Introduction

You can use Splunk platform visualizations to organize and communicate data insights. Visualizations and dashboards let you help users monitor or learn about important metrics and trends. You can use Simple XML or the dashboard editor to build dashboards and add interactive behavior.

Visualization Concepts

Getting Started

You can create visualizations from the Search page or when you are building a dashboard.

Search

Run a search to generate results that you want to visualize. Use Splunk Search Processing Language (SPL) commands to generate results for the visualization type that you are building.

After generating search results, click the ***Visualizations*** tab to select a visualization type and format the visualization.

Dashboard

You can create visualizations when you are building or editing a dashboard. Use the dashboard editor to add new visualizations or reuse prebuilt content.

Visualization types

There are several visualization options available in the Splunk platform. You can use the Visualization Picker interface to select a visualization type. You can also indicate a visualization type when building dashboards in Simple XML.

Visualization options include:

- Event lists
- Column, bar, area, and line charts
- Pie charts
- Scatter and bubble charts
- Single value visualizations and gauges
- Tables
- Maps
- Custom visualizations

Choose a visualization type that fits your use case and your data. For example, if you are comparing sales totals for different product types over a time period, you can use a bar or column chart. To show trends in product sales over a time period, you can use a line chart.

Search and data formatting

Data formatting means search result aggregations, data series grouping, or the result fields that a search must generate for a visualization to render.

Data format requirements vary by visualization type. When you create a visualization, you use search commands to generate results in a particular data format. This format should provide the fields or values that you want the visualization to represent.

For example, a single value visualization shows a single metric. You can use "…| stats count" to generate an aggregated count field that the single value represents.

When you hover over visualizations in the Visualization Picker, search syntax and commands are suggested to help you generate results in the correct data format.

Format visualizations

When creating or updating visualizations, you can use the ***Format*** menu to configure visualization components.

To customize visualizations in dashboards, you can also use Simple XML source code.

Depending on the visualization type, different format options are available. For example, you can configure axis label positioning in a bar or column chart. You can specify different map tiles to change the background of a Choropleth map. You can also configure ranges and colors for a single value visualization.

Publish visualizations

You can save a visualization as a dashboard panel or as a report.

Schedule reports to generate search results at a specific time interval. You can opt to include a visualization with them.

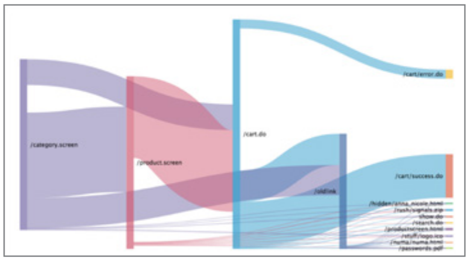
You can also save a visualization as a dashboard panel. Add it to an existing dashboard or use it to begin building a new dashboard. You can create and add more visualizations to a dashboard or edit the dashboard to add and configure content.

Additional visualization options

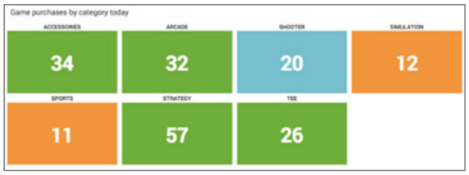
Custom visualizations

Available in Splunk Enterprise and Splunk Cloud versions 6.5.x and later.

To expand the visualization options in your Splunk deployment, you can download custom visualization apps from Splunkbase. Installed custom visualizations appear in the Visualization Picker.



Trellis layout



Available in Splunk Enterprise and Splunk Cloud versions 6.6.x and later.

You can apply trellis layout to split search results on a field or aggregation so that a visualization renders in several segments. Each segment represents a value in the split field.

Custom visualizations developer API

Available in Splunk Enterprise and Splunk Cloud versions 6.5.x and later.

Some use cases might require a customized visualization. The Splunk Custom Visualizations developer API lets you create custom visualization apps to use and share.

Visualization search examples

The following examples represent one possible use case and search for each visualization type. There are many additional search and use case options for visualizations.

Bar or column chart Compare sales totals by product.	`… | stats count by product`
Table Show sales metric for products and their product categories.	`… | stats count by product, category`
Area or line chart Show sales trends for different products.	`… | timechart count by product`
Pie chart Show how a daily sales total comprises different product categories.	`… | stats count by category`
Scatter or bubble chart Show earthquake event counts by magnitude, depth, and location.	`… | stats count by place, mag, depth`

Single value Show a current sales metric for a retail product.	…product=“video_game” | timechart count
Choropleth map Show sales totals for each state in the United States.	source=my_data.csv |lookup geo_us_states longitude as Longitude, latitude as Latitude | stats count by featureId | geom geo_us_states
Cluster map Show earthquake counts by location on a world map.	index=main mag>3 | geostats latfield=latitude longfield=longitude count

Dashboard Concepts

Dashboard

A dashboard is a group of visualizations and contextual content, such as titles and descriptions, that present information in a visual format. Dashboards use layout elements to structure their content.

Dashboard Editor

Use the dashboard editor to create and edit dashboards in Splunk Web. The editor gives you access to an editing user interface (UI) and to a Simple XML source code editor. You can use the editing UI or the source code editor to build dashboard components, change layout, and implement interactive behavior.

You can configure most dashboard functionality in either the editing UI or in Simple XML. Some advanced configurations, such as conditional drilldown, are only available in Simple XML.

Simple XML

Simple XML is source code that you can use to structure and customize dashboards. Simple XML is made up of parent and child elements. Elements can have configuration attributes. Additionally, visualization elements use <option> child elements for formatting and behavior configuration.

Most configurations that you make in Simple XML can also be made in the dashboard editing UI. Some customizations are only available in Simple XML, however. For example, conditional drilldown behavior or configuring responsive display changes require Simple XML. Use the dashboard source code editor to make these customizations.

Form

When you add inputs to a dashboard, it becomes a form. Its root element in Simple XML changes to `<form>`.

Forms and dashboards are similar in most ways. However, forms contain fieldsets to organize inputs for user interaction.

Form Input

You can add inputs to a form to capture user selections or typed text and trigger responsive behavior. Inputs are grouped inside a `<fieldset>` element in a form.

Available input types include radio buttons, selection lists, text fields, and time range pickers. Selection inputs can have static choice values and labels. You can also use a search to generate input choices dynamically.

Inputs use tokens to represent user selection or typed values. You can use input tokens to pass the user-provided value to visualizations or other elements in the dashboard and trigger responsive actions. For example, you can use an input token in a search to generate a visualization representing the value that a user selected.

Row

A dashboard uses rows to organize one or more panels horizontally.

Panels

A dashboard row contains one or more panels. Each panel has a visualization or HTML element. Panel visualizations use a search to generate the results that they render. You can use different types of searches to drive panel visualization content.

Panels have titles and descriptions that you can configure in the dashboard editor or Simple XML.

You can configure some dashboard interactivity at the panel level, including listening to token values to toggle panel display. You can also use token values to populate panel titles with dynamic values.

You can save panel content as a prebuilt panel to reuse in multiple dashboards.

Panel Searches

Searches provide the data that visualizations represent in dashboards.

You can use different types of searches to generate dashboard content.

- Inline search strings directly in a panel
- Saved searches that you reference in a panel
- Searches in prebuilt panels that you reuse
- Global base searches whose results you use with post-process searches to generate different results in various panels
- Searches generated with Pivot

Permissions

Dashboards

Dashboards are knowledge objects with access and editing permissions. Your user role and capabilities determine your options for creating, sharing, and administering dashboards.

If you have the admin role and its default capabilities, you can configure dashboard visibility in different apps in your deployment. You can also set read and write permissions associated with specific roles.

Saved searches in dashboards

Saved searches in a dashboard are knowledge objects with independent permissions. A saved search can run with the permissions of the user who created it or the user who is viewing its results, including in a visualization.

Depending on saved search permissions in your dashboards, some users might see visualizations that represent a more limited result set. Your user role and capabilities determine your options for adjusting permissions to manage your dashboard user experience.

Building Interactive Dashboards

Drilldown

Drilldown is a tool for creating dashboard interactivity. You can add drilldown to a dashboard visualization to share additional data insights with users when they click on it. Use the drilldown editor (Splunk Enterprise and Splunk Cloud versions 6.6.x and later) and Simple XML to add and configure drilldown in your dashboards.

Drilldown actions

Drilldown can trigger different interactive responses to a user click. You can configure drilldown to open a secondary search, another dashboard, or an external URL in the browser. You can also use drilldown to trigger contextual changes in the same dashboard.

You can use tokens to customize content in a drilldown target. Use tokens to capture and pass values to a drilldown target, such as a search string or a URL, and customize its content. You can also use tokens to trigger interactive content display, such as showing or hiding a panel or updating a visualization title.

Tokens

Tokens are like programming variables. They represent data that changes, such as a search result field, a user selection in an input, a user click for drilldown, a search result field value, or a flag that you set to trigger interactive behavior. As with programming variables, you can use tokens to capture dynamic values and to access them.

Some tokens are predefined in Splunk software. You can also create custom tokens to represent additional values or to control dashboard behavior.

Token syntax requires dollar signs or quotation marks around a token name. For example, $click.value$ references a clicked field value in a visualization where drilldown is enabled. Check Splunk documentation for more details on syntax, including special character escaping.

Predefined tokens for drilldown

Predefined token	Table	Chart	Single value	Map
$click.name$	Leftmost field (column) name in the table.	X-axis field or category name for the clicked location	Name of field that single value represents	Field name for the clicked location
$click.value$	Leftmost field (column) value in the clicked table row.	X-axis field or category value for the clicked location	Field value that the single value represents	Field value for the clicked location
$click.name2$	Clicked table cell field name.	Y-axis field or category value for the clicked location	Same as `$click.name$`	Same as `$click.name$`
$click.value2$	Clicked table cell value.	Y-axis field or category value for the clicked location	Same as `$click.value$`	Same as `$click.value$`
$row.<fieldname>$	Access any field (column) value from the clicked table row. For example, to get the sourcetype field value in the clicked row, use `$row.sourcetype$`	Access any y-axis field value corresponding to the clicked location x-axis. Not available if the user clicks the chart legend.	Access any field value from the Statistics table row for the single value.	Access field values related to the clicked location. Check the Statistics tab for available fields.

Drilldown examples

LINK TO A SEARCH

Goal: Open a secondary search in the browser when a user clicks on a visualization. Show search results related to the clicked value.

Scenario: A table shows customer actions on a retail website. When a user clicks on a table cell, show search results filtered for the clicked value.

How to set up the drilldown:

- Define the `<drilldown>` behavior using the `<link>` element.
- Include the `<target=”_blank”>` attribute to open the search in a new browser tab.
- Use the predefined `$click.value2$` token to populate the search with the clicked action field value.
- Wrap the search in `<![CDATA[]]>` tags to escape special characters in the search string.

Simple XML source code:

```
<drilldown>
    <link target=”_blank”>
      <![CDATA[ search?q=source=”my_retail_data_source” action=$click.value2$
              | stats count by productId&earliest=-24h@h&latest=now
      ]]>
    </link>
</drilldown>
```

LINK TO A DASHBOARD OR FORM

Goal: Open a target dashboard or form in the browser when a user clicks on a visualization. Show content customized to the clicked value and contextual values from the source dashboard.

Scenario: A table shows top sourcetypes. When a user clicks a table row, open a form showing content customized to the clicked row's sourcetype value.

How to set up the drilldown:

- Define the `<drilldown>` behavior using the `<link>` element.
- Include the `<target=”_blank”>` attribute to open the search in a new browser tab.
- Use the `$row.<fieldname>$` predefined token to capture the sourcetype field value from the clicked row.
- Pass the clicked value to the form by setting the `$form.sourcetype$` token value to the clicked value. When setting form tokens, prefix the token name with `form.<token_name>`. No prefix is necessary for setting tokens in a target dashboard.
- Pass the earliest and latest time range settings from the source dashboard to the target form.
- Wrap the search in `<![CDATA[]]>` tags to escape special characters in the search string.

Simple XML source code:

```
<drilldown>
    <link target=”_blank”>
            <![CDATA[
                  /app/search/form_for_drilldown?
                  form.sourcetype=$row.sourcetype$&earliest=$earliest$&latest=$latest$
            ]]>
    </link>
</drilldown>
```

LINK TO A URL

Goal: When a user clicks on a visualization, open a related website in the browser.

Scenario: A visualization shows failed website logins. Link to internal documentation when a user clicks on the visualization. Use conditions to specify linking to a runbook for handling excessive failed logins if the failure count is more than 5,000. Otherwise, link to an overview page on failed logins.

How to set up the drilldown:

- Inside the single value `<search>`, use the `<done>` search event handler element to access a result count value when the search completes.
- In the `<done>` element, set a custom count token to the value of the predefined `$result.count$` token. The `$result.count$` value is only available in the context of the `<search>` element. It cannot be accessed directly in a `<drilldown>`. Setting the count token lets you access this value in the `<drilldown>` element.
- Define the `<drilldown>` behavior using the `<link>` element.
- Inside the drilldown, set up conditional behavior. Use the `<condition match=” ”>` element to evaluate and respond to the `$count$` value when a user clicks on the visualization. In this scenario, a failed login count higher than 5,000 triggers the failed login runbook opening in the browser. A lower count triggers an informational web page opening instead.

```
 <single>
      <search>
        <query>source=”recent_login_events” type=failed_login | stats count</query>
          <earliest>-24h@h</earliest>
          <latest>now</latest>
        <done>
         <set token=”count”>$result.count$</set>
        </done>
      </search>
      <drilldown>
        <condition match=”$count$ > 5000”>
           <link>
              http://companydocs.com/high_failed_login_runbook
```

```
            </link>
         </condition>
         <condition match=”$count$ < 5000”>
            <link>
               http://companydocs.com/about_failed_logins
            </link>
         </condition>
       </drilldown>
   </single>
```

TRIGGER CONTEXTUAL CHANGES IN THE SAME DASHBOARD

Goal: When a user clicks on a visualization, show customized content in the same dashboard.

Scenario: A table visualization shows event counts by sourcetype and log level. When users click on a sourcetype value, the dashboard shows a single value aggregating events for the selected sourcetype. When users click on a log level value, an events list for the clicked log level appears instead.

How to set up the drilldown:

- Put `<condition>` elements inside the `<drilldown>` to define the two conditional responses to user clicks.
- Use the field attribute in each `<condition>` to check whether the user clicked a value in the `sourcetype` or `log_level` column.
- Use `<set>` to capture the clicked `sourcetype` or `log_level` value using the `$click.value2$` predefined token.
- Use additional `<set>` and `<unset>` elements to manage token values that control panel display.
- Use depends attributes in the panels to respond to token value changes from the `<drilldown>`. A `depends` attribute means that the panel displays only when the specified token is set. Similarly, you can use a `rejects` attribute to indicate that a panel should not display if the specified token is set.
- Use the `$selected_sourcetype$` and `$selected_log_level$` tokens in the panel search strings to generate content relevant to the user's clicked value.

Simple XML source code:

```
<row>
   <panel>
     <table>
       <title>Event counts by sourcetype and log level</title>
       <search>
         <query>index=_internal | stats count by sourcetype, log_level</query>
       </search>
       <drilldown>
         <condition field=”sourcetype”>
           <set token=”selected_sourcetype”>$click.value2$</set>
           <set token=”show_single_value”>true</set>
            <unset token=”show_event_list”></unset>
         </condition>
         <condition field=”log_level”>
           <set token=”selected_log_level”>$click.value2$</set>
           <set token=”show_event_list”>true</set>
            <unset token=”show_single_value”></unset>
         </condition>
       </drilldown>
     </table>
   </panel>
 </row>
 <row>
   <panel depends=”$show_single_value$”>
     <title>Event count for $selected_sourcetype$</title>
     <single>
```

```
        <search>
          <query>index=_internal sourcetype=$selected_sourcetype$ | stats count</query>
        </search>
        [...]
      </single>
    </panel>
    <panel depends="$show_event_list$">
      <title>Last five events with log level $selected_log_level$</title>
      <event>
        <option name="count">5</option>
        <search>
          <query>index=_internal log_level=$selected_log_level$</query>
        </search>
      </event>
    </panel>
       </row>
```

Additional Resources

There are many additional resources to help you with creating visualizations and dashboards.

Splunk Documentation

docs.splunk.com

Dashboards and Visualizations

docs.splunk.com/Documentation/Splunk/latest/Viz

Topics include guidance on:

- Search and data formatting for visualizations
- Visualization configurations
- Using trellis layout
- Building dashboards in the dashboard editor user interface
- Building dashboards in Simple XML
- Dashboard permissions
- Drilldown and dashboard interactivity
- Token usage in dashboards

Troubleshooting and reference topics:

- Chart display issues
- Searches power dashboards and forms
- Simple XML reference
- Event handler reference

Custom Visualization Apps

docs.splunk.com/Documentation/CustomViz

Custom Visualization Developer API documentation

docs.splunk.com/Documentation/Splunk/latest/AdvancedDev/CustomVizDevOverview

Splunk Dashboard Examples App

splunkbase.splunk.com/app/1603/

Splunk Education courses

splunk.com/view/education/SP-CAAAAH9

Splunk Answers

answers.splunk.com

Splunk user community on Slack

splunk-usergroups.signup.team/

splunk>

splunk.com
docs.splunk.com

Splunk Inc.
270 Brannan Street
San Francisco, CA 94107