remote_pipeline = <integer>
* Controls the number of search pipelines launched at the indexer during a DFS search.
* Increasing the number of search pipelines typically helps improve search performance,
  but requires additional thread and memory usage.
* Depending on data volume and cardinality, modifying this setting
  may lead to slower searches or unread records.
* Default: 12

dfs_meta_phase_exec_timeout = <integer>
* The amount of time, in seconds, to wait for various meta phase processes to complete
  during a federated search.
* Default: 300

dfs_enable_parallel_serializer = <boolean>
* Enable the DFS parallel serializer to dispatch data more efficiently from the
  indexers to the DFS executors.
* The DFS parallel serializer can support multi-threaded processing to dispatch data,
  which might increase CPU and memory usage but improves performance as opposed to
  the legacy DFS search.
* Default: true

dfs_num_of_remote_serializer_pipeline = <integer>
* Sets the number of DFS remote serializer pipelines.
* DFS serializer pipelines transmit intermediate search results from indexers to
  DFS executors.
* Modifying this setting can lead to slower searches, depending on data volume,
  cardinality, and CPU numbers.
* If not set, DFS uses only one serializer pipeline.
* Default: 1

dfs_remote_io_kickout_period = <positive integer>
* The maximum amount of time(in milliseconds) to wait for next I/O write.
* Decreasing the time period typically increases the I/O rate of sending
  results from indexers to executors, but at the cost of extra CPU cycles.
* Max period is 1000 milliseconds, min period is 5 milliseconds.
* Default: 20

enable_dfs_search_feedback = <boolean>
* This setting can enable dfs search feedback at the end of a search.
* Default: true

enable_dfs_search_fallback = <boolean>
* This setting can enable search engine selection; Allow DFS searches
  to fallback to legacy Splunk Enterprise search.
* Default: false

dfs_eventcount_limit = <integer>
* The setting sets the event count boundary for running search via DFS;
* Feedback regarding DFS candidacy is provided based on this event count limit.
* Queries that exceed this event count and contain commands that are dfs compatible
  will trigger a notification via warning message
* Default: 20000000
```

[segmenter]

```
use_segmenter_v2 = <bool>
* 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
```

limits.conf.example

```
#   Version 8.1.0
# 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

# If number of cpu's in your machine is 14 then total system wide number of
# concurrent searches this machine can handle is 20.
# which is base_max_searches + max_searches_per_cpu x num_cpus = 6 + 14 x 1 = 20
base_max_searches = 6
max_searches_per_cpu = 1

[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
cmds_black_list = 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 = false

[search_optimization::replace_datamodel_stats_cmds_with_tstats]
enabled = true
```

```
[search_optimization::dfs_job_extractor]
enabled = true

[dfs]
dfc_control_port = 17000
dfc_num_slots = 4
dfs_max_num_keepalives = 10
dfs_max_reduce_partition_size = 150000
dfs_max_search_result_size = 1000000
dfw_num_slots = 10
dfw_num_slots_enabled = true
dfw_receiving_data_port = 17500
dfs_max_num_keepalives = 10
dfw_receiving_data_port_count = 0
```

logging.conf

The following are the spec and example files for `logging.conf`.

logging.conf.spec

```
# The format and semantics of this file are described in this article at Python.org:
#
# [Configuration file
format](https://docs.python.org/2/library/logging.config.html#configuration-file-format)
#
# This file must contain sections called [loggers], [handlers] and
# [formatters] which identify by name the entities of each type which are defined in the
# file. For each such entity, there is a separate section which identifies how that entity
# is configured.

keys = <a list of available keys separated by comma>
* appears in [loggers], [handlers] or [formatters]
* describes the available [logger_<name>], [handler_<name>] or [formatter_<name>]

level = <DEBUG|INFO|WARNING|ERROR|CRITICAL|NOTSET>
*  For the root logger only, NOTSET means that all messages will be logged.

handlers = <comma-separated list>
*  A comma-separated list of handler names, which must appear in the
   [handlers] section.
*  These names must appear in the [handlers] section and have corresponding
   sections in the configuration file.

qualname = <string>
*  The hierarchical channel name of the logger (the name used by the
   application to get the logger).

propagate = <0|1>
*  Set to "1" to indicate that messages must propagate to handlers higher
   up the logger hierarchy from this logger.
*  Set to "0" to indicate that messages are not propagated to handlers
   up the hierarchy.

class = <string>
*  Indicates the handler's class, as determined by eval() in the logging package's namespace.
```

```
args = <comma-separated list>
*  The list of arguments to the constructor for the handler class, when
   eval()uated in the context of the logging package's namespace.

formatter = <string>
*  The key name of the formatter for this handler.
*  If blank, a default formatter (logging._defaultFormatter) is used.
*  If a name is specified, it must appear in the [formatters] section and
   have a corresponding section in the configuration file.

format = <logger format pattern>
* for pattern format see: https://docs.python.org/2/library/logging.config.html#user-defined-objects
```

logging.conf.example

```
No example
```

macros.conf

The following are the spec and example files for `macros.conf`.

macros.conf.spec

```
# Version 8.1.0
#
```

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 8.1.0
#
# 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
```

mad.conf

The following are the spec and example files for `mad.conf`.

mad.conf.spec

```
# This file contains possible settings you can use to configure metric anomaly detection.
# Use anomaly detection to identify trends and outliers in KPI search results that might
# indicate an issue with your system.
#
# There is a mad.conf in $SPLUNK_HOME/etc/apps/SA-ITSI-MetricAD/default. To set custom
# configurations, place a mad.conf in $SPLUNK_HOME/etc/apps/SA-ITSI-MetricAD/local.
#
# To learn more about configuration files (including precedence), see the
# documentation located at
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/ListofITSIconfigurationfiles

# To learn more about metric anomaly detection, see
# http://docs.splunk.com/Documentation/ITSI/latest/Configure/Enableanomalydetection

# In most situations, the default values specified in mad.conf should work as-is.
# Modifying this file can result in negative changes to anomaly detection accuracy.
# Do NOT remove any stanzas or settings in the configuration file.

# For <duration> format, this configuration file accepts the following units:
#    * ms => milliseconds
#    * s, sec, secs, second, seconds => second
#    * m, min, mins, minute, minutes => minute
#    * h, hr, hrs, hour, hours => hour
#    * d, day, days => day
```

[service]

```
unbounded_buffer_size = <duration>
* The size of the data buffer used in batch mode.
* For example, "4d" stores a maximum of 4 days of data.
* Default: 400d

kvstore_connect_interval = <duration>
* How often to retry connecting to the KV store when the connection is lost.
* Default: 30s

rest_ssl_permissive_trustmanager = <boolean>
* Whether to enable PermissiveX509TrustManager with HTTPS connection to Splunk REST API.
* Do not modify this setting unless Splunk is not running in HTTPS mode.
* Default: true

rest_ssl_permissive_hostnameverifier = <boolean>
* Whether hostname verification is strict or permissive.
* If set to "true", hostname verification is permissive.
* If set to "false", hostname verification is strict.
* This setting can be disabled when the Splunk certificate is not self-signed.
* Default: true

trending_bounded_buffer_size = <duration>
* The size of the data buffer for the trending algorithm in real-time mode.
* This setting MUST be larger than the value of the 'training_period'
  setting in the [trending] stanza.
* Default: 15d

cohesive_bounded_rt_buffer_size = <duration>
* The size of the real-time data buffer for the cohesive algorithm in real-time mode.
* Default: 12h

cohesive_bounded_backfill_buffer_size = <duration>
* The size of the backfill data buffer for the cohesive algorithm in real-time mode.
* Default: 25h
```

[trending]

```
* Use this stanza to configure the 'mad' command for the trending algorithm.

periods.days = <positive integer>
* How many days to look back for normal patterns in the data.
* Must be a value greater than zero.
* Default: 6

periods.weeks = <integer>
* How many weeks to look back for normal patterns in the data.
* Must be a value greater than or equal to zero.
* Default: 2

window_size = <positive integer>
* How many data points to use to construct an analysis window.
* Must be a value greater than 1.
* Default: 60

step_size = <positive integer>
* The offset size of two consecutive analysis window.
* Must be a value greater than 0.
* Default: 1

training_period = <duration>
* The amount of time used to train the algorithm.
* Must be a value greater than 1.
* Default: 7d

max_NA_ratio = <float>
* The maximum possible ratio of NaN (undefined) data points.
* Must be a decimal between 0.0 and 1.0.
* Default: 0.5

na_rm = <boolean>
* Whether or not to remove NaN (undefined) data points.
* If set to "true", NaN data points are removed.
* Default: true

Nkeep = <duration>
* How much data to keep in memory for analysis.
* Default: 50h

Naccum = <float>
* The accumulation score for anomaly alerting.
* Must be a value greater than zero.
* Default: 35.0
```

[trending:limits]

```
* Use this stanza to configure the 'naccum' command for trending algorithm.

Naccum_max = <float>
* The maximum accumulation score to use for detecting anomalies.
* This value MUST be larger than the 'Naccum' setting in the [trending] stanza.
* Default: 50.0

Naccum_min = <float>
* The minimum accumulation score to use for detecting anomalies.
* This value MUST be smaller than the 'Naccum' in the [trending] stanza.
* Default: 30.0

sensitivity_max = <integer>
* The number of sensitivity levels.
* Must be a value greater than 1.
* Default: 10
```

[cohesive]

```
* Use this stanza to configure the 'mad' command for the cohesive algorithm.

window_size = <positive integer>
* How many data points to use to construct an analysis window.
* Must be a value greater than 1.
* Default: 60

step_size = <positive integer>
* The offset size of two consecutive analysis windows.
* Must be a value greater than 0.
* Default: 1

training_period = <duration>
* The amount of time used to train the algorithm.
* Must be a value greater than 1.
* Default: 7d

max_NA_ratio = <float>
* The maximum possible ratio of NaN (undefined) data points.
* Must be a decimal between 0.0 and 1.0.
* Default: 0.5

na_rm = <boolean>
* Whether or not to remove NaN (undefined) data points.
* If set to "true", NaN data points are removed.
* Default: true

Nkeep = <duration>
* How much data to keep in memory for analysis.
* Default: 10h

Naccum = <float>
* The accumulation score for anomaly alerting.
* Must be a number greater than zero.
* Default: 35.0

norm_Ntrend = <integer>
* The window of moving median for normalization of incoming data.
* Default: 10

norm_maxNAratio = <float>
* The maximum ratio of NaN data points allowed in the dataset for normalization of incoming data.
* Must be a decimal between 0.0 and 1.0.
* Default: 0.5

norm_trendOnly = <boolean>
* Whether to use only the trend of the data for normalization.
* Default: false

norm_MAratio = 0.8
* The moving average ratio of the normalization window.
* Must be a decimal between 0.0 and 1.0.
* Default: 0.8

norm_NArm = <boolean>
* Whether to remove NaN (undefined) data points for normalization.
* Default: false

norm_Nwindow = <integer>
* The size, in data points, of the normalization buffer.
* Default: 10080

norm_Nshift = <integer>
* The interval at which the normalization constants are recalculated.
* After receiving this many data points, the constants are recalculated.
* Default: 1440

norm_Ninit = <integer>
* The number of data points needed to calculate the normalization constants.
* Default: 30

norm_batch = <boolean>
* Deprecated option
* Enable/disable batch normalization

metrics_maximum = <integer>
* The maximum number of metrics that can be analyzed for the cohesive algorithm.
* Default: 30
```

[cohesive:limits]

```
* Use this stanza to configure the 'naccum' command for the cohesive algorithm.

Naccum_max = <float>
* The maximum accumulation score that can be used for detecting anomalies.
* This value MUST be larger than the 'Naccum' setting in the [cohesive] stanza.
* Default: 50.0

Naccum_min = <float>
* The minimum accumulation score that can be used for detecting anomalies.
* This value MUST be smaller than the 'Naccum' setting in the [cohesive] stanza.
* Default: 30.0

sensitivity_max = <integer>
* The number of sensitivity levels.
* Must be a value greater than 1.
* Default: 10
```

[logging]

```
* Use this stanza to configure logging.

metric_registry = <boolean>
* Enable logging metrics of the 'mad' command.
* CAUTION: Enabling this setting will have a significant performance impact.
* Default: false
```

[alerting]

```
* Use this stanza to configure external HTTP endpoint connections for posting alerts.

rest_ssl_permissive_trustmanager = <boolean>
* Whether to enable PermissiveX509TrustManager with HTTPS connection to the Splunk REST API.
* Default: true

rest_ssl_permissive_hostnameverifier = <boolean>
* Whether to be strict or permissive in hostname verification.
* If set to "true", hostname verification is permissive.
* If set to "false", hostname verification is strict.
* Default: true

max_http_connection = 100
* How many simultaneous HTTP connections are allowed.
* Default: 100
```

mad.conf.example

```
No example
```

notable_event_actions.conf

The following are the spec and example files for `notable_event_actions.conf`.

notable_event_actions.conf.spec

```
# This file contains attributes and values for taking actions on episodes
# in Episode Review.
#
# There is a notable_event_actions.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place a notable_event_actions.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.

disabled = <boolean>
```

```
* Disable a notable event action by setting to 1.
* Optional.
* Default: 0

is_group_compatible = <boolean>
* Make an action available for episodes by setting to 1.
* Default: 1

is_bulk_compatible = <boolean>
* Make an action available for bulk episodes by setting to 1.
* Default: 0

run_bulk_action_iteratively = <boolean>
* If set to "1", bulk episode actions run iteratively rather than simultaneously.
* This value only takes effect if the 'is_bulk_compatible' setting is set to "1".
* For custom ServiceNow add-ons, this setting must be set to "1"  in order
  for bulk episode actions to function properly.
* Default: 0
```

[<action_name>]

```
* Each stanza represents an episode action. The action name
  is the type of action you want to configure.
* Options are email, script, itsi_sample_event_action_ping,
  itsi_event_action_link_ticket, snow_incident, and remedy_incident.

execute_in_sync = <boolean>
* If 1, ITSI executes the action synchronously.
* The UI notifies you when the action is truly complete, rather
  than requiring you to check back later to confirm.
* It is recommended that you set this value to 1 for an external
  ticket created by a Splunk custom search command or modular alert.
* Default: 0

execute_once_per_group = <boolean>
* If 1, ITSI executes the action exactly once in the case of a
  bulk action.
* In special cases (like if this alert action has 'type' set to "external_ticket"),
  the result of a refresh is associated with all the events in the group.
* Default: 0

type = <string>
* The type of action to take on the episode.
* Use this setting if you are creating a ServiceNow or Remedy ticket from
  an episode.
* The only supported value for this setting is "external_ticket",
  which creates a ticket in the external ticketing system you choose.
* If you set the value to "external_ticket", ITSI runs a refresh action
  right after execution.
* The attribute-value pairs below are applicable when 'type' is "external_ticket".

app_name = <string>
* The name of the app or app-on that runs the action.
* This settings is used to fetch the app version if the alt_command setting is configured.

alt_command_supported_version = <string>
* The version of the app or add-on that supports the alt_command setting, if configured.

alt_command = <string>
* A search command to execute the action instead of the specified action_name.
```

```
ticket_system_name = <string>
* The name of the external ticketing system in which to create the ticket.

relative_refresh_uri = <string>
* A relative URI for the search head where ITSI is installed.
* https://localhost:8089/ or something similar is prepended to the URI.
* ITSI constructs this link so you can navigate directly to the
  external ticket.
* ITSI issues a GET call on this URI and outputs JSON data.
* 'refresh_response_json_path' indicates the path to walk through the
  received JSON output.
* Do not change this from the default value or refresh will not work.

relative_refresh_correlation_key = <string>
* The key used to query the relative_refresh_uri. You only need to change
  this value if the relative_refresh_uri setting doesn't accept the value of
  the 'correlation_key' setting as a query parameter.
* Default: correlation_id

correlation_key = <string>
* Optional. The query parameter to be appended to 'relative_refresh_uri'.
* The parameter is also saved in the KV store collection that contains
  all created tickets.
* Do not change this from the default value or refresh will not work.
* Default: correlation_id

correlation_value = <string>
* The key in the raw notable event whose value to append
  to the refresh URI.
* If a 'correlation_key' exists, ITSI appends this value to the
  refresh URI instead.
* Do not change this from the default value or refresh will not work.
* Default: $result.event_id$

correlation_value_for_group = <string>
* The key in the episode whose value to append
  to the refresh URI.
* By default, ITSI uses the value corresponding to `itsi_group_id'.
* Do not change this from the default value or refresh will not work.
* Default: $result.itsi_group_id$

refresh_response_json_path = <string>
* Because the JSON output of 'relative_refresh_uri' can be nested and
  complex, this setting indicates the path to walk through the received output.
* Do not change this from the default value or refresh will not work.
* Default: entry.{0}.content

refresh_response_ticket_id_key = <string>
* After traversing the JSON path specified in 'refresh_response_json_path'
  and fetching a JSON blob, the key corresponding to the external ticket ID.
* Do not change this from the default value or refresh will not work.

refresh_response_ticket_url_key = <string>
* After traversing the JSON path specified in 'refresh_response_json_path'
  and fetching a JSON blob, the key corresponding to the external ticket URL.
* Do not change this from the default value or refresh will not work.

bulk_max = <string>
* The maximum number of episodes that this action can be executed on.
* Default: 25

send_first_event_only = <boolean>
```

```
* Flag to include only the first event when sending an episode to Phantom.
* If 1, ITSI sends the first event of an episode to Phantom. Otherwise, ITSI sends all events in the
episode.
* Default: 1

splunk_itsi_get_notables_search_api_page_size = <integer>
* The size of each page of results pulled from ITSI.
* Default: 50

phantom_artifacts_create_api_page_size = <integer>
* The size of each page of results pushed to Phantom from ITSI.
* Default: 50

num_parallel_job_slots = <integer>
* The number of slots in the ITSI backend to run parallel jobs for actions.
* Default: 5

job_refresh_interval = <integer>
* The interval, in seconds, that the backend checks for the status of parallel action jobs.
* Default: 2

max_num_internal = <integer>
* The maximum number of intervals to check for scheduled jobs.
* Default: 100

refresh_impact_tab = <boolean>
* Automatically reloads the Impact tab of an episode after an action runs. If set to "1", any tickets or
reference
  links added by the action immediately appear on the Impact tab without having to refresh the page.
* Optional
```

notable_event_actions.conf.example

```
# This is an example notable_event_actions.conf. Use this file to configure
# episode actions.
#
# To use one or more of these configurations, copy the configuration block
# into notable_event_actions.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# This example alert includes running a script, sending an email, and
# creating an incident in ServiceNow.

[email]
disabled = 0

[script]
disabled = 0

[snow_incident]
disabled = 0
type = external_ticket
execute_in_sync = 1
execute_once_per_group = 1

ticket_system_name = ServiceNow
```

```
relative_refresh_uri = /servicesNS/nobody/-/service_now_incident/snow_incident

correlation_key = correlation_id
* Refresh URI now becomes
  /servicesNS/nobody/-/service_now_incident/snow_incident?correlation_id

correlation_value = $result.event_id$
* Assuming we are dealing with an event whose event_id is 'myevent1234',
  the URI now becomes:
  /servicesNS/nobody/-/service_now_incident/snow_incident?correlation_id=myevent1234
* Final URI with output_mode:
  /servicesNS/nobody/-/service_now_incident/snow_incident?correlation_id=myevent1234&output_mode=json
* If there is no 'correlation_key' specified, the final URI looks like:
  /servicesNS/nobody/-/service_now_incident/snow_incident/myevent1234?output_mode=json

correlation_value_for_group = $result.itsi_group_id$
* When operating on an episode, we will use the value corresponding to 'itsi_group_id'
  as the correlation_id. Similar to correlation_value mentioned above.

refresh_response_json_path = entry{0}.content
* Assuming the JSON response looks like the following:
     {
       ...
       "entry": [
         {
           ...
           "content": {
             "number": "INC0047495",
             "url": "https://abc.service-now.com/incident.do?sysparm_query=correlation_id=myevent1234",
             ...
           }
         }
       ],
     ...
     }
  ... the path value is indicative of how to extract the ticket_id and ticket_url.

refresh_response_ticket_id_key = number
* After extracting the JSON blob we are interested in, which looks like the following:
     {
       ...
       "number": "INC0047495",
       "url": "https://abc.service-now.com/incident.do?sysparm_query=correlation_id=myevent1234"
     }
  ... 'number' is the value we are interested in.

refresh_response_ticket_url_key = url
* After extracting the JSON blob we are interested in, which looks like the following:
     {
       ...
       "number": "INC0047495",
       "url": "https://abc.service-now.com/incident.do?sysparm_query=correlation_id=myevent1234"
     }
  ... 'url' is the value we are interested in.
```

notable_event_commonality.conf

The following are the spec and example files for `notable_event_commonality.conf`.

notable_event_commonality.conf.spec

```
# This file contains possible attribute/value pairs for blacklisting
# notable event fields from the Common Fields section of episodes.
#
# There is a notable_event_commonality.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place a notable_event_commonality.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/local. 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

[common_event_fields]

```
black_list_fields = <comma-separated list>
* A list of field names in a notable event that will not appear in the
  Common Fields section of an episode.
* By default, ITSI blacklists fields that are not core to the raw event
  itself, or ones that are mainly used internally.
* Add fields here that you don't necessarily care about, but that you know
  will probably appear in most of your events.
```

notable_event_commonality.conf.example

```
No example
```

notable_event_correlation.conf

The following are the spec and example files for `notable_event_correlation.conf`.

notable_event_correlation.conf.spec

```
# This file contains attributes and values that ITSI Smart Mode uses to correlate
# notable events.
#
# There is a notable_event_correlation.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default/.
# To set custom configurations, place a notable_event_correlation.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/local. 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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

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.
```

[<smart_mode_correlation_engine>]

```
* The settings under this stanza determine how ITSI Smart Mode analyzes notable
  event fields to determine whether they contain textual or categorical content.
* Smart Mode uses machine learning to compare event field values and group
  events that are related to each other.
* CAUTION: This configuration file does not support adding any additional stanzas.
  Do not add, remove, or change any of the settings or stanzas in this file unless
  specifically instructed to by a Splunk support specialist.
```

BLACKLIST FIELDS

```
black_list_fields = <comma-separated list>
* A list of field names in a notable event whose values to discard
  from consideration for Smart Mode event correlation.
```

TEXTUAL FIELDS

```
text_field_names = <comma-separated list>
* A list of field names in a notable event that usually
  represent textual content.
* A text field is a data structure that holds alphanumeric data,
  such as name and address.
* Defaults: comment,description,summoary,review,message

ignore_fields_that_contain = <comma-separated list>
* A list of field names to implicitly ignore because they are not useful
  for event correlation.
* ITSI ignores field names that contain any of the words in this list.
* For example, with the default "time", ITSI ignores fields that represent
  time, like alert_triggertime, alerttriggertime, lasttimeup, etc.
* Default: time

threshold_event_coverage_perc = <int>
* A threshold value for considering a notable event field
  as a text field.
* If the count (total number of occurrences) of a field divided by
  the total number of events processed in the time frame is less
  than the percentage specified in 'threshold_event_coverage_perc',
  then the notable event field is a text field.
* Default: 10
```

CATEGORICAL FIELDS

```
threshold_distinct_value_perc = <int>
* A threshold value for considering a notable event field
  as a categorical field.
* If the distinct_count (count of distinct values) of a field
  divided by the count (total number of occurrences) of the field is
  less than the percentage specified, then the notable event field
  is a categorical field.
* Categorical fields have a distinct value, such as a status field,
  as opposed to textual data, descriptions, numerical values, and comments.
* If this setting determines that a field is NOT a categorical field, ITSI uses
  the two settings below ('min_distinct_value_perc' and 'max_count_perc')
  in a second calculation to check whether the field is a categorical field.
* Default: 35

min_distinct_value_perc = <int>
* Helps confirm whether a notable event field is a categorical field.
* Sets the minimum distinctive value percentage that a notable event field must
  be to be considered a categorical field.
* If the cumulative sum of 'min_distinct_value_perc' of distinct_count (count
  of unique values) of a field is at least 'max_count_perc' of the count
  (total number of occurrences) of the field, then the notable event field is
  considered a categorical field.
* For example, consider the following field:value pairs:
  {field:value1 count:34},{field:value2 count:31}, {field:value3 count:5},
  {field:value4 count:5} , ..., {field:value18 count:1} {field:value19 count:1},
  {field:value20 count:1}
  There are 20 different values listed for this field, so distinct_count = 20.
  ITSI sums the counts of all the values, so count = 80.
  80% of count = 64
  10% of distinct_count = 2, so you add the counts of the first two values above (34 + 31).
  {field:value1 count:34} + {field:value2 count:31} = 34 + 31 = 65 > 64
  Because 65 is at least 64, "field" is a categorical field.
* Default: 10

max_count_perc = <int>
* Helps confirm whether a notable event field is a categorical field.
* Sets the maximum count percentage that a notable event field must
  be to be considered a categorical field.
* If the cumulative sum of 'min_distinct_value_perc' of distinct_count (count
  of unique values) of a field is at least 'max_count_perc' of the count
  (total number of occurrences) of the field, then the notable event field is
  considered a categorical field.
* See the example for the 'min_distinct_value_perc' setting to understand
  how this setting works.
* Default: 80
```

notable_event_correlation.conf.example

```
No example
```

props.conf

The following are the spec and example files for `props.conf`.

props.conf.spec

```
#   Version 8.1.0
#
# 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

# 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

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.
* 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

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 *higher* 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 lower, as 'aaa' is lower 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|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: if you want no field/value extraction to take place.
  * auto: extracts field/value pairs separated by equal signs.
  * auto_escaped: extracts fields/value pairs separated by equal signs and
                  honors \" and \\ as escaped sequences within quoted
                  values, e.g field="value with \"nested\" quotes"
  * multi: invokes the multikv search command to expand a tabular event into
           multiple events.
  * xml : automatically extracts fields from XML data.
  * json: automatically extracts fields from JSON data.
* Setting to 'none' can ensure that one or more user-created regexes are not
  overridden by automatic field/value extraction for a particular host,
  source, or source type, and also increases search performance.
```

```
* The 'xml' and 'json' modes do not extract any fields when used on data
  that isn't of the correct format (JSON or XML).
* 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 for search-time field extractions only.
* Specifies whether to try json extraction automatically.
* 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_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 8.1.0
#
# 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 syslog events.

[syslog]
invalid_cause = archive
unarchive_cmd = gzip -cd -


# 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
```

restmap.conf

The following are the spec and example files for `restmap.conf`.

restmap.conf.spec

```
# Version 8.1.0
#
# 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

[<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.
* (OOPTIONAL) 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>
* If set to "true", sends the driver the HTTP headers of the request.
* 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

[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.

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.

[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.

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.

[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 8.1.0
#
# 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)
```

savedsearches.conf

The following are the spec and example files for `savedsearches.conf`.

savedsearches.conf.spec

```
# This file contains possible attribute/value pairs for saved search entries in
# savedsearches.conf.  You can configure saved searches by creating your own
# savedsearches.conf.
#
# There is a default savedsearches.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/default. To
# set custom configurations, place a savedsearches.conf in
# $SPLUNK_HOME/etc/apps/SA-ITOA/local/. For examples, see
# savedsearches.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
```

Event generator settings

```
action.itsi_event_generator = <boolean>
* Whether the alert is enabled.

action.itsi_event_generator.param.title = <string>
* The title of the notable event in Episode Review.
* Optional. If title is not provided then the search name
  becomes the title.
```

```
action.itsi_event_generator.param.description = <string>
* A description of the notable event.
* Optional. If a description is not provided then the search
  description becomes the event description.

action.itsi_event_generator.param.owner = <string>
* The initial owner of the notable event.
* Optional.
* Default: unassigned

action.itsi_event_generator.param.status = <string>
* The triage status of the event in Episode Review.
* Values must match an integer specified in the default version of
  itsi_notable_event_status.conf (or the local version if you created one).
* Optional.
* Default: 1 (New)

action.itsi_event_generator.param.severity = <string>
* The level of importance of the event.
* Values must match an integer specified in the default version of
  itsi_notable_event_severity.conf (or the local version if you created one).
* Optional.
* Default: 1 (Info)

action.itsi_event_generator.param.itsi_instruction = <string>
* Instructions for how to address the notable event.
* Must use tokens such as %fieldname% to map the field name from an external event.
  Static instructions are not supported.
* You can use an aggregation policy to aggregate individual instructions into an episode.
  By default, episodes display the instructions for the first event in an episode.
* Optional.

action.itsi_event_generator.param.drilldown_search_title = <string>
* You can drill down to a specific Splunk search from an event or episode.
* The name of the drilldown search link.
* Optional.

action.itsi_event_generator.param.drilldown_search_search = <string>
* The drilldown search string.
* Optional.

action.itsi_event_generator.param.drilldown_search_latest_offset = <seconds>
* Defines how far ahead from the time of the event, in seconds,
  to look for related events.
* This offset is added to the event time.
* Default: 300 (Next 5 minutes)

action.itsi_event_generator.param.drilldown_search_earliest_offset = <string>
* Defines how far back from the time of the event, in seconds,
  to start looking for related events.
* This offset is subtracted from the event time.
* Default: -300 (Last 5 minutes)

action.itsi_event_generator.param.drilldown_title = <string>
* You can drill down to a specific website from an event or episode.
* The name of the drilldown website link.
* Optional.

action.itsi_event_generator.param.drilldown_uri = <string>
* The URI of the website you drill down to.
* Optional.
```

```
param.event_identifier_fields = <comma-separated list>
* A list of fields that are used to identify event duplication.
* Default: source

action.itsi_event_generator.param.service_ids = <comma-separated list>
* A list of service IDs representing one or more ITSI services to
  which this correlation search applies.
* Optional.

action.itsi_event_generator.param.entity_lookup_field = <string>
* The field in the data retrieved by the correlation search that
  is used to look up corresponding entities. For example, host.
* Optional.

action.itsi_event_generator.param.search_type = <string>
* The search type.
* Optional.
* Default: custom

action.itsi_event_generator.param.meta_data = <string>
* The search type of any stored metadata.
* Optional.

action.itsi_event_generator.param.is_ad_at = <boolean>
* Whether this correlation is created by enabling adaptive
  thresholding or anomaly detection (AT/AD) for KPIs or services.
* Optional.
* If "1", the correlation is created by AT/AD.
* If "0", the correlation is not created by AT/AD.
* Default: 0

action.itsi_event_generator.param.ad_at_kpi_ids = <comma-separated list>
* A list of KPIs where AT/AD is enabled.
* Optional.

action.itsi_event_generator.param.editor = <string>
* The type of editor used to create the correlation search.
* Can be either "advance_correlation_builder_editor", which is the correlation
  search editor in ITSI, or "multi_kpi_alert_editor", which is the multi-KPI
  alert builder.
* Default: advance_correlation_builder_editor
```

savedsearches.conf.example

```
# This is an example savedsearches.conf. Use this file to configure
# saved searches.
#
# To use one or more of these configurations, copy the configuration block
# into savedsearches.conf in $SPLUNK_HOME/etc/apps/SA-ITOA/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/ITSI/latest/Configure/ListofITSIconfigurationfiles

[Test ITSI Reporting Search]

cron_schedule                       = */5 * * * *
```

```
disabled                            = False
dispatch.earliest_time              = -5m
dispatch.latest_time                = now
enableSched                         = True
search                              = | stats count | eval demo="Demo Search" | fields - count

action.itsi_event_generator = 1

action.itsi_event_generator.param.title = "Host $result.host$ is down"

action.itsi_event_generator.param.description = Test if host $result.host$ is down or not

action.itsi_event_generator.param.owner = admin

action.itsi_event_generator.param.status = 1

action.itsi_event_generator.param.severity = 2

action.itsi_event_generator.param.drilldown_search_title = Raw search of seeing $result.host$ events

action.itsi_event_generator.param.drilldown_search_search= index=_internal host="$result.host$"

action.itsi_event_generator.param.drilldown_search_latest_offset = 30

action.itsi_event_generator.param.drilldown_search_earliest_offset = -30

action.itsi_event_generator.param.drilldown_title = Go to deep dive "$result.sourcetype$"

action.itsi_event_generator.param.drilldown_uri = "/en-US/app/itsi/search/"

[Test ITSI Notable Event Search]

cron_schedule                       = */5 * * * *
disabled                            = False
dispatch.earliest_time              = -5m
dispatch.latest_time                = now
enableSched                         = True
search                              = index=_internal | head 4

alert.digest_mode          = 0

action.itsi_event_generator = 1

action.itsi_event_generator.param.title = "Host $result.host$ is down"

action.itsi_event_generator.param.description = Test if host $result.host$ is down or not

action.itsi_event_generator.param.owner = admin

action.itsi_event_generator.param.status = 1

action.itsi_event_generator.param.severity = 2

action.itsi_event_generator.param.drilldown_search_title = Raw search of seeing $result.host$ events

action.itsi_event_generator.param.drilldown_search_search= index=_internal host=$result.host$

action.itsi_event_generator.param.drilldown_search_latest_offset = 30

action.itsi_event_generator.param.drilldown_search_earliest_offset = -30

action.itsi_event_generator.param.drilldown_title = Go to deep dive "$result.sourcetype$"
action.itsi_event_generator.param.drilldown_uri = "/en-US/app/itsi/search/"
```

searchbnf.conf

The following are the spec and example files for `searchbnf.conf`.

searchbnf.conf.spec

```
#   Version 8.1.0
#
#
# This file contain descriptions of stanzas and attribute/value pairs for
# configuring search-assistant via searchbnf.conf
#
# There is a searchbnf.conf in $SPLUNK_HOME/etc/system/default/.  It should
# not be modified.  If your application has its own custom python search
# commands, your application can include its own searchbnf.conf to describe
# the commands to the search-assistant.
#
# 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.
```

[<search-commandname>-command]

```
* This stanza enables properties for a given <search-command>.
* A searchbnf.conf file can contain multiple stanzas for any number of
  commands.  * Follow this stanza name with any number of the following
  attribute/value pairs.
* If you do not set an attribute for a given <spec>, the default is used.
  The default values are empty.
* An example stanza name might be "geocode-command", for a "geocode"
  command.
* Search command stanzas can refer to definitions defined in others stanzas,
  and they do not require "-command", appended to them.  For example:
```

[geocode-command]

```
  syntax = geocode <geocode-option>*
  ...
```

[geocode-option]

```
  syntax = (maxcount=<int>) | (maxhops=<int>)
  ...


#*******************************************************************************
# The possible attributes/value pairs for searchbnf.conf
#*******************************************************************************


syntax = <string>
* Describes the syntax of the search command.  See the head of
  searchbnf.conf for details.
* Required.

simplesyntax = <string>

* Optional simpler version of the syntax to make it easier to
  understand at the expense of completeness.  Typically it 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.  In this can, the
  simplesyntax can just pick the one (e.g., "minute").

alias = <commands list>
* Alternative names for the search command.  This further cleans
  up the syntax so the user does not have to know that
  'savedsearch' can also be called by 'macro' or 'savedsplunk'.

description = <string>
* Detailed text description of search command.  Description can continue on
  the next line if the line ends in "\"
* Required.

shortdesc = <string>
* A short description of the search command.  The full DESCRIPTION
  may take up too much screen real-estate for the search assistant.
* Required.

example<index> = <string>
comment<index> = <string>
* 'example' should list out a helpful example of using the search
  command, and 'comment' should describe that example.
* 'example' and 'comment' can be appended with matching indexes to
  allow multiple examples and corresponding comments.
* For example:
    example2 = geocode maxcount=4
    comment2 = run geocode on up to four values
    example3 = geocode maxcount=-1
    comment3 = run geocode on all values

usage = public|private|deprecated
* Determines if a command is public, private, depreciated.  The
  search assistant only operates on public commands.
* Required.

tags = <tags list>
```

```
* List of tags that describe this search command.  Used to find
  commands when the use enters a synonym (e.g. "graph" -> "chart")

related = <commands list>
* List of related commands to help user when using one command to
  learn about others.


#******************************************************************************
# Optional attributes primarily used internally at Splunk
#******************************************************************************

appears-in = <string>
category = <string>
maintainer = <string>
note = <string>
optout-in = <string>
supports-multivalue = <string>
```

searchbnf.conf.example

```
#   Version 8.1.0
#
# 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)
```

service_analyzer_settings.conf

The following are the spec and example files for `service_analyzer_settings.conf`.

service_analyzer_settings.conf.spec

```
# This file contains a setting for determining whether or not to
# always display the First Time Run modal in the Service Analyzer.
#
# To set custom configurations, place a service_analyzer_settings.conf in
# $SPLUNK_HOME/etc/apps/itsi/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
```

[settings]

```
ftr_override = [0|1]
* Whether or not to always display the First Time Run (FTR)
  modal in the Service Analyzer.
* If "1", every time you navigate to the Service Analyzer, the First
  Time Run modal is displayed.
* If "0", the behavior defaults to showing the FTR modal only when
  services are not present.
* Default: 0 (false)

show_cycles_warning = [0|1]
* Whether to display a warning banner on the Service Analyzer
  when there are one or more cyclic dependencies in the service topology.
* After the warning is ignored in the UI, this flag is set to "0" and the
  warning is never shown again unless manually changed.
* If "1", every time a cyclic dependency exists, a warning banner
  appears on the Service Analyzer.
* If "0", the banner never appears on the Service Analyzer.
* Default: 1 (true)
```

service_analyzer_settings.conf.example

```
No example
```

threshold_labels.conf

The following are the spec and example files for `threshold_labels.conf`.

threshold_labels.conf.spec

```
# Copyright (C) 2005-2020 Splunk Inc. All Rights Reserved.
#
# This file contains all possible attribute/value pairs for configuring settings
# for severity-level thresholds. Use this file to configure
# threshold names and color mappings.
#
# To map threshold names and colors, place a threshold_label.conf in
# $SPLUNK_HOME/etc/apps/itsi/local/. For examples, see threshold_label.conf.example.
#
# To learn more about configuration files (including precedence) see the documentation
# located at http://www.splunk.com/base/Documentation/latest/Admin/Aboutconfigurationfiles
```

```
#
# CAUTION: You can drastically affect your Splunk installation by changing any settings in
# this file other than the colors. Consult technical support (http://www.splunk.com/page/submit_issue)
# if you are not sure how to configure this file.
```

[<name>]

```
color = <string>
* A valid color code.
* Required.

lightcolor = <string>
* A valid color code to display for Episode Review "prominent mode".
* When you view Episode Review in prominent mode, the entire row is colored
  rather than just the colored band on the side.
* Required.

threshold_level = <integer>
* A threshold level that is used to create an ordered list of the labels.
* For example, if you set the 'Normal' threshold level to "1", it appears
  first when the levels are listed in the UI.
* Optional.

health_weight = <integer>
* The weight or importance of this status.
* This value should be between 0 and 1.
* In general, regular levels like Normal and Critical have a weight of "1", while
  less important levels like Maintenance and Info have a weight of "0".
* Required.

health_min = <integer>
* The minimum threshold value.
* This value must be a number between 0 and 100. 0 and 100 are inclusive but
  the minimum threshold value is exclusive.
* Required.

health_max = <integer>
* Themaximum threshold value.
* This value must be a number between 0 and 100. 0 and 100 are inclusive but
  the maximum threshold value is exclusive.
* Required.

score_contribution = <integer>
* The number, traditionally from 0 to 100, that this particular level will
  contribute towards health score calculations.
* Required.
```

threshold_labels.conf.example

```
# Copyright (C) 2005-2020 Splunk Inc. All Rights Reserved.
# This is an example threshold_labels.conf. Use this file to
# configure settings for severity-level thresholds.
#
# To use one or more of these configurations, copy the color code
# into threshold_labels.conf in $SPLUNK_HOME/etc/apps/itsi/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/ITSI/latest/Configure/ListofITSIconfigurationfiles
#
# This example alert includes showing raw events at selected time buckets,
# showing raw events from a displayed time range, showing KPI events for
# a host, and showing all events for a host.
#
# This file contains examples of brighter severity colors, with "Normal" severity
# being replaced with "Low" severity.

[info]
color = #6AB7C7
threshold_level = 1

[low]
color = #65A637
threshold_level = 2

[medium]
color =  #FAC51C
threshold_level = 3

[high]
color = #F7902B
threshold_level = 4

[critical]
color = #D85D3C
threshold_level = 5
```

threshold_periods.conf

The following are the spec and example files for `threshold_periods.conf`.

threshold_periods.conf.spec

```
# threshold_periods.conf is DEPRECATED and should not be edited.
```

[<threshold-period-number>]

```
past = <value>
* Label for how far in the past.

description=<value>
* The description.

relative=<value>
* Relative time range.
```

threshold_periods.conf.example

```
No example
```

transforms.conf

The following are the spec and example files for `transforms.conf`.

transforms.conf.spec

```
#   Version 8.1.0
#
# 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.
* 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.
* Optional. When you set INGEST_EVAL, this setting overrides all of the other
  index-time settings (such as REGEX, DEST_KEY, etc) and declares the
  index-time extraction to be evaluator-based.
* 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.
* 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
```

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>/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-bounded, 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>
* NOTE: This attribute is not valid for KV Store-based lookups.
* 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.
* For 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 (or ../etc/searchscripts).
* 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 bounded) 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 'replicationWhitelist' 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
```

transforms.conf.example

```
#   Version 8.1.0
#
# 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
```

visualizations.conf

The following are the spec and example files for `visualizations.conf`.

visualizations.conf.spec

```
#   Version 8.1.0
#
# 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 8.1.0
#
# 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 = <IP address:port>
* The IP address and host port of the splunkd process.
* Don't include "http[s]://" when specifying this setting. Only
  include the IP address and port.
* Default: 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'.
* No default.

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.

requireClientCert = <boolean>
* Requires that any HTTPS client that connects to the Splunk Web HTTPS
  server has a certificate that was signed by the CA cert installed
  on this server.
* If "true", a client can connect ONLY if a certificate created by our
  certificate authority was used on that client.
* If "true", it is mandatory to configure splunkd with same root CA in server.conf.
  This is needed for internal communication between splunkd and splunkweb.
* 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>
flash_minor_version = <integer>
flash_revision_version = <integer>
* 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 the static JavaScript files for modules are consolidated and minified.
* Setting this to "true" improves client-side performance by reducing the number of HTTP
  requests and the size of HTTP responses.

minify_css = <boolean>
* Indicates whether the static CSS files for modules are consolidated and
  minified
* Setting this to "true" improves client-side performance by reducing the number of HTTP
  requests and the size of HTTP responses.
* Due to browser limitations, disabling this when using Internet Explorer
  version 9 and earlier might result in display problems.

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 = <string> [, <string>]
* A comma-separated 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

splunk_dashboard_app_name = <string>
* Please do not change.
* Set the name for the Splunk Dashboard App.
* Default: splunk-dashboard-app

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

#
# 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

testing_endpoint = <relative_uri_path>
* The root URI path on which to serve Splunk Web unit and
  integration testing resources.
* NOTE: This is a development only setting, do not use in normal operations.
* Default: /testing

testing_dir = <relative_file_path>
* The path relative to $SPLUNK_HOME that contains the testing
  files to be served at endpoint defined by 'testing_endpoint'.
* NOTE: This is a development only setting, do not use in normal operations.
* Default: share/splunk/testing
```

```
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: 0 (unlimited size).

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: 0 (unlimited file size).

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_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

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

enableSearchJobXslt = <boolean>
* 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: true
```

```
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".

loginPasswordHint = <default_password_hint>
* The text to display the password hint at first time login on the login page.
* Text only.
* Default: "changeme"

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

# 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.

#
# 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
```

web.conf.example

```
#   Version 8.1.0
#
# 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

# First party apps:
splunk_dashboard_app_name = splunk-dashboard-app

# 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
```