

Best Practices for Developing Splunk Apps and Add-ons

Jason Conger

Staff Solutions Architect, Splunk

.conf2016

splunk>

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

whoami



jason.conger@splunk.com



[@JasonConger](https://twitter.com/JasonConger)



<http://www.linkedin.com/in/JasonConger>



blogs.splunk.com/author/jconger/
www.JasonConger.com

4+ years at Splunk

Created or consulted on numerous Splunkbase applications



Staff Solutions Architect
Global Strategic Alliances

Agenda

1. Creating a Splunk Application
2. Getting data into Splunk
3. Asking questions of your data with Splunk

“I wish I knew these things before I ever built my
first Splunk Application”

- Jason Conger

2



2



2

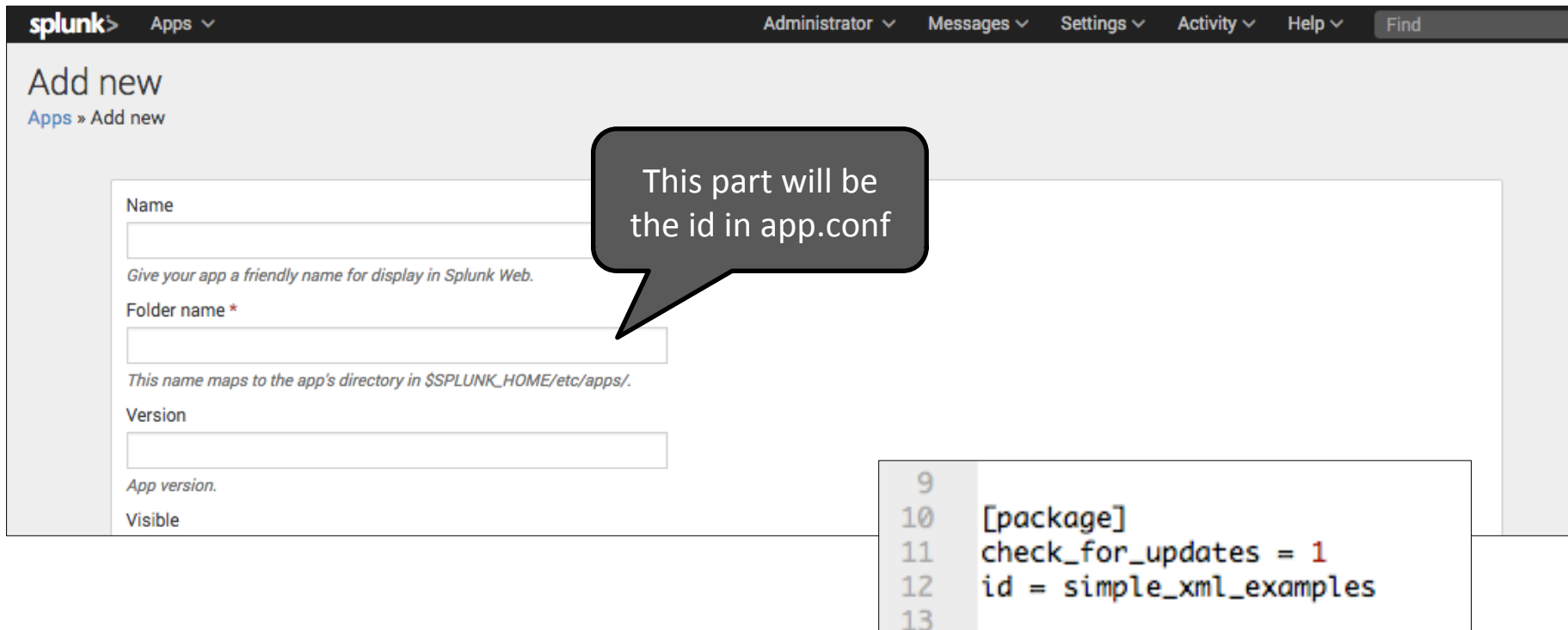


1

Creating An Application

.conf2016

Naming the Directory for Your App or Add-on



The screenshot shows the Splunk web interface for adding a new app. The top navigation bar includes 'splunk>' and 'Apps' with a dropdown arrow. The main header contains 'Add new' and a breadcrumb 'Apps » Add new'. The form fields are: 'Name' (with a hint 'Give your app a friendly name for display in Splunk Web.'), 'Folder name *' (with a hint 'This name maps to the app's directory in \$SPLUNK_HOME/etc/apps/.'), 'Version' (with a hint 'App version.'), and 'Visible'. A callout bubble points to the 'Folder name' field, stating: 'This part will be the id in app.conf'. To the right, a code block shows the corresponding configuration in app.conf.

splunk> Apps ▾

Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Add new

Apps » Add new

Name

Give your app a friendly name for display in Splunk Web.

Folder name *

This name maps to the app's directory in \$SPLUNK_HOME/etc/apps/.

Version

App version.

Visible

This part will be the id in app.conf

```
9  
10 [package]  
11 check_for_updates = 1  
12 id = simple_xml_examples  
13
```

Naming the Directory for Your App or Add-on

- ❑ For applications (dashboards, forms, alerts, etc.):
 - Vendor-app-product (example = acme-app-widget)
- ❑ For add-ons (data collection with no dashboards):
 - TA_vendor-product (example: TA_acme-widget)
- ❑ For Enterprise Security add-ons:
 - TA-<datasource> (example: TA-snort)

Note: you may see some other naming standards such as SA or DA out there.

Naming Your App or Add-on



Note: after uploading an application to Splunkbase, the directory name and the “id” parameter in app.conf cannot be changed.

The actual name of the application displayed on the Splunk start screen and on Splunkbase is controlled by a file named app.conf and is independent of the directory name mentioned previously.

App naming guidelines -> <http://docs.splunk.com/Documentation/Splunkbase/latest/Splunkbase/Namingguidelines>

Should You Break Up Your App?



Consolidated App



Distributed App

Should You Break Up Your App?

- Do you need to collect data from forwarders?
- Need to share knowledge objects with multiple apps?
- Distributed Environment?
- The Splunk App for AWS is a good example



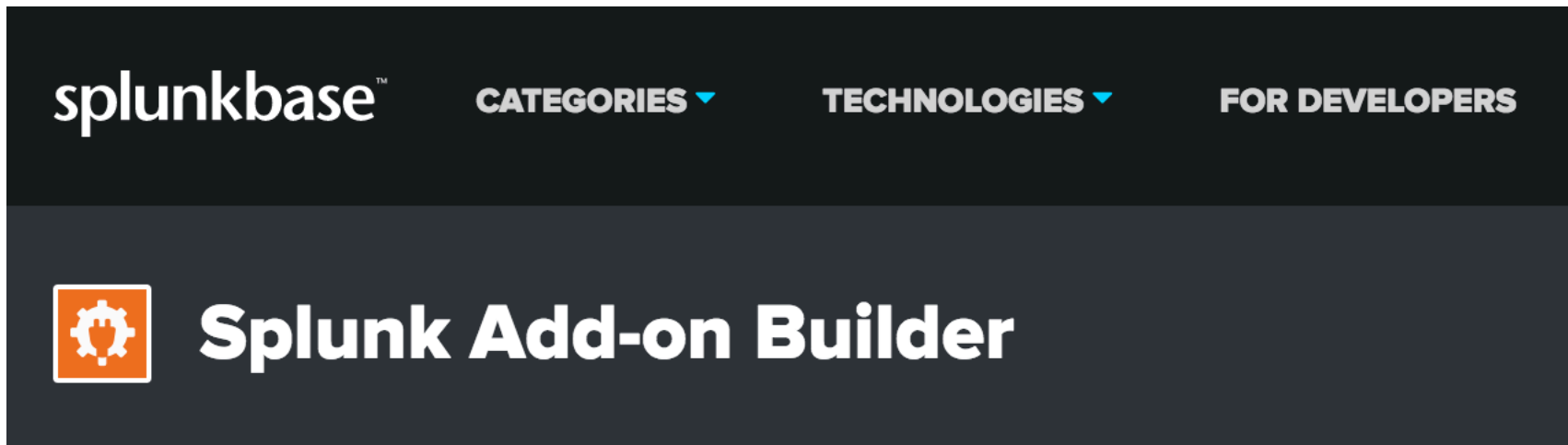
Splunk App for AWS

depends on



Splunk Add-on for Amazon Web Services

Quick Start = Splunk Add-on Builder



<https://splunkbase.splunk.com/app/2962/>

Home > Create project:



Name Project



Configure Data
Collection



Upload Sample
Data



Extract Fields



Map to CIM



Validate



Summarize

Step 1: Name Project

Home > Create project:TA_maxtest

Name Project

Configure Data
Collection

Upload Sample
Data

Extract Fields

Map to CIM

Validate

Summarize

Step 2: Configure Data Collection

Home > Create project:TA_dsg-demo

Name Project

Configure Data
Collection

Upload Sample
Data

Extract Fields

Map to CIM

Validate

Summarize

Step 4: Extract Fields > dsg.demo2

Build the field extractions for your add-on by selecting a sourcetype, then clicking Parse to parse the data. Or, click

- 1100 events in all
- ☒ Enable/Disable All
 - ☒ Group1
816 events, 74.2%
 - ☒ Group2
73 events, 6.6%
 - ☒ Group3
34 events, 3.1%
 - ☒ Group4
20 events, 1.8%
 - ☒ Group5
19 events, 1.7%
 - ☒ Group6
17 events, 1.5%
 - ☒ Group7
15 events, 1.4%
 - ☒ Group8
14 events, 1.3%
 - ☒ Group9
8 events, 0.7%
 - ☒ Group10
8 events, 0.7%
 - ☒ Group11
7 events, 0.6%
 - ☒ Group12
6 events, 0.5%
 - ☒ Group13
6 events, 0.5%

Pattern: `$($ipv4_1) %$(field_1); Built $(direction_1) $(transport_1) connect`

☐ Show the regular expression

Fields:

`ipv4_1` `field_1` `direction_1` `transport_1`

Events:

73 events, 100% matched, 0% unmatched.

- ✓ Mar 15 12:01:09 10.160.205.10 %ASA-6-302013: Built inbound TCP
- ✓ Mar 15 12:01:21 10.160.205.10 %ASA-6-302013: Built inbound TCP
- ✓ Mar 15 12:00:58 10.160.205.10 %ASA-6-302015: Built inbound UDP
- ✓ Mar 15 12:00:57 10.160.205.10 %ASA-6-302016: Built inbound UDP
- ✓ Mar 15 12:00:29 10.160.205.10 %ASA-6-302016: Built inbound UDP
- ✓ Mar 15 12:00:28 10.160.205.10 %ASA-6-302015: Built inbound UDP
- ✓ Mar 15 11:57:57 10.160.205.10 %ASA-6-302013: Built inbound TCP
- ✓ Mar 15 12:00:32 10.160.205.10 %ASA-6-302015: Built inbound UDP
- ✓ Mar 15 12:00:31 10.160.205.10 %ASA-6-302015: Built inbound UDP
- ✓ Mar 15 12:00:49 10.160.205.10 %ASA-6-302013: Built inbound TCP
- ✓ Mar 15 11:57:35 10.160.205.10 %ASA-6-302016: Built inbound UDP
- ✓ Mar 15 11:58:56 10.160.205.10 %ASA-6-302013: Built inbound TCP

Home > Create project:TA_dsg-demo



Step 5: Map to CIM

Map fields from your add-on to the Common Information Model. Start by selecting an event type. If the dropdown list doesn't show yo

Events

* Select an event type

dsg_eventtype_demo1

Add Event Type

CIMs

* Select a CIM data model

Network_Traffic

Home > Create project:TA_dsg-demo

Step 6: Validate

*Validation category: ☒ Best Practice ☒ CIM Mapping ☒ Field Extract ☒ Modular Input

Validate

93

Health Score

Failure Rule Count

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

5

2.5

Warning

7.5

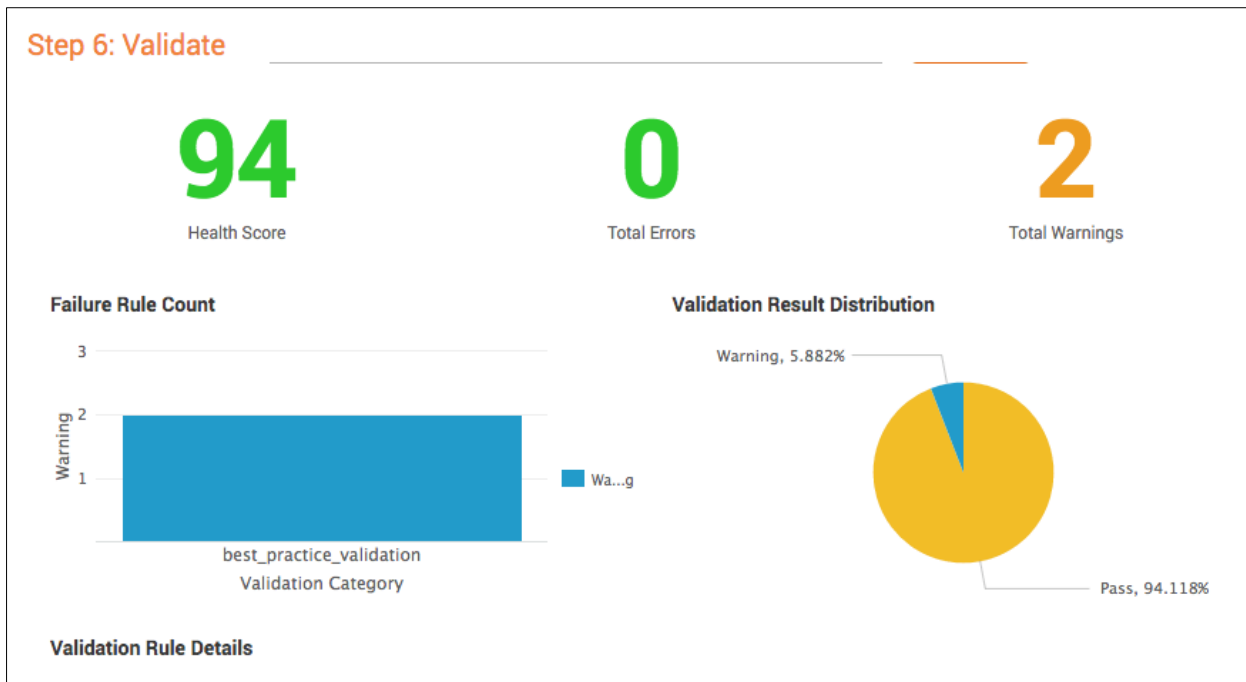
5

2.5

Warning

7.5

Use the Builder on Existing Content Too



Note: you may get some inapplicable warnings for apps since this version is mainly about add-ons.

2

Getting Data In

.conf2016

Getting Data In

Reaching out to get data	Listening for data
<ul style="list-style-type: none">• Reading files on a disk• Windows Inputs<ul style="list-style-type: none">• Perfmon• Event Logs• Registry• WMI• Scripts*• Modular inputs*	<ul style="list-style-type: none">• TCP• UDP• HTTP• Stream• Scripts*• Modular inputs*

* Scripts and modular inputs can really do either depending on what you code

Best Practices for Logging Data to be Consumed by Splunk

- Log in text format
- Start the log line event with a time stamp
- Use clear key-value pairs
- Create events that humans can read
- Use unique identifiers
- Keep multi-line events to a minimum
- Use JSON (JavaScript Object Notation) format

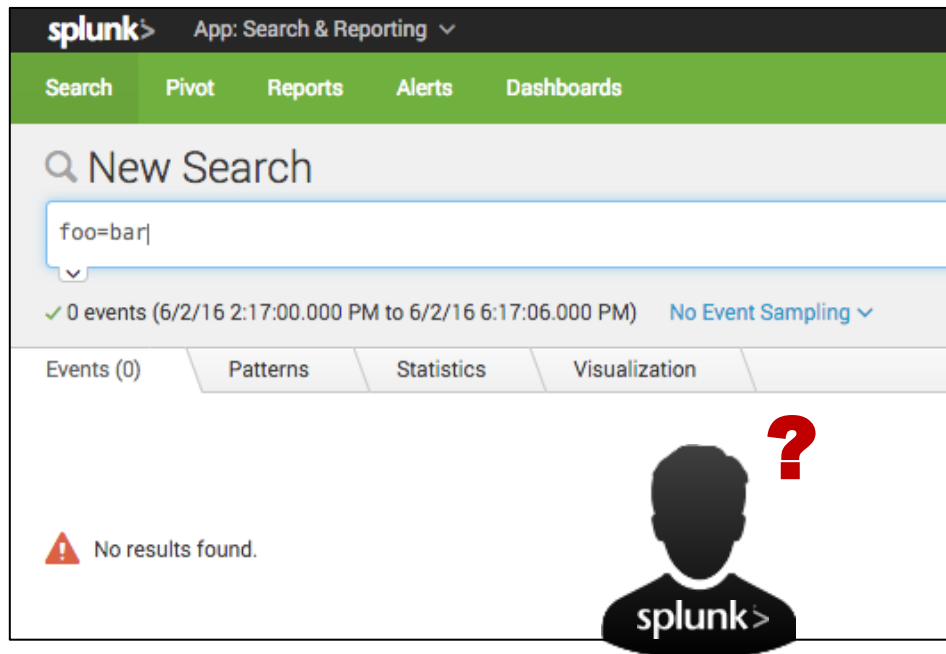
<http://dev.splunk.com/view/logging-best-practices/SP-CAAADP6>

Best Practices for Writing Data to an Index

Write to the default “main” index



main



custom index
contains foo=bar
data

Best Practices for Writing Data to an Index

Write to the default “main” index

Indexes searched by default


Set the index(es) that searches default to when no index is specified. User with this role can search other indexes using index= (e.g., "index=special_index").

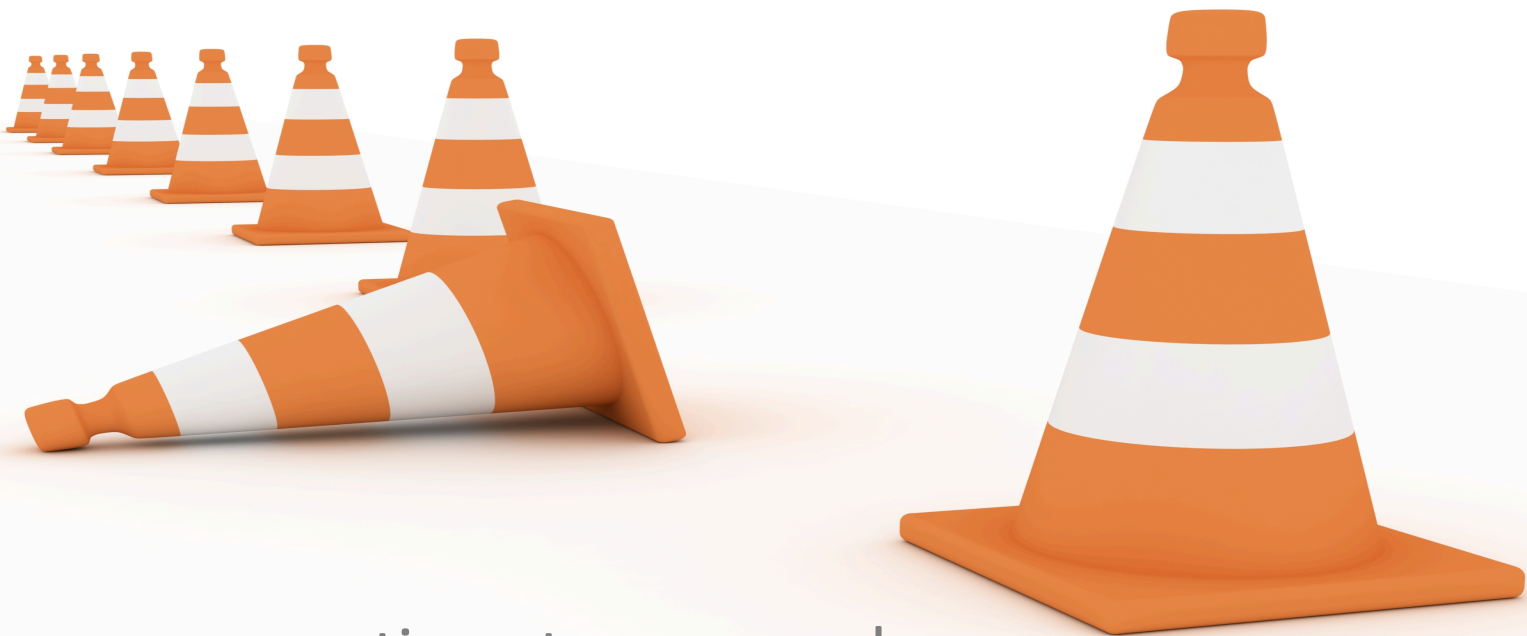
Available indexes [add all »](#) Selected indexes [« clear all](#)

- All non-internal indexes
- All internal indexes
- _audit
- _internal
- _introspection
- _thefishbucket
- add_on_builder_index
- custom_index
- history
- main

main

Custom_index is not searched by default for this user.





There are an exceptions to every rule

Exceptions to using the “main” Index

- Testing – writing data to a test index during development allows the developer to quickly and easily clear out all events in the index without impacting other events elsewhere.

```
$SPLUNK_HOME/bin/splunk clean eventdata custom_index
```

- Retention – data retention/aging is controlled on the index level. Some administrators may want to have custom retention policies based on the type of data.
- Security – using Splunk’s RBAC, the administrator can control who sees what data.

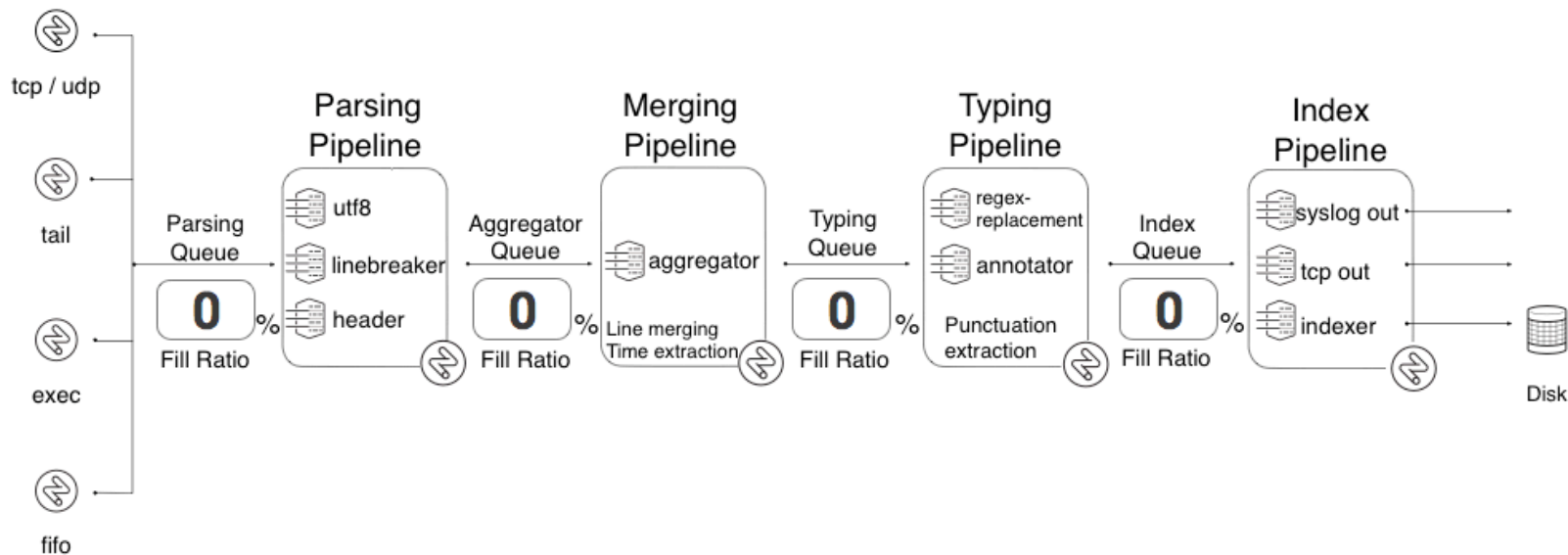
Exceptions to using the “main” Index

- Testing – writing data to a test index during development allows the developer to quickly and easily clear out all events in the index without impacting other events elsewhere
- Retention – data retention/aging is controlled on the index level. Some administrators may want to have custom retention policies based on the type of data.
- Security – using Splunk’s RBAC, the administrator can control who sees what data.



The last 2 exception decisions should be made by the Splunk admin – not the developer.

Get to Know Your Pipelines



Useful Index Time Processing Attributes

Event Breaking

LINE_BREAKER <where to break the stream>
SHOULD_LINEMERGE <enable/disable merging>

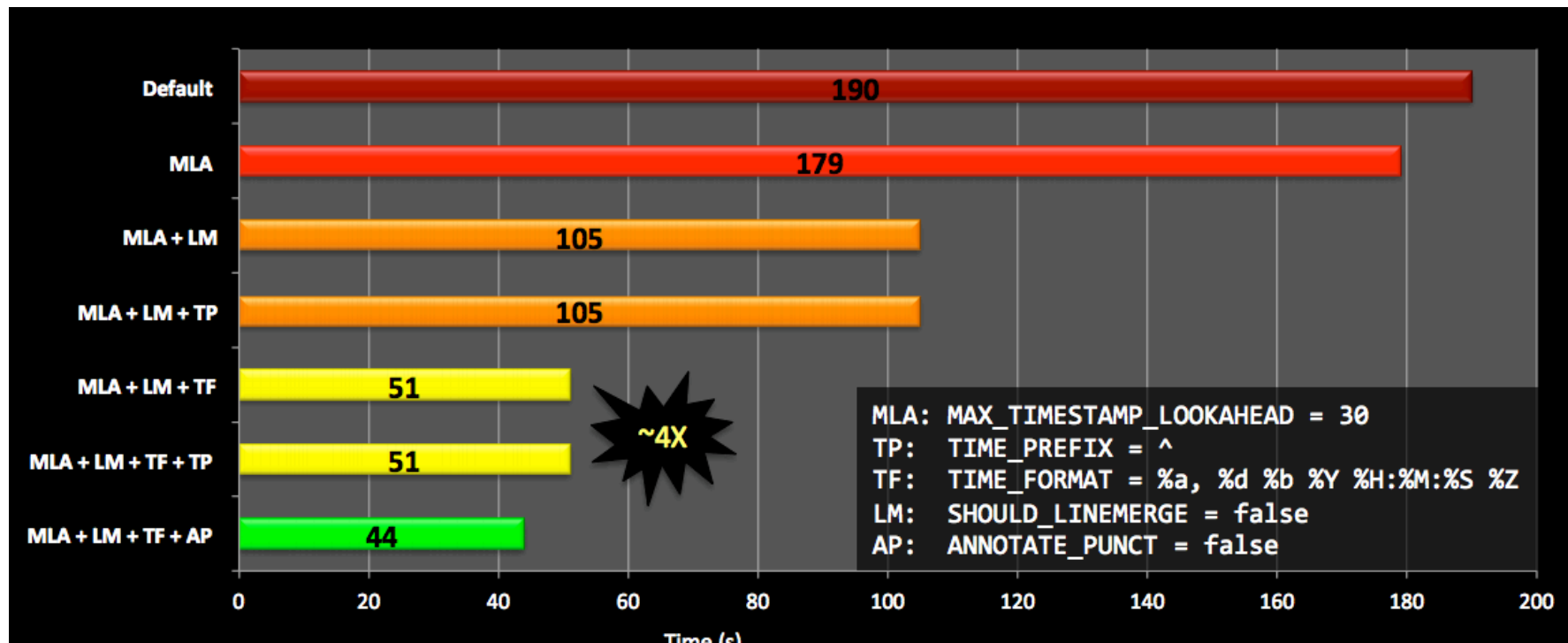
Timestamp Extraction

MAX_TIMESTAMP_LOOKAHEAD <# chars in to look for ts>
TIME_PREFIX <pattern before ts>
TIME_FORMAT <strftime format string to extract ts>

Typing

ANNOTATE_PUNCT <enable/disable punct:: extraction>

Useful Index Time Processing Attributes



HT: Dritan Bitincka

Adding Inputs

.conf2016

splunk>

Scripted versus Modular Inputs

Feature	Scripted Inputs	Modular Inputs
End user configuration	Inline arguments Often requires editing text configuration files	User interface provided in the Splunk Web interface. This makes the input “look and feel” as if it were a native Splunk feature.
Multi-platform support	No	Yes You can package your script to include versions for separate platforms.
Custom REST endpoints	No	Yes Modular inputs can be access and manipulated using Splunk REST endpoints.
Endpoint permissions	N/A	Access implemented using Splunk Enterprise capabilities.

More complete information can be found on the [Splunk documentation page](#).

Scripted versus Modular Inputs

Scripted inputs are more suited for trivial tasks such as running an OS command (like ***top*** for *nix or ***Get-Process*** from Windows PowerShell) and sending the output to Splunk.

Modular inputs are more suited for tasks that require end user setup or more advanced event processing. Calling a REST API with parameters is a good example of when to use a modular input.

This or...

```
script://./bin/zenoss_wrapper.sh -u admin -p password -a h8p://  
zenoss:8080 -z America/Los_Angeles -t 4320 -r 90 -s  
2015-03-16T00:00:00 -index-closed-events 1 -index-cleared-events 1 -  
index-archived-events 1 -index-suppressed-events 1 -index-  
repeatevents 1]
```

sourcetype = zenoss-events

interval = 60

index = zenoss

splunk> Apps ▾ Administrator ▾ Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

zenoss

[Data Inputs](#) » [Zenoss Events](#) » zenoss

Username *

Zenoss Username

Password *

Password

Confirm password

Zenoss Web Interface *

Zenoss web interface address; e.g. http://zenoss-server:8080

Device Name

Optional: Specify a device to pull events from or leave blank for all devices.

Timezone

Timezone of Zenoss server. Defaults to local time of this Splunk server if left blank

Archive Threshold (minutes)

Zenoss 'Event Archive Threshold (minutes)' setting. Interval to read archive table. Leave blank for Zenoss default of 4320.

Event Checkpoint Removal (days)

Zenoss 'Delete Archived Events Older Than (days)' setting. Used to keep checkpoint file clean. Leave blank for Zenoss default of 90.

Start Date

Optional: Specify a starting date to pull events from or leave blank for ALL events. Ex: 2015-03-16T00:00:00

☒ **Index Closed Events**
Optional: Index eventState "Closed"

☒ **Index Cleared Events**
Optional: Index eventState "Cleared"

☒ **Index Archived Events**
Optional: Index events from the Archive table

Scripted and Modular Input Best Practices

Do not hard code paths

Example (Python):

```
os.path.join(os.environ["SPLUNK_HOME"], 'etc', 'apps', APP_NAME)
```

Example (PowerShell):

```
Join-Path -path (get-item env:\SPLUNK_HOME).value "Splunk\etc  
\apps"
```

Scripted and Modular Input Best Practices

Use Error Trapping (so that you can search them in the _internal index)

```
import logging
try:
    Some code that may fail like opening a
    file
except IOError, err:
    logging.error(' %s - ERROR - File may not
    exist %s\n' % (time.strftime("%Y-%m-%d %H:%M:
    %S"), str(err)))
    pass
```

Scripted and Modular Input Best Practices

Error Trapping (you can use stderr too)

```
try:
    Some code that may fail like opening a
    file

except IOError, err:
    sys.stderr.write(' %s - ERROR - File may
not exist %s\n' % (time.strftime("%Y-%m-%d %H:
%M:%S"), str(err)))
    pass
```

Scripted and Modular Input Best Practices

Use Splunk methods to read cascaded settings

Example (Python):

```
import splunk.clilib.cli_common

def __init__(self, obj):
    self.object = obj
    self.settings =
splunk.clilib.cli_common.getConfStanza("acme",
"default")
```

- Give more explanation on previous slide
- Mention someone trying to read from default and write to local
- Maybe mention btool too

Scripted and Modular Input Best Practices

Disable any inputs by default

inputs.conf:

```
[my_stanza]  
disabled = 1
```

Scripted Inputs Best Practices

Test Scripts using Splunk CMD

Mac:

```
/Applications/Splunk/bin/splunk cmd python /Applications/  
Splunk/etc/apps/<your app>/bin/<your script>
```

Windows:

```
C:\Program Files\Splunk\bin\splunk.exe cmd C:\Program Files  
\Splunk\etc\apps\<your app>\bin\<your script>
```


Modular Inputs Best Practices

Use Splunk SDKs (these abstract a lot of code for you)

Python <http://dev.splunk.com/view/python-sdk/SP-CAAER3>

C# <http://dev.splunk.com/view/csharp-sdk/SP-CAAER4>

Java <http://dev.splunk.com/view/java-sdk/SP-CAAER2>

Modular Input SDKs

Before = 453 lines

```
438     return val_data
439
440 if __name__ == '__main__':
441
442     if len(sys.argv) > 1:
443         if sys.argv[1] == "--scheme":
444             do_scheme()
445         elif sys.argv[1] == "--validate-ar
446             do_validate()
447         else:
448             usage()
449     else:
450         do_run()
451
452 sys.exit(0)
453
```

After = 92 lines

```
85
86 except Exception as e:
87     raise e
88
89 if __name__ == "__main__":
90     exitcode = MyScript().run(sys.argv)
91     sys.exit(exitcode)
92
```

Modular Input SDK Logging

By default, only INFO and higher events are logged to `_internal`.

Server logging	
Server settings » Server logging	
modular	
Showing 1-3 of 3 items	
Log channel ↕	Logging level ↕
AdminHandler:ModularInputs	WARN
ModularInputs	INFO
ModularUtility	WARN



Modular Inputs Best Practices

Validate User Input

```
# Try to connect to the Azure API to validate the given arguments
work
try:
    access_token = get_token_from_client_credentials(
        endpoint = val_data["token_endpoint"],
        client_id = val_data["client_id"],
        client_secret = val_data["client_secret"])
except Exception, e:
    raise Exception, "Could not connect to the Azure REST endpoint.
Check the token endpoint, client ID, and client secret/key: %s" %
str(e)
```

Modular Inputs Best Practices

Validate User Input

Encountered the following error while trying to save: In handler 'AzureAudit': Invalid configuration specified: Could not connect to the Azure REST endpoint. Check the token endpoint, client ID, and client secret/key: Invalid URL 'blah': No schema supplied. Perhaps you meant http://blah?

Azure Audit Input Name *

Name of this Azure Audit Input

Azure Subscription ID *

To ingest data for more than one subscription, create a new Azure Audit input with other subscription IDs.

The OAuth 2.0 Token Endpoint of the Azure AD application *

Modular Inputs Best Practices

Test Inputs using Splunk CMD

Example (real):

```
/Applications/Splunk/bin/splunk cmd splunkd print-modinput-  
config AzureDiagnostics AzureDiagnostics://gsa1892 | /  
Applications/Splunk/bin/splunk cmd python /Applications/  
Splunk/etc/apps/TA_Azure/bin/AzureDiagnostics.py
```

Modular Inputs Best Practices

Test Inputs using Splunk CMD

Example (real):

Name of the input

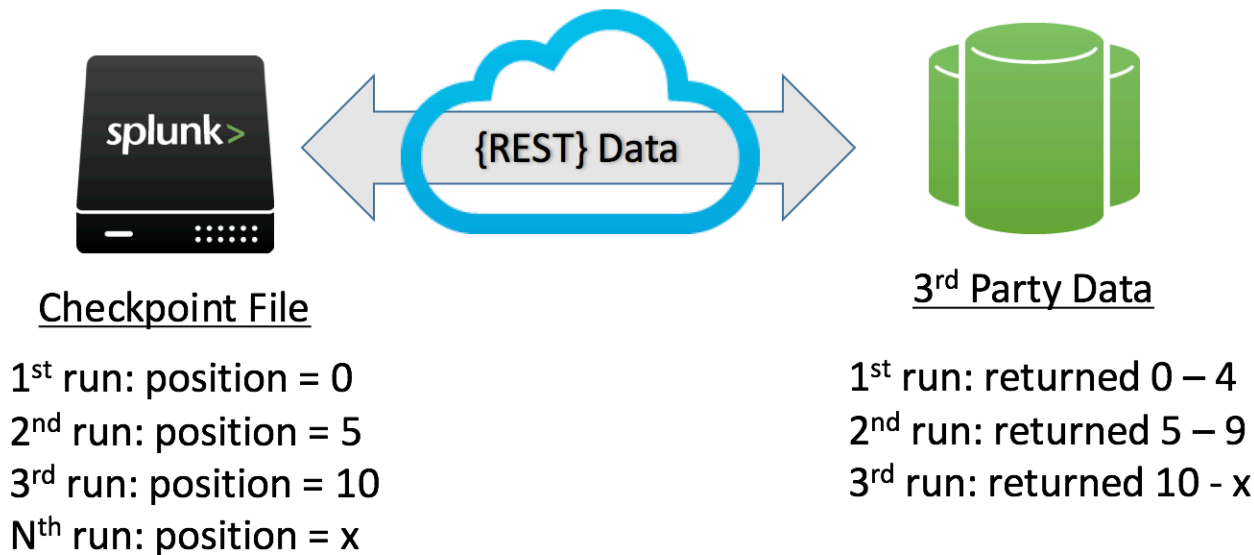
Instance of the input

```
/Applications/Splunk/bin/splunk cmd splunkd print-modinput-  
config AzureDiagnostics AzureDiagnostics://gsa1892 | /  
Applications/Splunk/bin/splunk cmd python /Applications/  
Splunk/etc/apps/TA_Azure/bin/AzureDiagnostics.py
```

Input code

Modular Inputs Best Practices

Use the checkpoint parameter to persist data



<http://blogs.splunk.com/2016/05/11/splunking-continuous-rest-data/>

Eventgen

YouTube

github.com/splunk/eventgen

splunk / eventgen

Code Network Pull Requests Issues Wiki Graphs Admin

Splunk Event Generator — Read more

Clone in Mac ZIP HTTP SSH https://github.com/splunk/eventgen.git

branch: master Files Commits Branches Tags Downloads

Latest commit to the master branch

Fixed bug with replay mode and getting the proper index, host, source... [coocyx]

coocyx authored 15 minutes ago

name	age	message	history
README	a day ago	Completed build script [coocyx]	
bin	15 minutes ago	Fixed bug with replay mode and getting the proper index, host, source... [coocyx]	
default	15 minutes ago	Fixed bug with replay mode and getting the proper index, host, source... [coocyx]	
lib	15 minutes ago	Fixed bug with replay mode and getting the proper index, host, source... [coocyx]	
local	15 minutes ago	Fixed bug with replay mode and getting the proper index, host, source... [coocyx]	

Eventgen

Clint Sharp

Subscribe 9

1,720

Add to Share More

6 0

This repository Search

Explore Gist Blog Help

Jason

splunk / eventgen

Watch 33

Splunk Event Generator

184 commits 4 branches 1 release 4 contributors

branch: master eventgen / +

Trying to fix conflict with oidemo

Clint Sharp authored on Feb 13 latest commit dadf4148b1

README	Added documentation for timeField	9 months ago
bin	Moving back to threading	8 months ago
default	Trying to fix conflict with oidemo	7 months ago
lib	more debugging when splunk errors out in splunkstream	8 months ago
local	Trying to fix conflict with oidemo	7 months ago
metadata	Fixed bug with flattening that was failing regressions for ES. Had to...	2 years ago
samples	Removed mobile useragents	11 months ago
.gitignore	Updated build system to use ant	a year ago

Anonymize Eventgen Samples

Regex Find and Replace Tools are Your Friend!

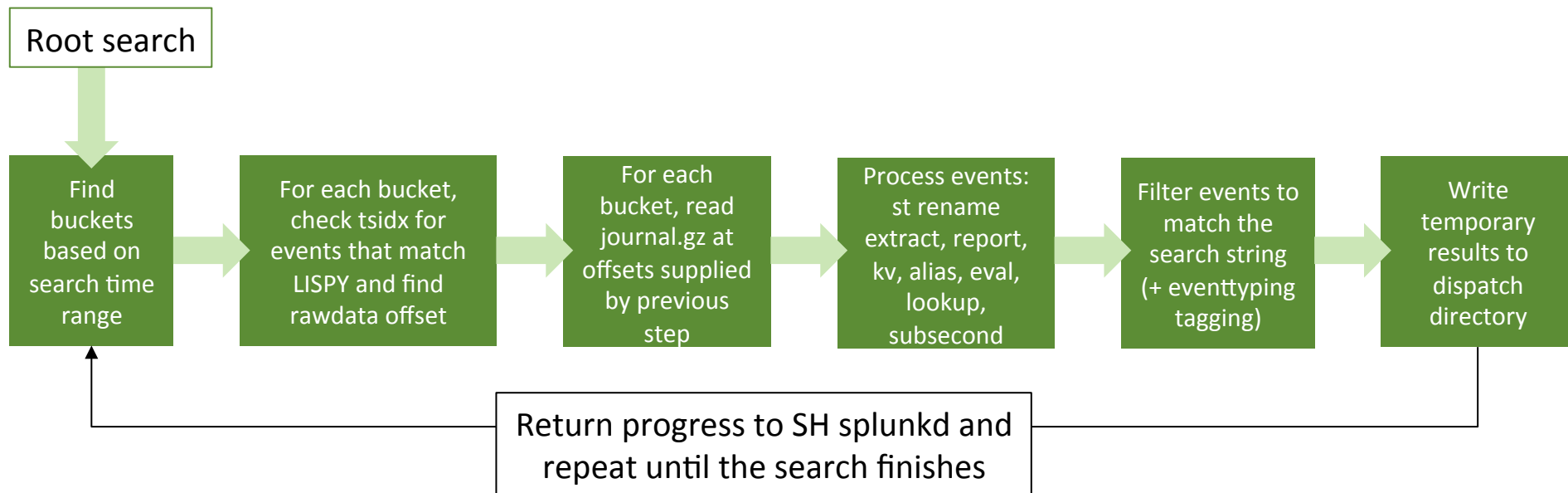


3

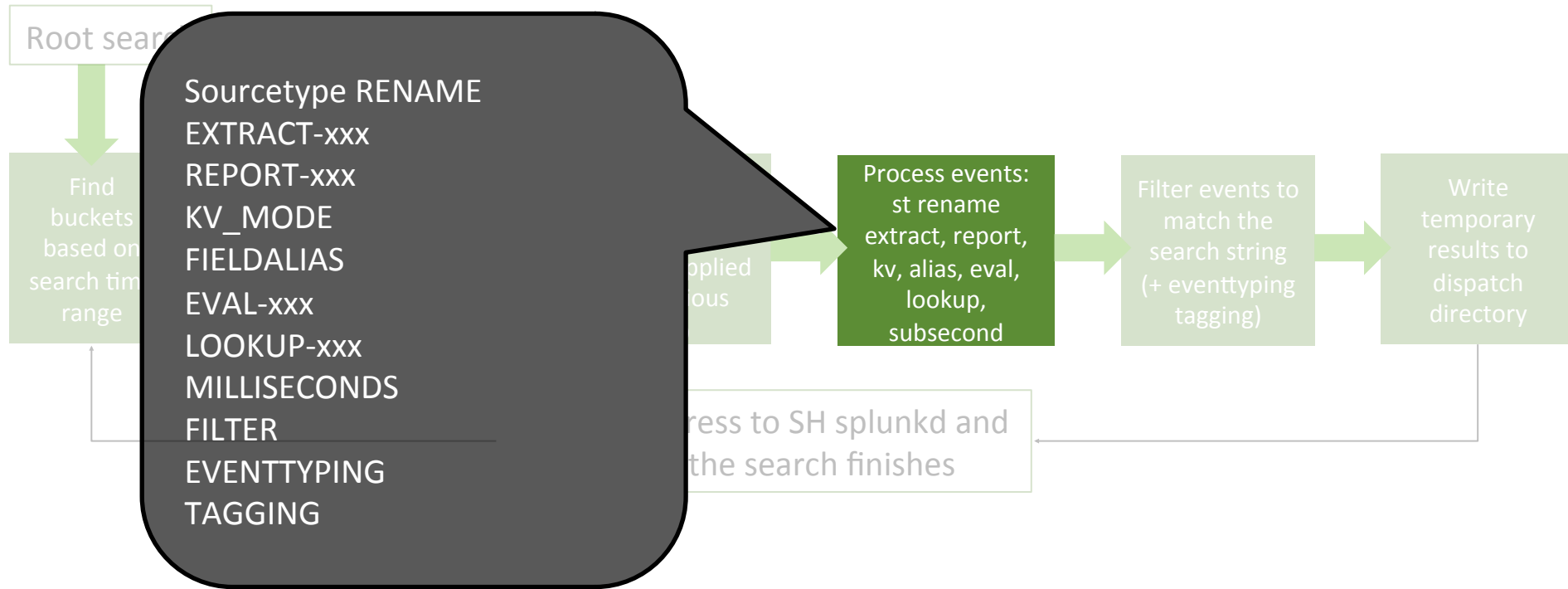
Asking Questions of your Data

.conf2016

Get to Know the Search Pipeline



Get to Know the Order of Operations



Parameterize Root Searches

macros.conf example:

```
[acme_index]  
definition = index=acme
```

Example search using macro:

```
`acme_index` sourcetype=widiget |  
stats count
```

Remember that main index thing earlier?

Get to Know Distributed Search

macros.conf

[my_index]

definition = index=main

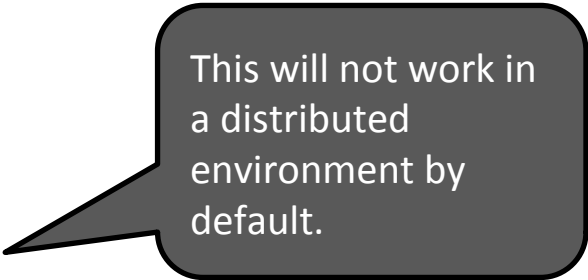
eventtypes.conf

[my_eventtype]

search = `my_index` sourcetype="foo"

Example search:

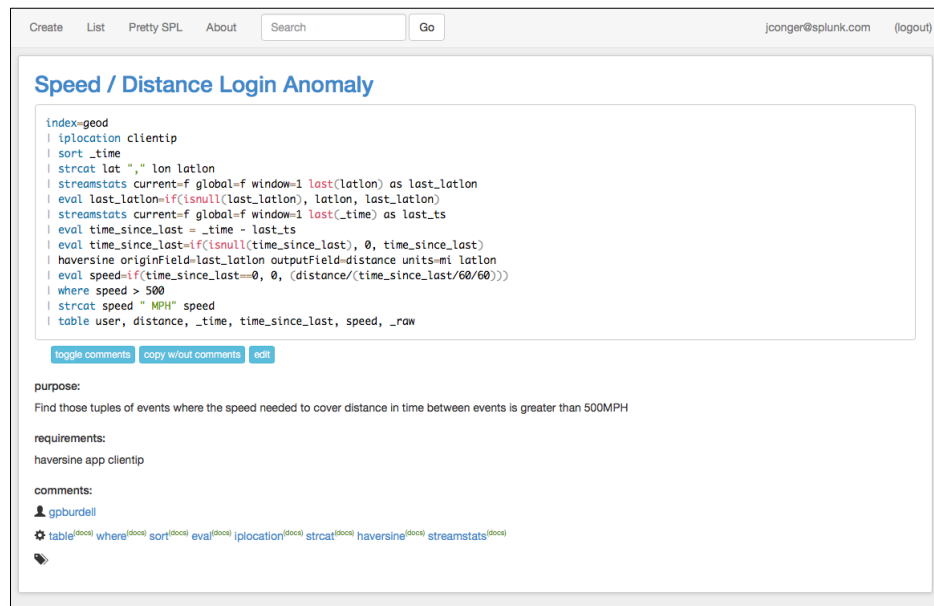
eventtype=my_eventtype | stats count



This will not work in a distributed environment by default.

Get to Know the Big Book of Search

www.bbosearch.com



The screenshot shows the Splunk Search interface. At the top, there are tabs for 'Create', 'List', 'Pretty SPL', and 'About', along with a search bar and a 'Go' button. The user is logged in as 'jconger@splunk.com'. The search title is 'Speed / Distance Login Anomaly'.

```
index=geod
| iplocation clientip
| sort _time
| strcat lat "," lon latlon
| streamstats current=f global=f window=1 last(latlon) as last_latlon
| eval last_latlon=if(isnull(last_latlon), latlon, last_latlon)
| streamstats current=f global=f window=1 last(_time) as last_ts
| eval time_since_last = _time - last_ts
| eval time_since_last=if(isnull(time_since_last), 0, time_since_last)
| haversine originField=last_latlon outputField=distance units=mi latlon
| eval speed=if(time_since_last=0, 0, (distance/(time_since_last/60/60)))
| where speed > 500
| strcat speed " MPH" speed
| table user, distance, _time, time_since_last, speed, _raw
```

Below the search bar, there are buttons for 'toggle comments', 'copy w/out comments', and 'edit'.

purpose:
Find those tuples of events where the speed needed to cover distance in time between events is greater than 500MPH

requirements:
haversine app clientip

comments:
gpburdell

gear icon table where sort eval iplocation strcat haversine streamstats

Include Prebuilt Panels

Even if it just to verify the thing is working

Sourcetype Counts

Edit ▾ More Info ▾ ⬇️ 🖨️

If you see data here, it is working...

Sourcetype Counts

	sourcetype ↕	count ↕
1	eventgen	113
2	eventgen_metrics	5900
3	mongod	52
4	scheduler	3
5	splunk_app_addon-builder_validation_mi-2	81
6	splunk_ta_snow_ticket	351
7	splunk_web_access	2768
8	splunk_web_service	322
9	splunkd	1385
10	splunkd_access	94

« prev 1 2 next »

Use the Dashboards Example App

Overview Examples Dashboards Search Splunk 6.x

Examples

Edit ▾

Basic Elements

- Chart Elements
- Table Elements
- Single Value Elements
- Map Elements
- Search Types
- Form Input Elements
- Drilldown Elements
- Layout Elements
- Custom Visualizations
- Token Customization

Basic Elements

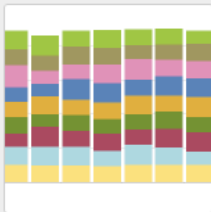
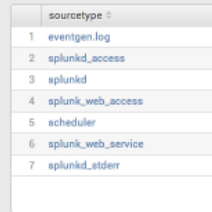


Chart Element

Add graphs, charts, and gauges to dashboards.

6.2 6.3 6.4

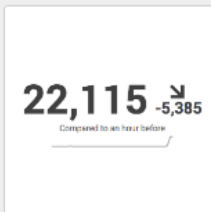


source type	count
eventgen.log	1
splunkd_access	2
splunkd	3
splunk_web_access	4
scheduler	5
splunk_web_service	6
splunkd_stderr	7

Table Element

Create a simple table using the dashboard editor.

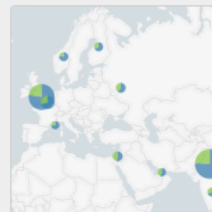
6.2 6.3 6.4



Single Value Element

Demonstrate a single value element with basic drilldown and rangemap configurations.

6.2 6.3 6.4



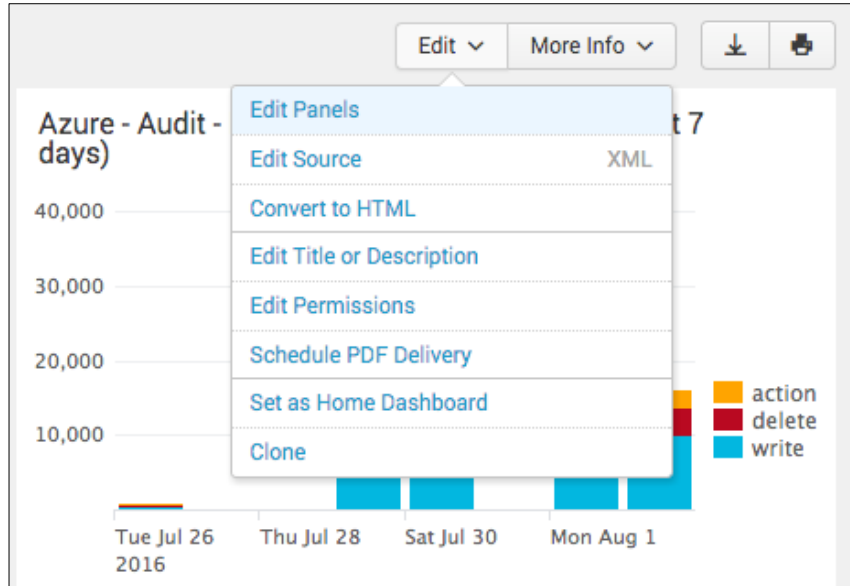
Map Element

Plot geographical data on integrated maps.

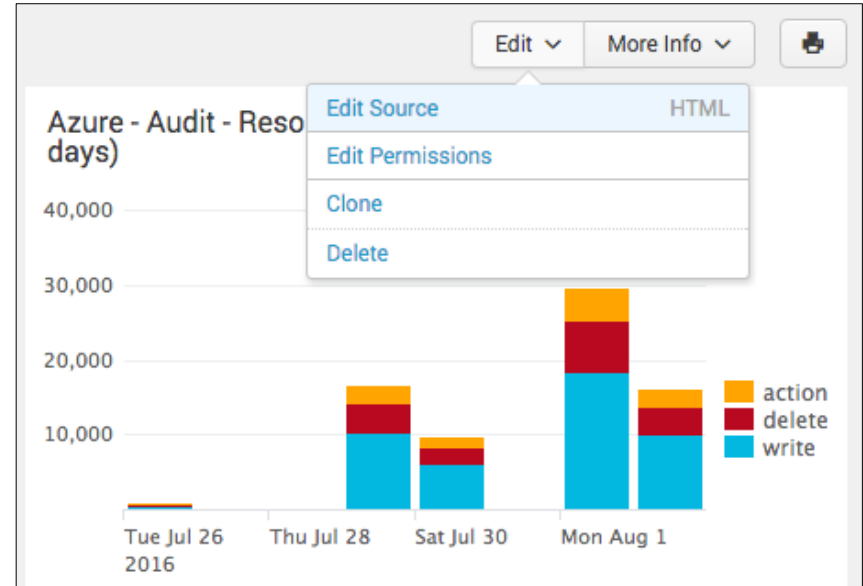
6.2 6.3 6.4

<https://splunkbase.splunk.com/app/1603/>

Use Simple XML as much as possible

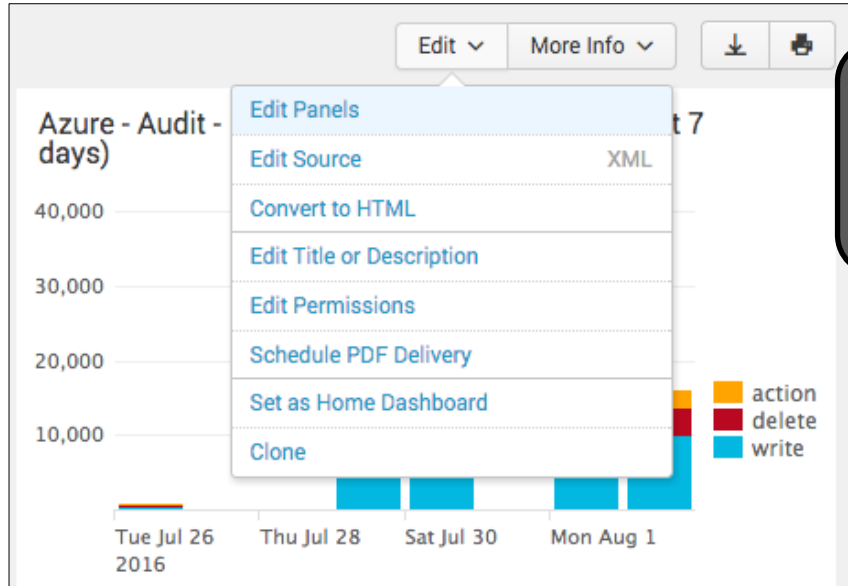


Simple XML



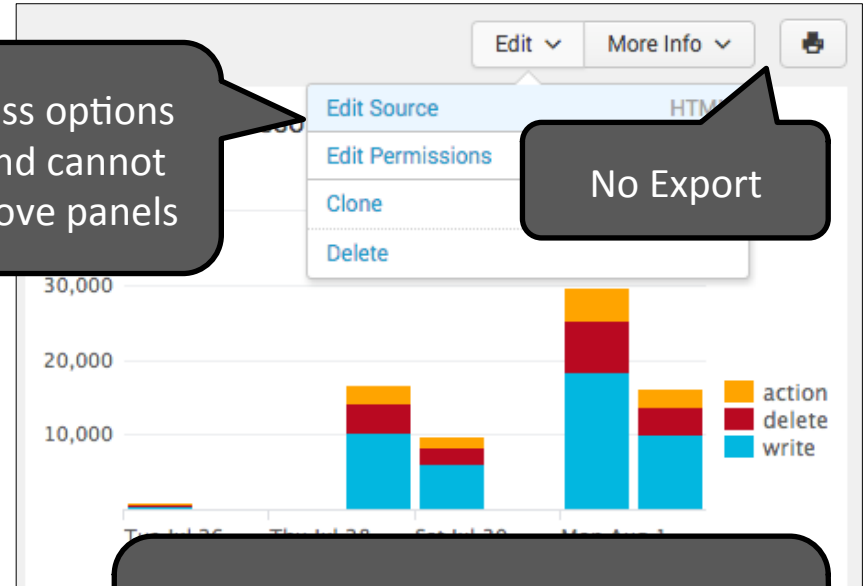
HTML

Use Simple XML as much as possible



Simple XML

Less options
and cannot
move panels



No Export

But, it is HTML and you have
complete control of the look and
feel.

Get to know JavaScript and jQuery

The screenshot shows the 'Splunk 6.x Dashboard Examples' app interface. At the top, there are navigation tabs: 'Overview', 'Examples', 'Dashboards', and 'Search'. The main title is 'Table Row Expansion (More Details)'. To the right of the title are buttons for 'Edit', 'More Info', and download/print icons. Below the title is a table with two columns: 'sourcetype' and 'count'. The table lists various sourcetypes and their corresponding counts. The 'splunkd_access' row is highlighted. Below the table is a 'Description' section with text explaining row expansion. At the bottom, there is a 'Source Code' section with tabs for 'custom_table_row_expansion.xml' and 'custom_table_row_expansion.js'. The 'custom_table_row_expansion.js' tab is active, showing the start of a JavaScript file with the line '1. require(['

i	sourcetype	count
>	eventgen	3605
>	eventgen_metrics	80072
>	mongod	82
>	scheduler	27
>	splunk_app_addon-builder_validation_mi-2	81
>	splunk_ta_snow_ticket	15992
>	splunk_web_access	17666
>	splunk_web_service	
>	splunkd	
>	splunkd_access	

Description

With tables you can specify content to be shown when a user expands a row. This example shows a basic renderer that shows the content for the value that was expanded. Only one row can be expanded at a time.

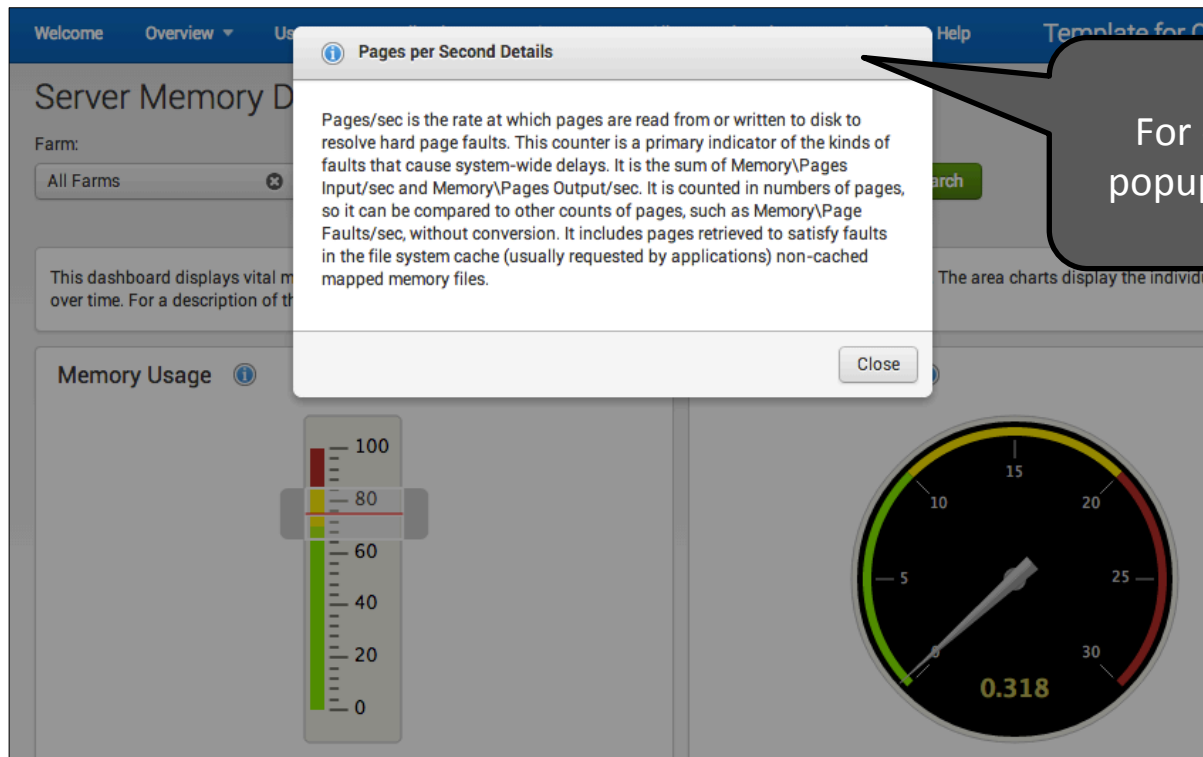
Source Code

- custom_table_row_expansion.xml
- custom_table_row_expansion.js

```
1. require([
```

The Dashboard Examples app has some great relevant code and ideas.

Bootstrap can add functionality



For example, easily add a modal popup to a Simple XML dashboard!

Get to know CSS

- All Splunk elements have an id now
- Check out Firefox's 3D view for layering



Splunk Cloud has Best Practices too

The screenshot shows the Splunk Cloud developer portal. At the top, there's a navigation bar with 'splunk>dev' logo, links for 'Get Started', 'Web Framework', 'REST API', 'SDKs', 'Tools', and 'Developer License', a search bar, and a 'FREE SPLUNK' button. Below the navigation bar, there's a breadcrumb trail: 'Overview | Developer Guidance | Integrate and Extend | App Certification'. The main heading is 'Splunk Cloud app requirements and best practices'. The content includes an introduction paragraph, a paragraph about following recommended practices, and an 'Important' callout box. A sidebar on the right titled 'APP CERTIFICATION' lists various links. At the bottom, there's a list of sections contained in the topic.

Splunk.com | Community | Login

splunk>dev

Get Started | Web Framework | REST API | SDKs | Tools | Developer License

FREE SPLUNK

Overview | Developer Guidance | Integrate and Extend | App Certification

Splunk Cloud app requirements and best practices

For an app to be installed in Splunk Cloud, it must meet the requirements specified in the first two sections of this topic. The third section lists several highly recommended, but not required, actions.

Whenever possible, you should follow the recommended practices for the Splunk App Certification Program. Splunk-certified apps are automatically approved for installation in Splunk Cloud. For more information, see [About app certification](#).

Important: Splunk Cloud app developers and users of Splunk Cloud apps assume responsibility for ensuring proper usage of any third-party services that they choose to use in connection with Splunk Cloud, including compliance with any relevant terms and licenses. As a reminder and pursuant to [Splunk Cloud Terms of Service](#), Splunk is not liable for any problems that might arise from sending data to those third-party services (including, without limitation, any disclosure, modification or deletion of data resulting from access to such third-party services) and does not provide any support for those services.

This topic contains the following sections:

- [Required behaviors](#)
- [Prohibited behaviors](#)

APP CERTIFICATION

- About app certification
- App certification process
- Splunk Cloud best practices**
- Security best practices
- App certification criteria
- Resources and helpful links
- Submit an app or add-on for certification
- Access download leads

<http://dev.splunk.com/view/app-cert/SP-CAAEE85>

Do's and Don'ts – Packaging Applications

Do	Don't
Follow the guidelines found at https://docs.splunk.com/Documentation/Splunk/7.2.5/GettingStartedWithSplunk/PackageApps	Leave any hidden files in the app such as Mac's ._ files.
Include a screen shot of your application in the correct location.	
Let the user choose which inputs are enabled for their environment.	Enable all inputs by default if not necessary.
Use a build automation tool such as Apache Ant if necessary to ensure a clean build/package.	Leave anything in: \$SPLUNK_HOME/etc/apps/<app>/local directory \$SPLUNK_HOME/etc/apps/<app>/metadata/local.meta
Ensure the appropriate settings are set in app.conf	
Document your app with a README.txt file	
Test your application on a clean system	

Do's and Don'ts – Data Collection

Do	Don't
Support multiple platforms.	Code for a single OS.
Use scripting language utilities such as <code>os.path.join()</code> and the special environment variable <code>\$(SPLUNK_HOME)</code> to construct paths in scripts.	Hard code script paths.
Write data to the “main” index. This ensures that your data is searchable by default.	Hard code index names in searches if you must use a custom index.
Use key=value pairs in writing to log files (if you have control of the logging output).	Use name abbreviations.
Throttle how much data is collected at one time from an API.	Overwhelm a system by pulling exorbitant amounts of data at one time from an API.
Use logging and error trapping in scripts and inputs.	

Do's and Don'ts - Applications

Do	Don't
Use setup.xml or a mod input configs to allow the end user to configure the app	Make users manually enter information such as API credentials into configuration files.
Encrypt user input passwords. https://docs.splunk.com/Documentation/Splunk/7.0.2/InstallingApps/EncryptPasswords	Store clear text passwords in .conf files.
Parameterize indexes so that they can be easily changed	Hard code indexes in your searches
Use the CIM add-on https://docs.splunk.com/Documentation/Splunk/7.0.2/InstallingApps/CIM/InstallCIMAddOn	
Place all .conf files in default \$SPLUNK_HOME/etc/apps/<your_app>/default	Leave any content in \$SPLUNK_HOME/etc/apps/<your_app>/local
Set default permissions in: \$SPLUNK_HOME/etc/apps/<your_app>/metadata/default.meta	Have a local.meta file located in: \$SPLUNK_HOME/etc/apps/<your_app>/metadata

THANK YOU

.conf2016