Following best practices for working with dashboards

Dashboards are a great way to organize and communicate data insights. They help users monitor or learn about important metrics and trends. However, setting them up and running them isn't always as simple as you'd like it to be. Editing the XML can be time consuming and cumbersome. You need some ways to make dashboard setup and maintenance easier.

Here are four ways you can work with dashboards more effectively:

- **Use base searches**
- **Use text inputs with tokens for conditional logic**
- **Preload drilldowns**
- **Embed saved searches**

Use base searches

Running lots of separate searches might cause your to bump up against quotas and other limits and can negatively affect performance. If you use a base search as a starting point to build other searches from, you can avoid these issues.

You can use multiple base searches on a dashboard, but there are a few caveats to using base searches.

- They work best with transforming searches.
- They have a 50,000 event limit and require field references for non-transforming searches.
- They have a 30 second search limit.

Build a new base search

You can build a base search directly in the XML. This example has a search id of "main_search", and it sets the index, sets the time range, and creates a timechart.

```
<dashboard>
  <label>Base Search</label>
  <search id="main_search">
    <query>
      index=_internal | timechart count
    </query>
    <earliest>-24h</earliest>
    <latest>@</latest>
  </search>
</dashboard>
```

Then use the search ID you created to add the search to a panel. This example adds a stats command to the base search.

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Reference a saved search

A second option for building a base search is to reference a saved search. Saved searches are easily reusable across many dashboards, and if they run on an interval, the dashboard always displays the latest information. This example has a search ID of "energyBase" and references the saved search "Device Energy Consumption".

```
<row>
  <panel>
    <single>
      <search base="main_search">
        <query>stats sum(count) as count</query>
      </search>
      <drilldown>
        <set token="Show_4_row"></set>
      </drilldown>
    </single>
  </panel>
</row>
```

Use the search base attribute to reference the base search ID. This example also adds a stats command to the base search.

```
<row>
  <panel></panel>
  <panel>
    <single>
      <title>Todays Electricity Cost</title>
      <search base="energyBase">
        <query>stats latest(cost) device.name = "Energy Monitor channel 1"</query>
      </search>
    </single>
  </panel>
</row>
```

Use text inputs with tokens for conditional logic

Tokens are variables you can use to make your dashboards interactive. For example, you might use them for drilldowns, using set or unset to define which drilldowns apply to which panel. However, in dashboards with lots of panels and lots of tokens, this process can become time consuming and error prone, as you have to set or unset each panel separately.

A better way to manage tokens is by creating a single token that hides all other tokens by default.

1. At the bottom of your XML, create a row that is always hidden using depends="$always_hidden$". This is where your inputs for the token will live. On the dashboard, you can tell the row is hidden because it has a dotted line border.
2. In the first set of inputs:
   1. Define the hide_row_1 token.
   2. Configure all drilldown tokens to unset.
3. Create a form input that will show the drilldown token you request using <set>token="form.show_row_1">$value$</set>.

3. In the second set of inputs:
   1. Define the show_row_1 token from the first set of inputs.
   2. Set conditional values for all your other tokens.

```
<row depends="always_hidden$">
  <panel>
    <input type="text" token="hide_row_1">
      <default></default>
      <change>
        <unset token="show_weather"></unset>
        <unset token="show_energy"></unset>
        <unset token="show_mower"></unset>
        <unset token="show_washer"></unset>
        <set token="form.show_row_1">$values$</set>
      </change>
    </input>
    <input type="text" token="show_row_1">
      <default></default>
      <change>
        <condition value="weather">
          <set token="show_weather"></set>
        </condition>
        <condition value="energy">
          <set token="show_energy"></set>
        </condition>
        <condition value="mower">
          <set token="show_mower"></set>
        </condition>
        <condition value="washer">
          <set token="show_washer"></set>
        </condition>
      </change>
    </input>
  </panel>
</row>
```

Now, in your drilldown, you can use the hide_row_1 token to hide all drilldowns except the one you specify, which is weather in the example below.

```
<single>
  <title>Weather</title>
  <search></search>
  <option name="drilldown">all</option>
  <drilldown>
    <set token="form.hide_row_1">weather</set>
  </drilldown>
</single>
```

If you need to make additional changes, you only have to edit the inputs on your hidden row instead of each panel in your dashboard.

---

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
Preload drilldowns

Drilldowns can take a long time as you wait for a search to run again. Instead, you can use a drilldown search from a parent dashboard and reload it into a target dashboard.

When a search completes, use the change event handler of "done". Assign the search ID (sid) job property with $job to a token with a name related to the search, as shown here:

```
<search id="energyBase" ref="Device Energy Consumption">
  <done>
    <set token="energy_sid">$job.sid$</set>
  </done>
</search>
```

Next, add the drilldown block to your panel. Identify your target dashboard and then use form.input_sid to pass the token. You don’t have to use an input to pass the token, but doing so gives you greater flexibility for conditional handling.

```
<drilldown html targets="Blank" app="westys_world/Westy_house_electricity_use" form.input_sid="energy_sid" />
```

Finally, in the search block, use loadjob to add the form input. You can also set further tokens that define actions to take if the search results in an Error, Fail, or Done, as shown in this example:

```
<search>

  <query>1 loadjob $form.input_sid$</query>
  <error>
    <set token="tok_fail">Device Energy Consumption</set>
  </error>
  <fail>
    <set token="tok_fail">Device Energy Consumption</set>
  </fail>
  <done>
    <set token="form_base_sid">$result.info_sid$</set>
  </done>
</search>
```

Embed saved searches

If you have a search that is saved as a report and you want to embed it in your dashboard, you must use an iFrame. However, this means the search cannot be edited until you disable the iFrame. Then you make your edits and create the iFrame again. This is especially problematic if you have the search embedded onto multiple dashboards.

A faster way to edit this search is to realize that saved searches are macros, which you can edit directly. In the edit menu, select Open in Search and you will see the macro name, identified by the back ticks that surround it. Find the macro in Settings > Advanced Search > Search macros and edit it as needed.

For more information, see:

- Use search macros in searches
Next steps

The content in this guide comes from a **Conf21 breakout session**, one of the thousands of Splunk resources available to help users succeed. In addition, these Splunk resources might help you understand and implement this use case:

- Splunk 2021 BSides: [Dashboarding Wowzas](#)
- White Paper: [Dashboards quick reference guide](#)