

# SPLUNK SIMPLE XML FORM CHEAT SHEET

## To Create a Form From Scratch

1. Start with a simple search using one of the values you will eventually change with a pulldown.
2. Click on "Create" (upper right area) and follow the instructions to create a Dashboard Panel, including saving the search and creating a new dashboard.
3. Open the new dashboard, turn on edit mode.
4. Click the edit button on the panel and choose "Edit search".
5. Click on "Edit as inline search" and hit "Save".
6. Click on "Edit XML" button.
7. Change "<dashboard>" and "</dashboard>" to "<form>" and "</form>" respectively.
8. At the top of the form, add "<fieldset>" and "</fieldset>". Between those tags, add the desired input elements and options. Use the rest of this document for reference.

## To Modify an Existing Form

*There are many ways to skin this cat, including editing the text file directly. This is the one that I've found works the best for me.*

1. Go to "Manager > User Interface > Views"
2. Find the form you want to modify.
3. Under the "Actions" column, option click (or right click) on "Open" in order to open the form in a new window or tab.
4. Then click on the name of the form in the "View name" column to open the Simple XML interface. At this point, you have one window or tab with the edit interface and one with the finished form.
5. After you make modifications in the Simple XML interface, hit save, move to the other window and tab and reload. You should see your changes.
6. As you iterate, it may be useful to learn the quick key for "previous page" and for switching tabs or windows in your browser.

## To Modify raw Text

The brave among you may also modify the raw text. Edit in: `$$SPLUNK_HOME/etc/users/[you]/search/local/data/ui/views/[dashboard_name].xml`  
If you do this, you must refresh after each change. In a new browser window, load the url:  
`http://[your splunk host]:8000/en-US/debug/refresh?entity=/[you]/views`

## FIELDSET

All form elements must be surrounded by the fieldset tags.

### Code Sample (italics = optional)

```
<fieldset autoRun="true" submitButton="false">  
...  
</fieldset>
```

### Options

<b>autoRun</b>	When "true", searches are triggered at page load. Default is "false".
<b>submitButton</b>	When "false", button omitted from form, and form is triggered when form elements change. Default is "true".

## TIME

Generates a pulldown to modify the time range of all searches on the page.

*Note: time specifications embedded in searches will override this.*

### Code Sample (italics = optional)

```
<input type="time">  
  <default>Last 30 minutes</default>  
  <label>Enter a Label here</label>  
</input>
```

### Options

<b>default</b>	If value between tags exactly matches one of the values in pulldown, the pulldown will load with that value. NOTE: doesn't seem to work with Splunk 5.
<b>label</b>	Value will appear as label of form element.

## TEXT

Generates a text box. Values are passed along to variables surrounded by dollar signs (i.e. "\$user\$") embedded in searches.

### Code Sample (italics = optional)

```
<input type="text" token="user">  
  <default>admin</default>  
  <label>Enter User Name</label>  
  <prefix>user="</prefix>  
  <suffix>"</suffix>  
</input>  
...  
<searchString>user=$user$</searchString>
```

### Options

<b>token</b>	Required. This is the variable name that will be plugged into subsequent searches. For example, if token is "username", the search will be "user=\$username\$". See code sample for illustration.
<b>default</b>	Value will appear in text box at page load and will be submitted when the dashboard is loaded.
<b>label</b>	Value will appear as label of form element.
<b>prefix</b>	Value will be prepended to submitted value. For example, if prefix is "username=" and "Fred" is entered in the form, \$username\$ will not be "Fred" but "username=Fred".
<b>suffix</b>	Value will be appended to submitted value.

## PULLDOWN AND RADIO BUTTONS

Pull-down and radio input elements are formatted identically. Only the type in the input tag is different.

### Code Sample (italics = optional)

```
<input type="dropdown" token="username">
  <default>Sam</default>
  <label>Choose User Name</label>
  <prefix>username=</prefix>
  <suffix>"</suffix>
  <choice value="admin">admin</choice>
  <populatingSearch fieldForValue="user" fieldForLabel="user" earliest="-1d@d" latest="-0d@d">
    <![CDATA[index=_audit | stats count by user]]>
  </populatingSearch>
</input>
```

### Options

<b>type</b>	Required. Choose between "dropdown" and "radio", depending on your preference.	
<b>token</b>	Required. This is the variable name that will be plugged into subsequent searches. For example, if token is "username", the search will be "user=\$username\$". See code sample for illustration.	
<b>default</b>	Value will appear in text box at page load.	
<b>label</b>	Value will appear as label of form element.	
<b>prefix</b>	Value will be prepended to submitted value. For example, if prefix is "username=" and "Fred" is entered in the form, \$username\$ will not be "Fred" but "username=Fred".	
<b>suffix</b>	Value will be appended to submitted value.	
<b>choice</b>	Manually add an item to the pull-down or radio form. Specify value within the tag and label between tags, as in "<choice value="value">label</choice>". Will appear before values generated by populating search.	
<b>populatingSearch</b> or <b>populatingSavedSearch</b>	populatingSearch and populatingSavedSearch add values to a pull-down or collection of radio buttons from a Splunk search. Containing the search in a CDATA tag (see code sample) will ensure that special characters in the search are not unintentionally parsed by the XML.	
	<b>fieldForLabel</b>	Required. Field extracted from the populating search and placed as the label for the list option.
	<b>fieldForValue</b>	Required. Field extracted from the populating search and placed in the value of the generated list option.
	<b>earliest</b>	Optional. Restrict search results to a specific time window, specifying one or both of these attributes. For example, specify earliest="-7d" latest="-1d". Specify "rt" to enable real-time searches. <i>Only available for populatingSearch.</i>
	<b>latest</b>	

## Putting it All Together

```
<?xml version='1.0' encoding='utf-8'?>
<form>
<fieldset>
  <input type="time">
    <default>Last 30 days</default>
    <label>Choose a time range</label>
  </input>

  <input type="text" token="user">
    <default>admin</default>
    <label>Enter User Name</label>
    <prefix>user=</prefix>
    <suffix>"</suffix>
  </input>

  <input type="dropdown" token="user">
    <default>admin</default>
    <label>Choose User Name</label>
    <prefix>user=</prefix>
    <suffix>"</suffix>
    <choice value="admin">admin</choice>
    <populatingSearch
      fieldForValue="user"
      fieldForLabel="user"
      earliest="-1d@d"
      latest="-0d@d">
      <![CDATA[index=_audit | stats count by user]]>
    </populatingSearch>
  </input>
</fieldset>

<label>Pull-down Test</label>
<row>
<event>
  <title>Event view</title>
  <searchString>index=_audit $user$</searchString>
  <fields>added deleted changed</fields>
  <option name="showPager">true</option>
  <option name="count">20</option>
  <option name="displayRowNumbers">false</option>
</event>
</row>
</form>
```