

Splunk® Enterprise 6.3.3

Alerting Manual

Generated: 2/09/2016 1:52 pm

Table of Contents

Alerting overview.....	1
Getting started with alerts.....	1
Choose an alert type.....	5
Alert types and scenarios.....	5
Create alerts.....	7
Create scheduled alerts.....	7
Create per-result alerts.....	13
Create rolling-window alerts.....	14
Manage alert timing and frequency.....	17
Throttle alerts and related searches.....	17
Configure alert actions.....	19
Set up alert actions.....	19
Email notification action.....	19
Use a webhook alert action.....	26
List instances of triggered alerts.....	29
Run a script alert action.....	29
Custom alert actions.....	31
Using custom alert actions.....	31
Manage alert and alert action permissions.....	32
Alert permissions.....	32
Alert action permissions.....	33
View and update alerts.....	34
Update and expand alert functionality.....	34
Specify alert fields.....	34
Alerts page.....	35
Alert details page.....	36
Using the alert actions manager.....	37
Triggered alerts.....	38
Enable summary indexing.....	40
Update alerts from Settings.....	40

Table of Contents

Alert examples.....	44
Alert examples.....	44
Manual alert configuration with .conf files.....	50
Configure alerts in savedsearches.conf.....	50
Send SNMP traps to other systems.....	56
Configure a script for an alert action.....	59

Alerting overview

Getting started with alerts

What is an alert?

If you want to receive notifications about certain events, you can use alerts. When you set up an alert, search results trigger an alert action if they match the alert's conditions.

Alert basics

To get started with an alert, there are a few things to consider.

- **Conditions:** What do you want to know about?
You can start with a search for the events you want to track. As an example, if you have an online store you can track when customers purchase your newest product. You can use an alert whose conditions are website purchase events that also involve this product.
- **Type and Frequency:** How often do you want to know about the event?
You can receive a notification about every customer purchase of a new product as it occurs. Or, you can get a notification on a weekly basis. You can choose continuous per-result, rolling, or scheduled alerts, and adjust their frequency.
- **Alert Action:** What should happen when an alert is triggered?
Once you set up an alert, when customer purchases of the new product show up in search results, they match the alert's conditions. Matching results trigger an alert action according to the frequency you choose. There are several options for alert actions. For example, you can receive an email or update a web resource in response to the triggered alert.

About alert types

There are a few alert types that you can use. Each type works differently with a search to trigger alert actions. You can choose an alert type depending on what event you are tracking and when you want to know about it.

Here is a quick reference guide to alert types and behavior:

Alert type	How it works with searches	Triggering this alert
Per-result alert	Based on a continuous real-time search.	This basic alert triggers any time its search returns a result.
Scheduled alert	Runs a search according to a schedule that you specify when creating the alert.	You can specify which search results trigger the alert.
Rolling-window alert	Based on a continuous real-time search.	You can specify the time window and the conditions that, together, trigger the alert.

To learn about choosing an alert type for different scenarios, see [Alert types and scenarios](#).

For more information on setting up specific alerts, check out resources on [creating per-result alerts](#), [scheduled alerts](#), and [rolling-window alerts](#) in this manual.

You can also check out [Alert examples](#) to get an idea of how each alert type can work.

Choosing an alert type

You can consider using different alerts for different scenarios. Depending on how you want to search for results and set up an alert, you can opt for a per-result, scheduled, or rolling-window alert.

To see some example scenarios and learn about choosing an alert type, see [Alert types and scenarios](#).

Managing alert frequency

You can throttle an alert if you want to change how often it runs an alert action. Throttling an alert does not change how often search results meet the alert conditions. Instead, it changes how often search results matching the alert conditions trigger an alert action.

To learn about changing alert frequency, look at [Throttle Alerts and Related Searches](#) in this manual.

Using alert actions

When search results match an alert's conditions, they trigger the alert action. What happens next?

There are many options for configuring alert actions. For example, you can opt for an email based on the search results. If you want to see updates in a chat room, blog, or other web resource, you can use a [webhook alert action](#).

To learn about setting up different alert actions, see [Set up alert actions](#) in this manual.

Alert and alert action permissions

Alerts and alert actions are knowledge objects with defined permissions. User roles and capabilities determine alert and alert action permissions.

By default, only users with the Admin or Power roles can:

- Create alerts.
- Run real-time searches.
- Schedule searches.
- Save searches.
- Share alerts.

To learn more about configuring alert permissions, see [Alert Permissions](#).

To learn more about configuring alert action permissions, check out [Alert Action Permissions](#) and [Using the Alert Action Manager](#).

To learn more about permissions for knowledge objects, see Manage knowledge object permissions in the *Knowledge Manager* manual.

Scheduled reports and scheduled alerts are not the same

A scheduled report is similar to a scheduled or rolling-window alert in some ways. You can schedule a report and set up an action to run each time the scheduled report runs.

Scheduled reports are different from alerts, however, because a scheduled report's action will run every time the report is run. The report action does not depend on trigger conditions like an alert action does.

As an example, you can monitor guest check-ins at a hotel using an hourly search. Here are the differences between a scheduled report and an alert with email notification actions.

- **Scheduled report:** runs its action and sends an email every time the report completes, even if there are no search results showing check-ins. In this case, you get an email notification every hour.
- **Alert:** only runs alert action when it is triggered by search results showing one or more check-in events. In this case, you only get an email notification if results trigger the alert action.

For more information about scheduled reports, see Schedule reports in the *Reporting Manual*.

Choose an alert type

Alert types and scenarios

There are a few alert types that you can use. Each type works differently with a search to trigger alert actions. You can choose an alert type depending on what event you are tracking and when you want to know about it. You can also throttle an alert if you want to change its frequency.

Here are some scenarios for using each type of alert. To learn how to throttle an alert, see [Throttle alerts and related searches](#).

Per result alert

Use a per result alert to notify when a real-time search returns a result that matches a condition. Typically, you specify a throttle condition so that the alert triggers only once for a specified time period.

Per result examples include the following:

- Trigger an alert for every failed login attempt.
- Trigger an alert when a specific type of error occurs on any host.
You can choose field values that suppress hosts for which you do not want an alert notification.
- Trigger an alert when a CPU on a host sustains 100% utilization for an extended period of time.

Caution: Be careful using a per result alert in a high availability deployment. If a peer is not available, a real-time search does not warn that the search might be incomplete. Use a scheduled alert for this scenario.

Scheduled alert

Use a scheduled alert to notify when a scheduled search returns results that meet a specific condition. A scheduled alert is useful when an immediate reaction to the alert is not a priority. Scheduled alert examples include:

- Trigger an alert that runs daily, notifying when the number of items sold that day is less than 500.
- Trigger an alert that runs hourly, notifying when the number of 404 errors in any hour exceeds 100.

Rolling-window alert

Use a rolling window alert to monitor the results of a real-time search within a specified time interval. For example, monitor the results every 10 minutes or every four hours. Rolling-window alert examples include:

- Trigger an alert when a user has three consecutive failed logins within a 10 minute period.
You can set a throttle condition to suppress an alert to once an hour from any user.
- Trigger an alert when a host is unable to complete an hourly file transfer to another host.
Set a throttle condition so the alert fires only once every hour for any specific host.

Caution: Be careful using a real-time search in a high availability deployment. If a peer is not available, a real-time search does not warn that the search might be incomplete. Use a scheduled alert for this scenario.

Create alerts

Create scheduled alerts

A scheduled alert evaluates the results of a historical search that runs over a specified time range on a regular schedule. The alert fires when it encounters the trigger condition.

For example, you can create a scheduled alert to monitor online sales. The search runs daily at midnight and triggers when the sum of the sales of a specific item is below 500 for the previous day. When the alert triggers, it sends an email to the appropriate administrators monitoring sales.

1. From the Search Page, create the following search. Select **Last 24 Hours** for the time range:

```
index=_internal (log_level=ERROR OR log_level=WARN* OR  
log_level=FATAL OR log_level=CRITICAL) | stats count as  
log_events
```

2. Select **Save As > Alert**

The **Save As Alert** dialog box opens.

3. Specify **Settings**:

- ◆ **Title**: Server Errors Last 24 hours
- ◆ **Alert Type**: Scheduled
- ◆ **Time Range**: Run Every Day
- ◆ **Schedule At**: 0:00
- ◆ **Trigger Condition**: Number of Results
- ◆ **Trigger if number of results**: is Greater than 5

4. Specify **Trigger Conditions**:

- ◆ **Trigger alert when**: Number of Results is Greater than 5
- ◆ **Trigger it**: Once

5. Specify **Trigger Actions**:

- ◆ **Add Actions**: List in Triggered Alerts

See [Set up alert actions](#) for information on other actions.

6. Click **Save**.

Use cron notation for scheduled alerts

When scheduling an alert, you can use cron notation for customized schedules. When specifying a cron schedule, only five cron parameters are available, not six. The sixth parameter for year, common in other forms of cron notation, is not available.

The following cron parameters:

* * * * *

correspond to:

minute hour day month day-of-week

Following are some cron examples:

* / 5 * * * *	Every 5 minutes.
* / 30 * * * *	Every 30 minutes.
0 * / 12 * * *	Every 12 hours, on the hour.
* / 20 * * * 1-5	Every 20 minutes, Monday through Friday.
0 9 1-7 * 1	First Monday of each month, at 9am.

When you select **Run on Cron Schedule** for the time range of a scheduled alert, enter the earliest and latest parameters for a search. What you enter overrides the time range you set when you first ran the search.

To avoid overlaps or gaps, the execution schedule should match the search time range. For example, to run a search every 20 minutes the search's time range should also be 20 minutes (-20m).

The screenshot shows a 'Save As Alert' dialog box with the following fields and values:

- Title: Server Errors Last 24 Hours
- Description: optional
- Alert type: Scheduled (selected), Real Time
- Time Range: Run on Cron Schedule (selected)
- Earliest: -20m (with a date/time picker showing 9/13/14 9:24:27.000 AM and a link to 'Learn More')
- Latest: now (with a date/time picker showing 9/13/14 9:44:49.000 AM and a link to 'Learn More')
- Cron Expression: */20 * * * * (with a link to 'Learn More')
- Trigger condition: Number of Results (selected)
- Trigger if number of results: is Greater than 5

Buttons: Cancel, Next

Manage the priority of concurrently scheduled searches

Depending on your Splunk Enterprise deployment, you might be able to run only one scheduled search at a time. In this case, when you schedule multiple searches to run at approximately the same time, the search scheduler ensures that all scheduled searches run consecutively for the period of time over which they gather data.

However, you might have cases where you need certain searches to run ahead of others. This is to ensure that the searches obtain current data or to ensure that there are no gaps in data collection.

You can configure the priority of scheduled searches in the `savedsearches.conf` configuration file. See "Configure the priority of scheduled reports" in the *Reporting Manual*.

Best practices for scheduled alerts

This section discusses some best practices for scheduled alerts.

Coordinate an alert's schedule with the search time range

Coordinating the alert's schedule with the search time range prevents situations where event data is evaluated twice by the search. This can happen if the search time range exceeds the search schedule, resulting in overlapping event data sets.

In cases where the search time range is shorter than the time range for the scheduled alert, an event might never be evaluated.

Schedule alerts with at least 60 seconds of delay

This practice is important in distributed search deployments where event data might not reach the indexer precisely at the moment when it is generated. A delay ensures that you are counting all events, not just the events that were quickest to get indexed.

Best practices example

This example shows how to configure an alert that builds 30 minutes of delay into the alert schedule. Both the search time range and the alert schedule span one hour, so there are no event data overlaps or gaps.

The alert runs every hour at the half hour. It collects an hour's worth of event data, beginning an hour and a half before the search runs. When the scheduled search kicks off at a designated time, such as 3:30 pm, it collects the event data that was indexed from 2:00 pm to 3:00 pm.

1. From the Search Page, create a search and select **Save As > Alert**.
2. In the **Save As Alert** dialog, specify the following to schedule the alert:

- ◆ **Title:** Alert Example (30 Minute Delay)
- ◆ **Alert Type:** Scheduled
- ◆ **Time Range:** Run on Cron Schedule
- ◆ **Earliest:** -90m
- ◆ **Latest:** -30m

Earliest and Latest values set the time that the search covers to a period that begins 90 minutes before the search launch time, ending 30 minutes before the search launch time.

- ◆ **Cron Expression:** 30 * * * *

The alert runs every hour on the half hour

The screenshot shows the 'Save As Alert' dialog box with the following configuration:

- Title:** Alert Example (30 Minute Delay)
- Description:** optional
- Alert type:** Scheduled (selected), Real Time
- Time Range:** Run on Cron Schedule
- Earliest:** -90m (with example: 5/14/14 6:40:39.000 AM)
- Latest:** -30m (with example: 5/14/14 7:41:13.000 AM)
- Cron Expression:** 30 * * * * (with example: 00 18 *** (every day at 6PM))
- Trigger condition:** Number of Results
- Trigger if number of results:** is Greater than 5

Buttons: Cancel, Next

3. Continue defining actions for the alert.

Set up triggering conditions for a scheduled alert

Trigger conditions apply to two types of conditional alerts:

- Basic conditional alert
- Advanced conditional alert

Set the triggering conditions when you set values for the **Trigger condition** field in the **Save As Alert** dialog box, as described in the following subtopics.

Basic conditional alert

A **basic conditional alert** triggers when the number of results of a scheduled search meet, exceed, or are less than a specified numerical value. When you create the alert, you can specify the following conditions:

- Number of results
- Number of hosts
- Number of sources

The alert triggers when the number of hosts in the results rises by a count of more than 12.

1. From the Search Page, create a search and select **Save As > Alert**.
2. In the **Save As Alert** dialog box, specify the following fields to schedule the alert:

- ◆ **Title:** Alert Example (Basic Conditional)
- ◆ **Alert Type:** Scheduled
You can also select Real Time for a basic conditional search.
- ◆ **Time Range and Schedule:** Select any time range and schedule.
- ◆ **Trigger Condition:** Number of Hosts
You can also select Number of Results or Number of Sources
- ◆ **Trigger if number of results:** Select a comparison operator and trigger value.

Save As Alert

Title: Alert Example (Basic Conditional)

Description: optional

Alert type: Scheduled Real Time

Time Range: Run every week

Schedule: On Monday at 6:00

Trigger condition: Number of Hosts

Trigger if number of results: Rises by 12

Cancel Next

3. Continue defining actions for the alert.

Basic conditional alert for rolling-window alerts

The behavior for basic conditional alerts differs slightly for a rolling-window alert. The alert triggers when the set condition occurs within the rolling time window of the search.

For example, a rolling-window alert that triggers when a time window of 60 seconds has five or more results. If the real-time search returns one result and then four more results five minutes later, the alert does not trigger. The alert does trigger if the search returns five results within a single 60-second span.

Advanced conditional alert

An **advanced conditional alert** uses a secondary, custom conditional search to evaluate the results of a scheduled or real-time search. The alert triggers when the custom search returns any number of results. If the alerting conditions are not met, then the custom conditional search returns zero results.

A secondary conditional search can help reduce the incidence of false positive alerts.

In the following example, the alert triggers when there are 10 or more log level events that are not INFO. When the alert triggers, it sends an email with the results of the search. The search results detail the count for each log level.

1. From the Search Page, create the following search. Specify **Last 7 days** for the time period.

```
index=_internal (log_level=ERROR OR log_level=FATAL OR  
log_level=CRITICAL) | stats count by log_level
```

2. Select **Save As > Alert**.
3. In the **Save As Alert** dialog box, specify the following fields to schedule the alert:

- ◆ **Title:** Alert Example (Advanced Conditional)
- ◆ **Alert Type:** Scheduled
You can also select Real Time for an advanced conditional search.
- ◆ **Time Range and Schedule:** Select any time range and schedule.
- ◆ **Trigger Condition:** Custom
- ◆ **Custom condition:** search count > 10

Save As Alert

Title: Alert Example (Advanced Conditional)

Description: optional

Alert type: Scheduled Real Time

Time Range: Run every week

Schedule: On Monday at 6:00

Trigger condition: Custom

Custom Condition: search count > 10 e.g. "search count > 10". Evaluated against the results of the base search.

Cancel Next

4. Define an action that sends an email that includes the results of the search.
When you configure a **Send Email** action that includes search results, the email contains the results of the original base search. It does not include the results of the custom search.

It might appear that you can get the same results if you specify instead, the following search for the base search of a basic conditional search:

```
log_level=ERROR OR log_level=FATAL OR log_level=CRITICAL) | stats count  
by log_level | search count > 10
```

However, a basic conditional alert based on this search provides different results. The search results contain only log level values that are greater than 10. The results from the advanced conditional search details the count for all log levels, but triggers only when the log levels are greater than 10.

Advanced conditional alert for rolling-window alerts

The behavior for advanced conditional alerts differs slightly for a rolling-window alert, which runs in real-time. For a rolling-window alert, the alert triggers when the set condition occurs within the rolling time window of the search.

For the previous example, you can design a rolling-window alert with the same base search and get similar results with the custom condition search. Set the rolling window to a 10 minutes time span. When the real-time search returns 10 log level entries within the 10 minute time span, the alert triggers.

For more examples of scheduled alerts, see ["Alert examples,"](#) in this manual.

Create per-result alerts

The per-result alert is the most basic type of alert. It runs in real-time over an "all-time" time span. The alert triggers whenever the search returns a result.

You can create a search to retrieve events from an index. You can also use transforming commands to return results based on processing the retrieved events. A per-result alert triggers in both cases, when the search returns an event or when a transforming command returns results.

Create a per-result alert

The following procedure shows how to create a per-result alert.

1. From the Search Page, enter the following search:

```
index=_internal (log_level=ERROR OR log_level=WARN* OR
log_level=FATAL OR log_level=CRITICAL) | stats count as
log_events
```

2. Select **Save As > Alert**
3. In the **Save As Alert** dialog box, enter a **Title** for the alert.
4. For **Alert Type**, select **Real Time**.
A per-result alert is always a real-time alert type.
5. For trigger condition, select **Per-Result**.
6. Select the actions you want to enable.
For this example, select **List in Triggered Alert**.

See [Set up alert actions](#) for information on other actions.

7. Click **Save**.

Create rolling-window alerts

Use a rolling-window alert to monitor and evaluate events in real time within a rolling window. The alert triggers only when it meets the trigger condition within a specified time period.

The rolling-window alert type is in some ways a hybrid of a per-result alert and a scheduled alert. A rolling-window alert and a per result alert both run in real-time. But unlike the per result alert, a rolling-window alert does not trigger each time the search returns a result. A rolling-window alert fires only when it meets specified trigger conditions within the specified time window. This makes the alert similar to a scheduled alert.

1. From the Search Page, create the following search. Select **Last 24 Hours** for the time range:

```
index=_internal (log_level=ERROR OR log_level=WARN* OR
log_level=FATAL OR log_level=CRITICAL) | stats count as
log_events
```

2. Select **Save As > Alert**

3. In the **Save As Alert** dialog box, specify the following:

- ◆ **Title:** Alert Example (Rolling-Window)
- ◆ **Alert Type:** Real Time
- ◆ **Trigger alert when:** Number of Results is Greater than 5
- ◆ **in:** 30 minutes

The screenshot shows the 'Save As Alert' dialog box. The 'Title' field is set to 'Alert Example (Rolling-Window)'. The 'Description' field is set to 'optional'. The 'Alert type' is set to 'Real Time'. The 'Trigger condition' is set to 'Number of Results'. The 'Number of results is' is set to 'Greater than' and '10'. The 'in' field is set to '30' and 'minute(s)'. The 'Next' button is highlighted in green.

4. Continue defining actions for the alert.

See [Set up alert actions](#).

Set the width of the rolling window

When you create a rolling-window alert, you specify a time span for a real-time search window. Real-time search windows can be any number of minutes, hours, or days. The alert monitors events as they pass through the window in real-time.

For example, you can create an alert that triggers when a login for a user fails four times in a 10 minute period. When the alert runs, various login failure events pass through this window. The alert triggers only when four login failures for the same user occur within the span of the 10 minute window.

This example might appear to fail in the following scenario. A user experiences three login failures in quick succession. After 11 minutes pass, the user has another login failure. The alert does not trigger because the first three failures and the fourth failure are in different time windows.

Set up triggering conditions for a rolling-window alert

Trigger conditions apply to two types of rolling-window alerts:

- Basic conditional alert
- Advanced conditional alert

You set the triggering conditions when you set values for the **Trigger condition** field in the **Save As Alert** dialog, as described in the following subtopics.

Basic conditional alert

A **basic conditional alert** triggers when the number of results from a search, within a specified time window, meet, exceed, or are less than a specified numerical value. When you create the alert, you can specify the following conditions:

- Number of results
- Number of hosts
- Number of sources

You create a basic conditional alert for a rolling-window similarly to how you create one for a scheduled alert. See [Set up triggering conditions for a scheduled alert](#) for an example.

Advanced conditional alert

An **advanced conditional alert** uses a secondary, custom conditional search to evaluate the results of a scheduled or real-time search. For a rolling-window alert, the alert triggers when the custom search returns any number of results within the specified time window. If the alerting conditions are not met, then the custom conditional search should return zero results.

A secondary conditional search can help reduce the incidence of false positive alerts.

You create an advanced conditional alert for a rolling-window similarly to how you create one for a scheduled alert. See [Set up triggering conditions for a scheduled alert](#) for an example.

Manage alert timing and frequency

Throttle alerts and related searches

Use throttling to limit alert frequency

Use throttling to reduce the frequency at which an alert triggers. An alert can trigger frequently based on similar results that the search returns. The schedule to run an alert can also cause the alert to trigger frequently. To reduce the frequency of the alert firing, configure the following:

- A time period in which to suppress results.
- Field values that the search returns.

For example, you can create an alert that fires when a system error occurs. For this example, assume that when the system error occurs, it occurs 20 or more times each minute. However, you want to send an alert only once every hour. To reduce the frequency of the alert firing, configure throttling for the alert.

1. From the Search Page, enter the following search:

```
index=_internal log_level=ERROR
```

2. Select **Save As > Alert**
3. For **Result Type**, click **Real Time** to configure a per-result alert.
4. Click **Next**.
5. Select the actions you want to enable.
6. Select **Throttle**.
7. Enter **log_level** to suppress the alert for the field `log_level`.
You can configure throttling to suppress on more than one field. Use a comma-delimited list to specify fields for throttling.
8. Enter **1 hour** as the time to suppress triggering for the alert.

Save As Alert

Enable Actions

List in Triggered Alerts ☒ Triggered Alerts is available in the activity menu.

Severity Low ▾

Send Email ☐ Email must be configured in System Settings > Alert Email Settings. [Learn More](#)

Run a Script ☐

Action Options

Throttle ? ☒

Suppress results containing field value log_level

Suppress triggering for 1 hour(s) ▾

Sharing

Permissions Private Shared in App

Cancel Back Save

9. Click **Save**.

You can set up a per-result alert that throttles events that share the same `clientip` and `host` values. For example, a real-time search with a 60 second window triggers an alert every time an event with disk error appears. Ten events with the error message that occurs in the window triggers five disk error alerts, which is ten alerts within one minute. If the alert sends an email notification each time it triggers, you can overwhelm a email Inbox.

You can configure throttling so that when one alert of this type triggers, it suppresses all successive alerts of the same type for the next 10 minutes. After each successive 10 minutes period pass, the alert can trigger again.

Throttle scheduled and real-time searches

If you have scheduled searches that run frequently and you do not want to be notified for each run, set the throttling controls to suppress the alert to a longer time window.

For real-time searches, if you configure an alert so that it fires once for a trigger condition, you do not need to configure throttling. If the alert fires more than once for the trigger condition, consider throttling to suppress results.

When you configure throttling for a real-time search, start with a throttling period that matches the length of the base search's time window. Expand the throttling period if necessary. This prevents multiple notifications for a given event.

Configure alert actions

Set up alert actions

Alert action options

You can enable several alert actions to follow a triggered alert. There are also additional options for working with alerts or alert actions, such as listing triggered alerts, enabling alert summary indexing, and specifying search fields.

To learn about	See
Sending email notifications when alerts are triggered	Email notification action
Using a webhook to display a message in a chat room or update another web resource	Use a webhook alert action
Listing instances of triggered alerts	List instances of triggered alerts
Enabling summary indexing for an alert	Enable summary indexing for an alert
Specifying which search fields to show in an alert	Specify alert fields

Email notification action

You can configure an alert to send an email notification to specified recipients when the alert triggers. You can send the email notification as a multipart MIME message that includes both HTML and text parts. You can also send the notification as plain text.

You configure the email notification action for an alert when you save the alert from the Search page. You can also configure email notification from the Alerts Page and directly from a search command.

Email notification contexts

There are several contexts from which you can send email notifications. The email options available differ, depending on the context.

- **Alert actions**
Send email notifications as an alert action from a search. Specify the notification from the Search Page, a listing in the Alerts Page, or directly from the search command.
- **Scheduled report**
Configure email notifications for a scheduled report either from a listing in the Reports Page or from a report.
- **Scheduled PDF delivery of dashboards**
Configure PDF delivery either from a listing in the Dashboards Page or from a dashboard.

This topic covers alert actions from a search job. See [Schedule reports](#) and [Generate Dashboard PDFs](#) for information on the other contexts for email notification.

Configure email notification for alerts

You can configure email notifications when you save a search as an alert. You can also configure email notifications for when editing an alert's actions. The procedure is the same in both cases.

After running a search, save the search as an alert and configure email notification settings.

1. Run the search.
2. Select **Save As > Alert**.
3. Provide a **Title** and other information about the alert.
4. From the **Add Actions** menu, select **Send email**.

The screenshot shows the 'Save As Alert' dialog box with the following configuration:

- Trigger:** Once (selected), For each result
- Throttle?** ☐
- Trigger Actions:** + Add Actions
- When triggered:** ☒ Send email (Remove button)
- To:** [Empty text field] (Comma separated list of email addresses. Show CC and BCC)
- Priority:** Normal
- Subject:** Splunk Alert: \$name\$ (The email subject and message can include tokens that insert text based on the results of the search. [Learn More](#))
- Message:** The alert condition for '\$name\$' was triggered.
- Include:**
 - ☒ Link to Alert
 - ☒ Link to Results
 - ☐ Search String
 - ☐ Inline Table
 - ☐ Trigger Condition
 - ☐ Attach CSV
 - ☐ Trigger Time
 - ☐ Attach PDF
- Type:** HTML & Plain Text (selected), Plain Text
- Buttons:** Cancel, Save

5. Specify the following:

- ◆ **To, CC, and BCC** email recipients.
Specify a comma-separated list of email recipients.
- ◆ **Priority**
Enforcement of priority depends on your email client.
- ◆ **Subject**
- ◆ **Message**
- ◆ **Include**
You can include the following items:

Information about the search

Link to the alert
Search string
Trigger condition
Trigger time

Information about search results

Link to results
Inline listing of results, as a table, raw events, or CSV file
Results as a PDF attachment
Results as a CSV attachment

- ◆ **Type**
Select **HTML & Plain Text** (multi-MIME message) or **Plain Text**

6. Specify other alert actions.

See [set up alert actions](#) for more information.

7. Click **Save**.

Send email notification from a search command

You can send email notifications directly from the `sendemail` search command.
For example:

```
index=main | head 5 | sendemail to=example@splunk.com  
server=mail.example.com subject="Here is an email notification"  
message="This is an example message" sendresults=true inline=true  
format=raw sendpdf=true
```

See the `sendemail` command listing in the *Search Reference* for details.

Use tokens in email notifications

A token is a type of variable that represents data generated by a search job. Splunk Enterprise provides tokens that you can use to include information generated by a search in the fields of an email:

- To
- Cc
- Bcc
- Subject
- Message
- Footer

Access the value of a token with the following syntax:

`$<token-name>$`

For example, place the following token in the subject field of an email notification to reference the search ID of a search job.

Search results from `$job.sid$`

Tokens available for email notifications

This section lists common tokens you can use in email notifications. There are four categories of tokens that access data generated from a search. The context for using the tokens differ.

Category	Description	Context
Search metadata	Information about the search.	Alert actions from search Scheduled reports Scheduled PDF delivery of dashboards
Search results	Access results of a search	Alert actions from search Scheduled reports
Job information	Data specific to a search job	Alert actions from search Scheduled reports
Server information	Information about the Splunk Enterprise server	Alert actions from search

		Scheduled reports Scheduled PDF delivery of dashboards
--	--	---

In addition to the common tokens listed in this topic, the `savedsearches.conf` and `alert_action.conf` configuration files list attributes whose values are available from tokens. To access these attribute values, place the attribute between the '\$' token delimiters. For example, to access the subject of an email notification, reference the following attribute listed in `savedsearches.conf`:

`$action.email.subject$`

Tokens that access search metadata

Common tokens that access information about a search. These tokens are available from the following contexts:

- Alert actions
- Scheduled reports
- Scheduled PDF delivery of dashboards

Here are some of the common tokens available.

Token	Description
<code>\$action.email.hostname\$</code>	Hostname of the email server.
<code>\$action.email.priority\$</code>	Priority of the search.
<code>\$app\$</code>	Name of the app containing the search.
<code>\$cron_schedule\$</code>	Cron schedule for the app.
<code>\$description\$</code>	Description of the search.
<code>\$name\$</code>	Name of the search.
<code>\$next_scheduled_time\$</code>	The next time the search runs.
<code>\$owner\$</code>	Owner of the search.
<code>\$results_link\$</code>	(Alert actions and scheduled reports only) Link to the search results.
<code>\$search\$</code>	The actual search.
<code>\$trigger_date\$</code>	(Alert actions only) The date that triggers the alert.
<code>\$trigger_time\$</code>	(Alert actions only) The scheduled time the alert runs.

\$type\$	Indicates if the search is from an alert, report, view, or the search command.
\$alert.severity\$	Severity level of the alert.
\$alert.expires\$	Time the alert expires.

Tokens available from results

From results, you use the `result.<fieldname>` token to access the first value of a specified field in search results. This token is available from the following contexts:

- Alert actions
- Scheduled reports

Token	Description
\$result.fieldname\$	Returns the first value for the specified field name from the first result in the search. The field name must be present in the search.

Tokens that access job information

Common tokens that access data specific to a search job, such as the search ID or messages generated by the search job. These tokens are available from the following contexts:

- Alert actions
- Scheduled reports

Token	Description
\$job.earliestTime\$	Initial time a search job starts.
\$job.eventSearch\$	Subset of the search that contains the part of the search before any transforming commands.
\$job.latestTime\$	Latest time recorded for the search job.
\$job.messages\$	List of error and debug messages generated by the search job.
\$job.resultCount\$	Number of results returned by the search job.
\$job.runDuration\$	Time, in seconds, that the search took to complete.
\$job.sid\$	Search ID.
\$job.label\$	Name given to the search job.

Tokens available from server

Common tokens that provide details available from your Splunk Enterprise server. They are available in the following contexts:

- Alert actions
- Scheduled reports
- Scheduled PDF delivery of dashboards

Token	Description
\$server.build\$	Build number of the Splunk Enterprise instance.
\$server.serverName\$	Server name hosting the Splunk Enterprise instance.
\$server.version\$	Version number of the Splunk Enterprise instance.

Deprecated email notification tokens

The following tokens from prior releases of Splunk Enterprise are deprecated.

Token	Description
\$results.count\$	(Deprecated) Use \$job.resultCount\$.
\$results.url\$	(Deprecated) Use \$results_link\$.
\$results.file\$	(Deprecated) No equivalent available.
\$search_id\$	(Deprecated) Use \$job.id\$.

Configure email notification settings

Before you send an email notification for an alert, configure email notification settings.

Prerequisites

- (Optional) Scheduling PDF delivery requires additional configuration of user roles. For more information, see [User role configuration to schedule PDF delivery of dashboards](#).
- (Optional) To learn about using tokens in email configuration fields, see [Use tokens in email notifications](#).

Here are the steps for configuring email notification settings in Splunk Web.

1. Navigate to **Settings > Server settings > Email settings**.
 2. Select **Mail Server Settings**. Enter the following details.
 - ◆ **Mail host**. The default is localhost.
 - ◆ **Email security**.
 - ◆ (Optional) **Username**.
 - ◆ (Optional) **Password**.
 3. Specify **Email Format**. Specify the following details.
 - ◆ **Link hostname**. The host name of the server used for creating URLs for outgoing results.
 - ◆ **Send emails as**. Enter the email address for the sender.
 - ◆ **Email footer**. Footer text for each email. You can use tokens in the email footer.
 4. Specify the following **PDF Report Settings**.
 - ◆ **Report Paper Size**.
 - ◆ **Report Paper Orientation**.
 5. Click **Save**.
-

To learn about configuring email alert notifications using a configuration file, see `alert_actions.conf`.

User role configuration to schedule PDF delivery of dashboards

For a user to schedule PDF delivery of dashboards, the user role must contain the following capabilities:

- `schedule_search`
- `admin_all_objects`

This capability is required only if the mail host requires log-in credentials.

See About defining roles with capabilities.

Use a webhook alert action

What is a webhook?

Webhooks allow you to define custom callbacks on a particular web resource. For instance, you can set up a webhook to make an alert message pop up in a chat room or post a notification on a web page.

About webhook alert actions

You can create a webhook action for instant alert notifications at a particular URL. When an alert is triggered, the webhook will make an HTTP POST request on the URL. The webhook passes JSON formatted information about the alert in the body of the POST request.

A webhook starts with an alert. You can define conditions for triggering the webhook alert action.

As an example, imagine that you have an alert set up to trigger whenever a new customer signs up on your company's website. Let's also imagine that you have a web-based chat client at work where employees can exchange quick updates or ask questions.

A webhook can help you use your chat client as a real time information hub for customer sign-ups. You can set up a webhook with the chat client's URL. Each time the webhook's alert triggers, the webhook makes an HTTP POST request to that URL. The POST request carries a data payload to deliver to the URL.

For a webhook, the POST request's JSON data payload includes:

- Search ID or SID for the saved search that triggered the alert
- Search owner and app
- First result row from the triggering search results

Here is an example of what the JSON information might look like:

```
{
  "result": {
    "sourcetype": "mongod",
    "count": "8"
  },
  "sid": "scheduler_admin_search_w2_at_1427942640_178",
  "results_link": "http://windu.splunk.local:8000/app/search/@go?sid=scheduler_admin_search_w2_at_1427942640_178",
  "search_name": null,
  "owner": "admin",
  "app": "search"
}
```

In this example, the SID is

"scheduler_admin_search_w2_at_1427942640_178". The owner role is "admin", and this alert comes from the Search and Reporting app.

The data payload may contain more information from the alert. You can configure the way your web resource handles the data payload.

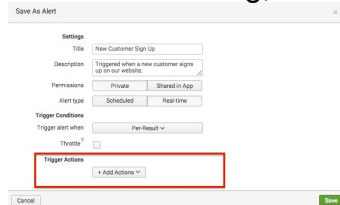
Continuing with our example, your chat client can use the POST request data to show a notification. Using a webhook, you can monitor customer sign-ups in real

time.

Set up a webhook

You can set up a webhook starting when you save a search as an alert.

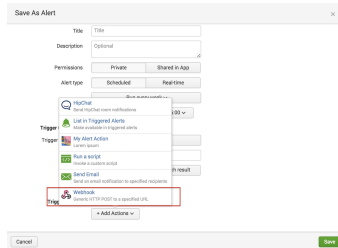
- In the **Save As Alert** dialog, find the **Trigger Actions** menu. Click **+Add**



The screenshot shows the 'Save As Alert' dialog box. The 'Trigger Actions' section is highlighted with a red box, showing a '+ Add Actions' button. The dialog includes fields for Title, Description, Permissions, Alert type, Trigger Conditions, and a checkbox for 'Threats'.

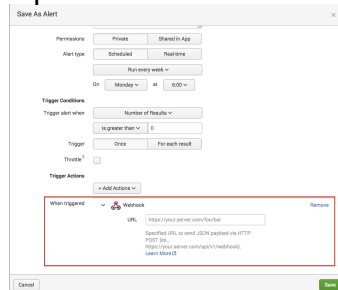
Actions.

- Select **Webhook**.



The screenshot shows the 'Save As Alert' dialog box. The 'Trigger' section is highlighted with a red box, showing a dropdown menu with 'Webhook' selected. The dialog includes fields for Title, Description, Permissions, Alert type, Trigger Conditions, and a checkbox for 'Threats'.

- Input a URL for the webhook.



The screenshot shows the 'Save As Alert' dialog box. The 'Webhook' section is highlighted with a red box, showing a URL input field. The dialog includes fields for Title, Description, Permissions, Alert type, Trigger Conditions, and a checkbox for 'Threats'.

- Click **Save**.

List instances of triggered alerts

Select the **List in Triggered Alerts** action to display a list of instances when the alert triggers.

You can see records of recently triggered alerts from the Triggered Alerts page or from an Alert Details page. The Triggered Alerts page shows all instances of triggered alerts. The Alert Details page shows all instances of triggered alerts for a specific alert. Details of triggered alerts are available for 24 hours or a specified duration.

See "[Review triggered alerts](#)" in this manual.

Give tracked alerts a severity level

When listing a triggered alert, you can specify a **Severity** level. Severity levels are informational only. They let you group and highlight alerts in the Alert Manager according to the severity levels. You decide which level applies to the alert.

You can choose from the following severity levels. The default level is Medium.

- Info
- Low
- Medium
- High
- Critical

Run a script alert action

The run a script alert action is officially deprecated. It has been replaced with custom alert actions as a more scalable and robust framework for integrating custom actions. See [About custom alert actions](#) for information on building customized alert actions that can include scripts.

You can run an alert script when a alert triggers. Select **Run a script** from the **Add Actions** menu. Enter the file name of the script that you want to run.

For example, you can configure an alert to run a script that generates a Simple Network Management Protocol (SNMP) trap notification. The script sends the notification to another system such as a Network Systems Management console.

You can configure a different alert that runs a script that calls an API, which in turn sends the triggering event to another system.

Note: For security reasons, place all alert scripts in either of the following locations:

```
◇ $SPLUNK_HOME/bin/scripts  
◇ $SPLUNK_HOME/etc/<AppName>/bin/scripts
```

For details on alert script configuration in `savedsearches.conf` for a shell script or batch file that you create, see [Configure scripted alerts](#) in this manual.

If you are having trouble with alert scripts, see [Troubleshooting alert scripts](#) on the Splunk Community Wiki.

Custom alert actions

Using custom alert actions

App developers can build custom, user-configurable, alert actions into their apps. Users can find apps with built-in custom alert actions from the alert actions manager page.

To try using a custom alert action, you can use the built-in webhook alert action to send notifications to a web resource, like a chat room or blog. For more information, see [Use a webhook alert action](#).

To learn how to find apps with built-in alert actions, see [Using the alert actions manager](#).

For more information on how to develop and use custom alert actions, see Custom alert actions in *Developing Views and Apps for Splunk Web*.

Manage alert and alert action permissions

Alert permissions

Alerts are knowledge objects with defined permissions. User roles and capabilities determine alert creation, usage, editing, and other permissions.

By default, only users with the Admin or Power roles can do the following.

- Create alerts.
- Run real-time searches.
- Schedule searches.
- Save searches.
- Share alerts.

Authorized users can share an alert with other app users by editing the alert permissions. When sharing an alert with a user without the Admin or Power role, the user needs permission to access the alerting features. For example, a user needs the capability to run a real-time search in order to access a real-time alert.

Admins can configure alert action permissions to change what alert actions are available to users in a particular app. For more information, see [Alert Action Permissions](#).

Sharing an alert

You can configure sharing preferences when creating an alert or edit alert permissions later. Here are the steps for editing alert permissions.

1. Navigate to the **Alerts** page in the **Search and Reporting** app.
2. Find the alert you want to share and select **Edit>Edit Permissions**.
3. Share the alert by configuring which users can access it. Here are the options.

Option	Sharing description
Owner	Makes the alert private to the alert creator.
App	Display the alert for all users of the app.
All apps	Display the alert for all users of this Splunk platform instance.

4. Select read and write permissions for the user roles listed.

- **Read:** Users can see the alert on the **Alerts** page and run the alert in the app.
- **Write:** Users with appropriate permissions can modify, enable, and disable the alert.

Alert action permissions

Depending on your user role, you can configure alert action permissions for available alerts.

For example, an admin can adjust alert actions permissions for the **Search and Reporting** app. The admin can change what alert actions are available to users who create an alert in this app.

To review and change alert actions permissions, use the **Alert actions** manager page. For more information, see [Using the alert actions manager](#).

Alert actions are knowledge objects. To learn more about managing knowledge object permissions, see Manage knowledge object permissions in the *Knowledge Manager* manual.

View and update alerts

Update and expand alert functionality

You can view alert information and configure alerts from the following places in Splunk Web.

Location	Description
Alerts page	Provides a listing of all alerts created within an app. It contains options for editing an alert. Click an alert entry to view the detail page for the alert. See Alerts page .
Alert detail page	Provides links to update an alert. When applicable, lists triggered alerts. See Alert Details page .
Alert Actions Manager	See Alert Actions Manager .
Settings	<p>Alerts are a type of saved search. You can view saved searches, reports, and alerts from the Searches, reports, and alerts view in Settings. From this view, you can do the following:</p> <ul style="list-style-type: none">• Create a new alert.• Create an alert based on an existing search.• Modify an alert.• Enable or disable an alert.• Delete alerts for which you have the appropriate permissions. <p>See Update alerts from Settings</p>

Specify alert fields

Specify fields to show in alerts through search language

The results of an alerting search job (in an alert email, for example) includes all the fields in those results. To include or exclude specific fields from the results, use the `fields` command in the base search for the alert.

- To eliminate a field from the search results, pipe your search to `fields - $FIELDNAME`.
- To add a field to the search results, pipe your search to `fields + $FIELDNAME`.

You can specify multiple fields in one string. The following search generates an alert that excludes `$FIELD1` and `$FIELD2`, but includes `$FIELD3` and `$FIELD4`.

```
yoursearch | fields - $FIELD1,$FIELD2 + $FIELD3,$FIELD4
```

Alerts page

The **Alerts** page lists all alerts for an app. It is available from the top-level navigation menu for an app. From the **Alerts** page you can use the following options.

Option	Description
Select a filtering option for displayed alerts.	<ul style="list-style-type: none"> • All. View all alerts for which you have view permission. • Yours. View alerts that you own. • This App's. View alerts for the current app. Only alerts for which you have permission to view display in the list.
Select any displayed alert	Opens the detail page for an alert. You can review and make additional edits to the alert on the detail page.
Open in Search	View or modify the alert's search in the Search page.
Edit	Opens the detail page for an alert. You can review and make additional edits to the alert on the detail page.

Edit an alert search

1. From the **Alerts page**, locate the alert and click **Open in Search**. The alert search opens in the **Search** page.
2. Edit the search as needed.
3. Run the edited search.
4. Click **Save** to update the alert. If prompted again, click **Save**.

5. Select from the following options.

Option	Description
"View alert"	Opens the alert detail page.
"Continue editing"	Return to the Search page.
"Permissions"	View and edit alert permissions.

Alert details page

Open an alert's detail page to modify the alert search or other settings. The alert details page also displays alert trigger history.

Modify the search for an alert from the Alert detail page

1. From the alert detail page, select **Edit > Open in Search**.
2. In the Search page that opens, modify the search.
3. Run the modified search.
4. Click **Save** to update the alert. Click **Save** in the dialog that appears.
5. Select from the following:

- ◆ **View Alert**

Opens the detail page for the alert.

- ◆ **Continue Editing**

Return to the search page

- ◆ **Permissions**

To view and modify the permissions for the alert.

View and modify alert details

The Alert details page provides a listing of the current settings for an alert. You can view and modify the following details:

- Whether the alert is enabled
- Alert type, Scheduled or Real-time
- Trigger condition
- Actions
- Permissions

View an alert's trigger history

If you specify List in Triggered Alerts as an alert action, the alert detail page lists the trigger history for the alert.

From the trigger history you can view the results that triggered the alert.

You can also view trigger history from the Alerts Manager.

1. From the Splunk Enterprise menu bar, select **Activity > Triggered Alerts**.
2. In the Alert Manager, filter the results according to **App**, **Owner**, **Severity**, and **Alert** name.
3. Take the following actions from the Alert Manager:

- ◇ View the results.
- ◇ Edit the search.
- ◇ Delete a triggered alert listing.

For more information, see [Review triggered alerts](#).

Using the alert actions manager

You can review and configure settings for available alert actions on the alert actions manager page.

Prerequisites

(Optional) Review [Alert action permissions](#).

1. From the top-level navigation bar, select **Settings > Alert actions**.
 2. Depending on your permissions, you can do the following for an alert action.
 - ◆ Enable or disable the alert action
 - ◆ Update permissions
 - ◆ Check usage stats
 - ◆ View log events
 3. (Optional) Click **Browse More** to find apps with built-in custom alert actions.
-

Triggered alerts

Review all recently triggered alerts on the **Triggered Alerts** page. You can also see recent trigger activity for a specific alert on its detail page.

For information on configuring the "Add to Triggered Alerts" action, see [List instances of triggered alerts](#).

Triggered alert listing

Alerts appear on the **Triggered Alerts** page under the following conditions.

- The "Add to Triggered Alerts" action is enabled for the alert.
- The alert triggered recently.
- The alert retention time is not complete.
- The triggered alert listing has not been deleted.

On the **Triggered Alerts** page, details appear in the following categories.

Category	Description
Time	Trigger date and time.
Fired alerts	Triggered alert name(s).
App	Alert app context.
Type	Alert type.
Severity	Assigned alert severity level. Severity levels can help you sort or filter alerts on this page.
Mode	Alert triggering configuration mode. "Per-result" means that the alert triggered because of a single event. "Digest" means that the alert triggered because of a group of events.

Access and update triggered alerts

Here are steps for accessing and using the **Triggered Alerts** page.

Prerequisites

(Optional) Review [Triggered alert listing](#).

1. From the top-level navigation bar, select **Activity > Triggered Alerts**.
2. Filter any displayed alerts according to **App, Owner, Severity**, and **Alert** (alert name).
3. (Optional) Use the keyword search to find triggered alerts by alert name or app context.
4. (Optional) Take the following actions from the Alert Manager.
 - ◇ View alert search results.
 - ◇ Edit the alert search.
 - ◇ Delete a triggered alert listing.

Configure triggered alert expiration

By default, each alert trigger record on the **Triggered Alerts** page expires after twenty-four hours. Here are steps for updating triggered alert expiration. These steps apply only to alerts with the "Add to Triggered Alerts" action enabled.

1. From the top-level navigation bar, select **Settings > Searches, reports, and alerts**.
2. Locate the alert that you want to modify under **Search Name**.
3. Select the alert. A configuration dialog opens.
4. Scroll to the **Expiration** settings dropdown.
5. Configure expiration time. Here are the available options.

Option	Additional steps for this option
Select one of the preset expiration options.	None
Select Custom	Use the text field and dropdown to define a custom expiration time.

6. Click **Save**.

Delete a triggered alert listing

By default, triggered alert records on the **Triggered Alerts** page expire after twenty-four hours. There are a few ways to change whether a triggered alert listing appears on this page.

- Update triggered alert listing expiration time.
- Delete a triggered alert listing from the **Triggered Alerts** page.
- Disable an alert to prevent it from triggering.

Enable summary indexing

Summary indexing is available on scheduled alerts. It can help you perform analysis or report on large amounts of data over long time ranges. Typically, this is time consuming and can impact performance if several users are running similar searches on a regular basis.

Prerequisites

Ensure that the alert's search generates statistical or summary data.

1. Using the top-level navigation bar, select **Settings>Searches, Reports, and Alerts**.
2. Select the alert to open the alert detail page.
3. To enable the summary index to gather data on a regular interval, set **Alert condition** to "Always".
4. Select **Enable** under **Summary Indexing**.

◇ Note that this option is unavailable for real-time alerts.

◇ If not already specified, this sets the **Alert condition** to "Always".

5. Click **Save**.

Searches and summary indexing

To use summary indexing with an alert, create a search that computes statistics or a summary for events over a period of time. Search results are saved into a summary index that you designate. You can search over this smaller summary index instead of working with the larger original dataset.

It is typical to use reporting commands in a search that populates a summary index. See *Use summary indexing for increased reporting efficiency* in the *Knowledge Manager* manual.

Update alerts from Settings

The **Searches, reports, and alerts** view in Settings lets you enter the information to create and modify alerts. Some fields for modifying an alert are available only from the Settings. You typically create alerts from the Search page by saving a search as an alert. You typically modify alerts from the Alerts page or an alert detail page.

However, you can create, view, and update alerts from Settings. From Settings you can also define the retention time and enable summary indexing for alerts. Retention time defines how long to keep a record of triggered alerts, and associated artifacts, available. Summary indexing enables faster overall searching.

Note: Creating and editing alerts from Settings is for advanced users.

To view a listing of alerts in Settings:

1. Select **Settings > Searches, reports, and alerts**.
This view lists all saved searches and reports. An alert is a type of saved search.
2. Filter the list of searches and reports using the **App context** and **Owner** menus.

Create an alert from Settings

1. In the **Searches, reports, and alerts** view in Settings, click **New**.
This opens a view that lets you create a new scheduled search.
2. Fill in the details of the scheduled search you want to create.
3. Click **Schedule this search** to create the alert.
4. Specify details for the alert.
The editing fields here correspond to the editing fields described in [Create per-result alerts](#), [Create scheduled alerts](#), and [Create rolling-window alerts](#).
5. Click **Save**.

Convert an existing search to an alert

1. In the **Searches, reports, and alerts** view in Settings, locate the search for which you want to create an alert.
2. Click the name of the search.
3. Click **Schedule this search** to create the alert.
4. Specify details for the alert.
The editing fields here correspond to the editing fields described in [Create per-result alerts](#), [Create scheduled alerts](#), and [Create rolling-window alerts](#).
5. Click **Save**.

Modify an alert from Settings

The following alert properties are only available from the **Searches, reports, and alerts** view.

- Expiration
- Summary indexing

See [Define alert retention time](#) and [Enable summary indexing for an alert](#) for details. To modify an alert from this view:

1. In the **Searches, reports, and alerts** view in Settings, locate the alert that you want to modify.
2. Click the name of the search.
3. Click **Schedule this search** to create the alert.
4. Specify details for the alert.
The editing fields here correspond to the editing fields described in [Create per-result alerts](#), [Create scheduled alerts](#), and [Create rolling-window alerts](#).
5. Click **Save**.

Define alert retention time

Retention time is how long to keep a record of triggered alerts, and associated artifacts, available. You can view the listing of triggered alerts from the detail page for an alert.

1. When editing an alert, select the retention time from the **Expiration** menu. Select from the presets or specify a custom time.
2. Verify that the **List in Triggered Alerts** check box is selected.

To review and manage your triggered alerts, go to the Alert manager by clicking the **Triggered Alerts** link on the Splunk Bar. For more information, see [Review triggered alerts](#) in this manual.

Enable summary indexing for an alert

You can enable **summary indexing** for an alert. Summary indexing lets you write the results of a report to a separate index. This enables faster searching overall. See [Use summary indexing for increased reporting efficiency](#).

- To enable summary indexing, click the **Enable** check box in the **Summary**

Indexing section.

The **Alert condition** changes to "always." Summary indexing for an alert cannot be conditional. If you want the alert to trigger on certain conditions, disable summary indexing for the alert.

Alert examples

Alert examples

This chapter shows examples of creating various types of alerts.

- Scheduled alert
- Real-time alert
- Custom conditional alert

Scheduled alert example

A scheduled alert runs periodically at a scheduled time, responding to a condition that triggers the alert.

This example uses a search to track when there are too many errors in a Splunk Enterprise instance during the last 24 hours. When the number of errors exceeds 5, the alert sends an email with information about the conditions that triggered the alert. The alert sends an email every day at 10:00AM when the number of errors exceed the threshold.

1. From the Search Page, create the following search

```
index=_internal " error " NOT debug source=*splunkd.log*  
earliest=-24h latest=now
```

2. Click **Save As > Alert**.
3. Specify the following values for the fields in the **Save As Alert** dialog box:

Title: Errors in the last 24 hours

Alert type: Scheduled

Time Range: Run every day

Schedule: At 10:00

Trigger condition: Number of Results

Trigger if number of results: is Greater than 5.

Save As Alert

Title: Errors in the last 24 hours

Description: optional

Alert type: ☒ Scheduled ☐ Real Time

Time Range: Run every day

Schedule At: 10:00

Trigger condition: Number of Results

Trigger if number of results: is Greater than 5

4. Click **Next**.
5. Click **Send Email**.
6. Set the following email settings, using tokens in the **Subject** and **Message** fields:

To: email recipient

Priority: Normal

Subject: Too many errors alert: \$name\$

Message: There were \$job.resultCount\$ errors reported on \$trigger_date\$.

Include: Link to Alert and Link to Results

Accept defaults for all other options.

For more information on tokens, see [Use tokens in email notifications](#)

Save As Alert

Enable Actions

List in Triggered Alerts ☐ Triggered Alerts is available in the activity menu.

Send Email ☒

To: cosmo@example.com

Priority: Normal

Subject: Too many errors alert: \$name\$

Message: There were \$job.resultCount\$ errors reported on \$trigger_date\$

Include

☒ Link to Alert ☒ Link to Results

☐ Search String ☐ Inline Table

☐ Trigger Condition ☐ Attach CSV

☐ Trigger Time ☐ Attach PDF

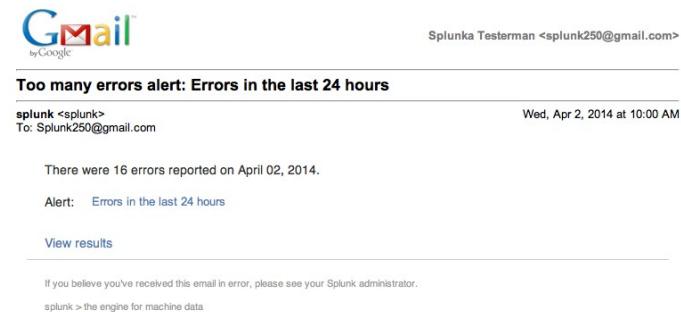
Run a Script ☐

Action Options

7. Click **Save**.

After you create the alert you can view and edit the alert in the Alerts Page.

When the alert triggers, it sends the following email:



Real-time alert example

You can configure a real-time alert to ensure that you get timely updates to the condition that triggers the alert. The procedure to configure a real-time alert is similar to that of a scheduled alert, but contains differences to ensure timely delivery.

In this example, do not specify a time range for the search. The real-time alert specifies when the search runs.

1. From the Search Page, create the following search:

```
index=__internal " error " NOT debug source=*splunkd.log*
```

2. Click **Save As > Alert**.

3. Specify the following values for the fields in the **Save As Alert** dialog box:

Title: Errors reported (Real-time)

Alert type: Real Time

Trigger condition: Number of Results

Trigger if number of results: is Greater than 5 in 1 minute.

Save As Alert

Title: Errors reported (Real-time)

Description: optional

Alert type: Scheduled Real Time

Trigger condition: Number of Results

Number of results is: Greater than 5

in: 1 minute(s)

Cancel Next

4. Click **Next**.
5. Click **Send Email**.
6. Specify the following email settings, using tokens in the **Subject** and **Message** fields:

To: email recipient

Priority: Normal

Subject: Real Time Alert: \$name\$

Message: There were \$job.resultCount\$ errors.

Include: Link to Alert, Link to Results, Trigger Condition, and Trigger Time.

Accept defaults for all other options.

For more information on tokens, see [Use tokens in email notifications](#)

Save As Alert

Enable Actions

List in Triggered Alerts: ☐

Send Email: ☒

To: cosmo@example.com

Priority: Normal

Subject: Real Time Alert: \$name\$

Message: There were \$job.resultCount\$ errors.

Include: ☒ Link to Alert ☒ Link to Results ☐ Search String ☐ Inline Table ☒ Trigger Condition ☐ Attach CSV ☒ Trigger Time ☐ Attach PDF

Run a Script: ☐

Action Options

Cancel Back Save

7. Click **Save**.
- After you create the alert you can view and edit the alert in the Alerts

Modify trigger condition

If a search takes longer to run than the time specified in the trigger condition, then the alert could fail to fire. Modify the trigger condition accordingly.

For the previous real-time alert example, assume that the search takes longer than one minute to run. To ensure the alert fires, modify the trigger condition period to 10 minutes.

Modify throttling setting

For some searches, the trigger condition can happen many times during the period configured to fire the alert. For real-time alerts, this can result in numerous emails that can overwhelm your inbox. Use the throttle action to limit the number of emails. For the previous real-time alert example, when configuring alert actions specify a reasonable time to wait before the alert fires. For example:

1. In the Edit Alert dialog box, click **Throttle**.
2. For **Suppress triggering for** enter 10 minutes.

Conditional alert

When you create an alert you specify the trigger condition for the alert. You can choose from the following trigger conditions.

Directory	Description
Per result	Triggers when the search returns a result.
Number of results	Triggers when the search returns a specified number of results.
Number of hosts	Triggers when the search returns a specified number of hosts.
Number of sources	Triggers when the search returns a specified number of hosts.
Custom	Triggers on a custom search condition.

The following example shows how to create an alert with a custom search condition. This is referred to as an **advanced conditional alert**. The example

uses a base base search that checks for all errors. The trigger condition is when an error of type WARNING occurs. The alert action lists the triggered alert.

1. From the Search Page, create the following search

```
index=_internal source="*splunkd.log" ( log_level=ERROR OR  
log_level=WARN* OR log_level=FATAL OR log_level=CRITICAL)
```

2. Click **Save As > Alert**.
3. Specify the following values for the fields in the **Save As Alert** dialog box:

Title: Warning Errors

Alert type: Real-time

Trigger condition: Custom

Custom Condition: search log_level=WARN* in 1 minute

4. Click **Next**.
5. Click **List in Triggered Alerts**.
6. Click **Save**.

After you create the alert you can view and edit the alert in the Alerts Page. When the alert triggers, the Alerts Page lists the alert in the Trigger History section.

Manual alert configuration with .conf files

Configure alerts in savedsearches.conf

You can create and configure alerts in `savedsearches.conf`.

Before configuring an alert with `savedsearches.conf`, you can review the following topics in the *Admin Manual*.

- About configuration files
- `savedsearches.conf` example.

Configuration file paths

Create or edit `savedsearches.conf` in the local directory:

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

For apps, create or edit `savedsearches.conf` in the custom application directory:

```
$SPLUNK_HOME/etc/apps/
```

Configure an alert

Here are the steps for defining alerts in `savedsearches.conf`. Steps for defining alerts in Splunk Web are not included here.

1. Create and save a search.
You can save a search as an alert or add a new stanza to `savedsearches.conf`.
2. Schedule the search.
3. Define alert triggering.
4. Configure alert actions.
If you configure an email notification for the alert, configure the email notification settings in Settings. See [Configure email notification settings](#).

Example `savedsearches.conf` stanza

The `savedsearches.conf` file contains a stanza for each saved search. The following example shows the stanza for a saved search. Within the stanza are alert attributes for the search.

```

[Too Many Errors Today]
# send an email notification
action.email = 1
action.email.message.alert = The alert condition for '$name$' in the
$app$ fired with $job.resultCount$ error events.
action.email.to = address@example.com
action.email.useNSSubject = 1

alert.suppress = 0
alert.track = 0

counttype = number of events
quantity = 5
relation = greater than

# run every day at 14:00
cron_schedule = 0 14 * * *

#search for results in the last day
dispatch.earliest_time = -1d
dispatch.latest_time = now

display.events.fields = ["host", "source", "sourcetype", "latitude"]
display.page.search.mode = verbose
display.visualizations.charting.chart = area
display.visualizations.type = mapping

enableSched = 1

request.ui_dispatch_app = search
request.ui_dispatch_view = search
search = index=_internal " error " NOT debug source=*splunkd.log*
earliest=-7d latest=now
disabled = 1

```

Schedule the search

Schedule a search in `save searches.conf` by adding the following attributes to the stanza.

Attribute	Type	Default	Description
enableSched	Boolean	false	Enable scheduling for the report.
cron_schedule	text	—	<p>Search cron schedule.</p> <p>The following cron schedule runs the search every 5 minutes:</p> <p style="text-align: center;">*/5 * * * *</p>

			<p>The following cron schedule specifies a real-time search.</p> <pre>* * * * *</pre> <p>See Cron notation for more details.</p>
dispatch.earliest dispatch.latest	time modifier	—	<p>Set the time window for a real-time alert.</p> <ul style="list-style-type: none"> • For per-event triggering, use: rt, for example <code>dispatch.earliest_time = rt</code> <code>dispatch.latest_time = rt</code> • For rolling time window triggering, use: <code>rt-[#][unit]</code>, for example <code>dispatch.earliest_time = rt-30m</code> <code>dispatch.latest_time = rt-0m</code> <p>See Specify time modifiers in your search for more information.</p>
max_concurrent	integer	1	<p>The maximum number of instances of the search that can run concurrently.</p>

Configure basic and advanced alert conditions in `savedsearches.conf`

Two categories of conditions can trigger an alert. You can configure both of these type of alerts in `savedsearches.conf`.

- **Basic conditional alerts**

Trigger alerts when the results of the search exceed the threshold for the number of events, sources, or hosts.

- **Advanced conditional alerts**

Trigger alerts based on the results of a conditional search that is evaluated against the results of the scheduled report. If the conditional search returns one or more events, the event triggers.

Configure a basic conditional alert

To configure a basic conditional alert in `savedsearches.conf`, use a combination of the following attributes:

Attribute	Type	Default	Description
counttype	text	–	<p>Set the type of count for alerting.</p> <p>Possible values:</p> <ul style="list-style-type: none">• always Default value for counttype. Triggers the alert each time the scheduled report runs. Use this value for per-result alerts. Per-result alerts are not conditional.• number of events• number of hosts• number of sources• custom Configure an advanced conditional alert.
relation	string	–	<p>Comparison factor between <code>counttype</code> and <code>quantity</code>.</p> <p>Possible values:</p> <ul style="list-style-type: none">• greater than• less than• equal to• drops by• rises by
quantity	integer	–	<p>Numeric value that triggers the alert. Use with <code>counttype</code> and <code>quantity</code>.</p>

For example, to trigger an alert if the results of a scheduled report rise by 25 between runs of the report, do the following:

```
counttype = number of events
relation = rises by
quantity = 25
```

The exception to using these settings together is to trigger an alert each time the scheduled report runs. In this case, use only the `counttype` attribute:

```
counttype = always
```

For more information, see [Set up triggering conditions for a scheduled alert](#).

Configure an advanced conditional alert

To configure an advanced conditional alert in `savedsearches.conf`, use the following attributes:

Attribute	Type	Default	Description
alert_condition	string	—	<p>A custom search string to trigger the alert.</p> <p>The search string is a secondary search of the artifacts of the report job that determines whether to trigger an alert. The alert triggers when the secondary search yields a non-empty search result list.</p> <p>If you specify <code>alert_condition</code>, set <code>counttype</code> to "custom." Do not use the other attributes for a basic conditional alert, <code>relation</code> and <code>quantity</code>.</p>
counttype	string	—	<p>Set the type of count for alerting.</p> <p>If you specify <code>alert_condition</code>, set <code>counttype</code> to "custom." Do not use the other attributes for a basic conditional alert, <code>relation</code> and <code>quantity</code>.</p>

For example:

```
counttype = custom
alert_condition = [search string]
```

For more information, see [Set up triggering conditions for a scheduled alert](#).

Configure an email alert action

Global defaults for all alert actions are configured in `alert_actions.conf`. You can override the defaults for a saved report in `savedsearches.conf`.

action.email

The `action.email` action sends email notifications when an alert triggers. The following example shows configuration parameters for `action.email`:

```
. . .
# send an email notification
action.email = 1
action.email.message.alert = The alert condition for '$name$' in the
$app$ fired with $job.resultCount$ error events.
action.email.reportServerEnabled = 0
action.email.to = Splunk250@example.com
action.email.useNSSubject = 1
. . .
```

Parameter	Type	Default	Description
action.email.to	email list	–	Comma-delimited list of email addresses to notify. You cannot define a default value for this in alert actions.conf.
action.email.from	text	splunk	The from email address for the email notification.
action.email.subject	text	Splunk Alert: \$name\$	The subject of the email notification.
action.email.sendresults	boolean	false	Include search results in the email. The can be attached or included in the body of the email. See the <code>action.email.inline</code> parameter. Results include only the results from the base search. It does not include results from secondary conditional searches.
action.email.inline	email list	–	Include results of the base search in the body of the email notification.
action.email.server	text	localhost	The address of the SMTP server that sends the alert

			emails.
email.preprocess_results	search string	empty string	Search string to preprocess results before sending the email notification. Use this parameter to filter unwanted fields.

Send SNMP traps to other systems

You can use Splunk as a monitoring tool to send SNMP alerts to other systems such as a Network Management System console.

Note: For information on how to index SNMP alerts on Splunk, read Send SNMP events to Splunk in the Getting Data In manual.

Create a script that sends the SNMP traps

Requirements

Requirements for the example script:

- Perl is required to run the script.
- Net-SNMP package is required in order to use the `/usr/bin/snmptrap` command. If you have another way of sending an SNMP trap from a shell script, modify the script as needed.
- Make sure there's admin access to the `$SPLUNK_HOME/bin/scripts` directory.
- For security reasons, scripts must reside in the `$SPLUNK_HOME/bin/scripts` directory.

Example script to send SNMP traps to other systems

Note the following:

- Create the script in the `$SPLUNK_HOME/bin/scripts` directory. Create the directory if it doesn't already exist. Copy the code listed below into `sendsnmptrap.pl`.
- Run `chmod +x sendsnmptrap.pl` to make the script executable.
- In the script, change the `Host:Port` of the SNMP trap handler, the paths to the external commands `splunk` and `snmptrap`, and the user/password if

necessary.

Sample script code

```
#!/usr/bin/perl
#
# sendsnmpttrap.pl: A script to enable using Splunk alerts to send an
# SNMP trap.
#
# Modify the following code as necessary for your local environment.
#
$hostPortSNMP = "qa-tml:162"; # Host:Port of snmpd or other SNMP trap
handler
$snmpTrapCmd = "/usr/bin/snmptrap"; # Path to snmptrap, from
http://www.net-snmp.org
$TRAPOID = "1.3.6.1.4.1.27389.1.2"; # Object Identifier for
traps/notifications
$OID = "1.3.6.1.4.1.27389.1.1"; # Object Identifier for objects, Splunk
Enterprise OID is 27389
# Parameters passed in from the alert.
# $1-$9 is the positional parameter list. $ARGV[0] starts at $1 in Perl.
$searchCount = $ARGV[0]; # $1 - Number of events returned
$searchTerms = $ARGV[1]; # $2 - Search terms
$searchQuery = $ARGV[2]; # $3 - Fully qualified query string
$searchName = $ARGV[3]; # $4 - Name of saved search
$searchReason = $ARGV[4]; # $5 - Reason saved search triggered
$searchURL = $ARGV[5]; # $6 - URL/Permalink of saved search
$searchTags = $ARGV[6]; # $7 - Always empty as of 4.1
$searchPath = $ARGV[7]; # $8 - Path to raw saved results in Splunk
instance (advanced)

# Send trap, with the parameter list above mapping down into the OID.
$cmd = qq/$snmpTrapCmd -v 2c -c public $hostPortSNMP ' ' $TRAPOID
$OID.1 i $searchCount $OID.2 s "$searchTerms" $OID.3 s "$searchQuery"
$OID.4 s
"$searchName" $OID.5 s "$searchReason" $OID.6 s "$searchURL" $OID.7 s
"$searchTags" $OID.8 s "$searchPath"/;
system($cmd);
```

For Windows

This Perl script will work on MS Windows systems with Perl. However, on some Windows systems, Perl may not be installed, or Perl scripts may not be configured to be directly executable via Splunk. In those cases, you might find it easier to use a Windows CMD script, as described in Sending SNMP traps on Windows.

Provide an MIB file

You can provide a Splunk MIB file for the SNMP monitoring agent. See [Splunk Alert MIB](#) for details.

Configure your alert to call the script

Follow these steps:

1. Create an alert. Read [About alerts](#) in the Alerting Manual for more information.
2. Set up your alert so that it calls the script. To do so, specify the name of the script (which must reside in `$SPLUNK_HOME/bin/scripts`).

☒ Trigger shell script

Filename of shell script to execute

Example script run

Here is an example of the script running, including what it returns:

```
[root@qa-tml ~]# snmptrapd -f -Lo
2007-08-13 16:13:07 NET-SNMP version 5.2.1.2 Started.
2007-08-13 16:14:03 qa-el4.splunk.com [172.16.0.121] (via UDP:
[172.16.0.121]:32883) TRAP, SNMP v1, community public
    SNMPv2-SMI::enterprises.27389.1 Warm Start Trap (0) Uptime: 96
days, 20:45:08.35
    SNMPv2-SMI::enterprises.27389.1.1 = INTEGER: 7 SNMPv2-
SMI::enterprises.27389.1.2 = STRING: "sourcetype::syslog" SNMPv2-
SMI::enterprises.27389.1.3 = STRING: "search sourcetype::syslog
starttime:12/31
/1969:16:00:00 endtime::08/13/2007:16:14:01"
SNMPv2-SMI::enterprises.27389.1.4
= STRING: "SyslogEventsLast24" SNMPv2-SMI::enterprises.27389.1.5 =
STRING:
"Saved Search [SyslogEventsLast24]: The number of hosts(7) was greater
than 1"
SNMPv2-SMI::enterprises.27389.1.6 = STRING:
"http://qa-el4:18000/?q=sourcetype
%3a%3asyslog%20starttime%3a%3a0%20endtime%3a%3a1187046841" SNMPv2-
SMI::enterprises.27389.1.7 = STRING:
"/home/tet/inst/splunk/var/run/splunk
/SyslogEventsLast24"
2007-08-13 16:14:15 NET-SNMP version 5.2.1.2 Stopped.
```

Configure a script for an alert action

The run a script alert action is officially deprecated. It has been replaced with custom alert actions as a more scalable and robust framework for integrating custom actions. See [About custom alert actions](#) for information on building customized alert actions that can include scripts.

You can configure an alert to run a shell script or batch file when the alert triggers. This topic shows how to access information about an alert in a script that runs as an alert action.

The script or batch file that an alert triggers must be at either of the following locations:

```
$SPLUNK_HOME/bin/scripts
$SPLUNK_HOME/etc/apps/<AppName>/bin/scripts
```

Working directories for scripts

Specify an absolute path whenever a path is needed. If you use relative paths, it is important to remember that they are rooted in the **Search and Reporting** app's `bin` folder. Here is the path. `$SPLUNK_HOME/etc/apps/search/bin/`

Access arguments to scripts that are run as an alert action

When you run a script as an alert action, positional arguments that capture alert information are passed to the script. The positional arguments are also available as environment variables.

You can access information from each argument using the notation in the following table.

Arg	Environment Variable	Value
0	SPLUNK_ARG_0	Script name
1	SPLUNK_ARG_1	Number of events returned
2	SPLUNK_ARG_2	Search terms
3	SPLUNK_ARG_3	Fully qualified query string
4	SPLUNK_ARG_4	Name of report
5	SPLUNK_ARG_5	Trigger reason

		For example, "The number of events was greater than 1."
6	SPLUNK_ARG_6	Browser URL to view the report.
7	SPLUNK_ARG_7	Not used for historical reasons.
8	SPLUNK_ARG_8	File in which the results for the search are stored. Contains raw results in gzip file format.

You can reference the information captured by these arguments in UNIX shell scripts or Microsoft batch files, as shown below. In other languages, such as perl and python, use the methods native to the language to access script arguments.

```
# UNIX scripts can access environment variables and positional args
$SPLUNK_ARG_0
$0
```

```
# Microsoft batch files capture environment variables reliably
%SPLUNK_ARG_0%
```

Test script that accesses positional arguments

Use the following test script to see the results of accessing the positional arguments.

To use this test script, create an alert that runs the script as an alert action. Then check the contents of the generated `echo_output.txt` file:

```
# $SPLUNK_HOME/bin/scripts/echo.sh
# simple script that writes parameters 0-7 to
# $SPLUNK_HOME/bin/scripts/echo_output.txt
# $SPLUNK_ARG_0 and $0 show how to use the long and short form.

read sessionKey
echo "'$SPLUNK_ARG_0' '$0' '$1' '$2' '$3' '$4' '$5' '$6' '$7' '$8'
'$sessionKey'" >> \
"$SPLUNK_HOME/bin/scripts/echo_output.txt"
```

- Note: The `sessionKey` is URL encoded.

For an example of how to configure scripts to work with alerts, see the topic "Send SNMP traps to other systems," in this manual.

Script example: Write to syslog

You can configure a script for an alert to write to the system log daemon. This is useful if you have syslog set up to send alerts to other applications and you want to include alerts from the Splunk platform.

1. Create a script, `logIt` that calls `logger`, or any other program that writes to syslog.

Place the script in `$SPLUNK_HOME/bin/scripts`.

2. Add the following in `logIt`:
`logger $5`

The script can access any of the arguments available when called as an alert action.

3. Create an alert on a report that runs `logIt` as an alert action.
When the alert triggers, the log entry looks something like this:
Aug 15 15:01:40 localhost logger: Report [j_myadmin]: The number of events(65) was greater than 10

See Best practices for using UDP when configuring a syslog input, a topic in the Splunk Community Wiki.

Script example: Write to the Windows Event Log

For Windows platforms, you can configure an alert action to run a script that writes to the Windows Event Log.

The following example shows a script that calls the `EVENTCREATE` utility that writes to the Event log. The script can access any of the environment variables available with an alert. You can substitute the `EVENTCREATE` utility with any command-line executable that writes to the Event Log.

1. Create the following batch file, `logIt.bat`.
Place the script in `$SPLUNK_HOME/bin/scripts`.
2. Include the following command in the batch file:
`@echo off`
`EVENTCREATE /T ERROR /SO Splunk /D %SPLUNK_ARG_5%`
Use the type that best suits the message contained in the argument. This example uses `ERROR`.
3. Create an alert to a report that runs `logIt.bat` as an alert action.

Troubleshoot scripts launched from an alert

The Splunk Community Wiki has a topic, [Troubleshooting alert scripts](#), that can help you configure and troubleshoot scripts launched from an alert.