

.conf2014

# YOUR DATA ADVENTURE

Speeding up Dashboards  
with Pivot

Rupak Pandya | OI Practice  
Manager, Function1



FUNCTION1

[www.function1.com](http://www.function1.com)

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.

# Agenda

- Background
- Splunk Search
- Overview of Data Models
- Overview of the Pivot Search Command
- Demonstration

# Meet Function1



- Founded in 2007, Function1 is an enterprise technology solution firm and has been a Preferred Splunk Partner since 2011
- We have 11 consultants in our Operational Intelligence group that specialize in delivering Splunk Professional Services

# Meet Rupak Pandya

- Practice Manager of the Function1 OI Group
- Joined the Operational Intelligence team as an experienced consultant from a large, global consulting company
- Avid follower of all of the Washington, D.C. area sports teams



*rupak@function1.com*  
*301.452.2475 ext 24*

# Anatomy of a Splunk Search

```
index=_internal sourcetype=splunkd group=per_sourcetype_thruput |  
eval host_series = host + "_" + series | stats sum(ev) by  
host_series | rename sum(ev) AS "Total Events" | sort - "Total  
Events"
```

- > Search And Filter
- > Enrich
- > Report
- > Format

# Challenges

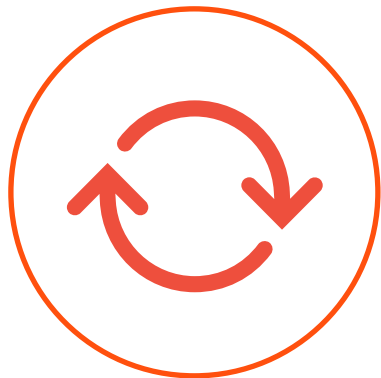
```
index=warum sourcetype="ri:pas:application" | lookup employee_details empusername as user
OUTPUT empshiftstarttime empshiftendtime empshiftdays | eval hour =
tonumber(strftime(_time,"%H")) | eval overnight = if(empshiftstarttime>empshiftendtime,1,0) | eval
valid_time=if((overnight==1 AND (hour >= empshiftstarttime OR hour <= empshiftendtime)) OR
(overnight==0 AND hour>=empshiftstarttime AND hour<empshiftendtime),1,0) | eval
shiftday=if(overnight==1 AND hour < empshiftendtime,relative_time(_time,"-1d"),_time) | eval
weekday = strftime(_time,"%a") | eval shiftday=strftime(shiftday,"%a") | eval
valid_day=if(match(empshiftdays,shiftday),1,0) | search valid_time=0 OR valid_day=0 | bucket _time
span=5m | stats count by _time user command object
```

- Complicated searches get very verbose
- Searchers need to understand data's structure
- Non-technical users might not have knowledge of underlying data
- Splunk admins do not always know what users will be searching on

# There Has to be a Better Way...



# Data Model Goals



Make it easy to  
share/reuse  
domain knowledge



Admins/power users  
build data models



Non-technical  
users interact with  
data via pivot UI

.conf2014

# YOUR DATA ADVENTURE

Data Models

splunk>

# What are Data Models?

The screenshot displays the Splunk Web Intelligence interface. On the left, a tree view under 'Objects' shows a hierarchy: 'HTTP\_Request' (selected) with sub-items like 'ApacheAccessSearch', 'ISAAccessSearch', 'HTTP\_Success', 'Pageview', 'AssetAccess', 'DocAccess', 'MediaAccess', and 'PodcastDownload'. Below this are 'SEARCHES' (User) and 'TRANSACTIONS' (WebSession). The main panel shows the configuration for the 'HTTP\_Request' data model. It includes a 'CONSTRAINTS' section with a single constraint: 'sourcetype=access\_\* OR sourcetype=!\*'. Below that is an 'ATTRIBUTES' table. The table lists attributes, their types, and whether they are required. Some attributes are marked as 'Auto-Extracted'. A note at the bottom states: 'Evals, Lookups, Regexp and Geo IPs are processed in the order above, so ensure any dependent attributes are defined first. Drag to rearrange.'

ATTRIBUTES			
Inherited			
_time	timestamp	View Values	
host	string	View Values	
source	string	View Values	
sourcetype	string	View Values	
uri	string (required)	View Values	
uri_path	string (required)	View Values	
status	number (required)	View Values	
clientip	ipv4 (required)	View Values	
referrer	string (required)	View Values	
useragent	string (required)	View Values	

- Hierarchically structured search-time mapping of semantic knowledge about one or more datasets
  - Fields that data models use are called attributes
  - To create an effective data model, you must understand your data sources and your data semantics

# A Data Model is a Collection of Objects

The screenshot shows the Splunk Web Intelligence interface. On the left, a tree view under 'Objects' shows a hierarchy of data models: HTTP\_Request (circled in red), ApacheAccessSearch, IISAccessSearch, HTTP\_Success, Pageview, AssetAccess, DocAccess, MediaAccess, PodcastDownload, and HTTP\_Error. The main panel displays details for the 'HTTP\_Request' object, including its constraints and attributes.

**HTTP\_Request**  
HTTP\_Request

Constraints

Constraint	Edit	Constraint	View Events
		sourcetype=access_* OR sourcetype=iis*	

Attributes

Inherited	Attribute	Type	View Values
	_time	timestamp	
	host	string	
	source	string	
	sourcetype	string	
Auto-Extracted	uri	string (required)	
Auto-Extracted	uri_path	string (required)	
Auto-Extracted	status	number (required)	

# Objects Have Constraints and Attributes

The screenshot shows the Splunk Web Intelligence interface. The top navigation bar includes 'splunk>', 'App: Search & Reporting', 'System', '2 Messages', 'Triggered Alerts', 'Jobs', and 'Help'. The main header displays 'Web Intelligence' with a 'Documentation' link and a 'Back to Data Models' link.

The 'Objects' section on the left contains a tree view with the following items: ApacheAccessSearch, IISAccessSearch, HTTP\_Success, Pageview, AssetAccess, DocAccess, MediaAccess, PodcastDownload, and HTTP\_Error. An 'Add Object' button is located at the top of this section.

The main content area displays the configuration for the 'HTTP\_Request' object. It is divided into two sections:

- CONSTRAINTS:** A table with one row showing a constraint on 'sourcetype'. The constraint is 'sourcetype=access\_\* OR sourcetype=iis\*'. There are 'Edit' and 'View Events' links for this constraint.
- ATTRIBUTES:** A table listing attributes for the object. It includes inherited attributes like '\_time' (timestamp), 'host' (string), 'source' (string), and 'sourcetype' (string). It also lists 'Auto-Extracted' attributes: 'uri' (string, required), 'uri\_path' (string, required), and 'status' (number, required). Each attribute has 'Edit', 'Delete', and 'View Values' links.

Two red ovals are drawn over the 'CONSTRAINTS' and 'ATTRIBUTES' sections to highlight them.

# Child Objects Inherit Constraints and Attributes

The screenshot shows the Splunk Web Intelligence interface. On the left, a tree view of objects is shown under 'EVENTS'. The 'HTTP\_Request' object is selected, and red arrows point to its child objects: 'ApacheAccessSearch', 'IISAccessSearch', 'HTTP\_Success', 'Pageview', 'AssetAccess', 'DocAccess', 'MediaAccess', and 'PodcastDownload'. The main panel displays the configuration for the 'HTTP\_Request' object, including its constraints and attributes.

**Web Intelligence**  
WebIntelligence [Documentation](#)

[Back to Data Models](#)

**Objects** Add Object

**EVENTS**

- HTTP\_Request
  - ApacheAccessSearch
  - IISAccessSearch
  - HTTP\_Success
  - Pageview
  - AssetAccess
  - DocAccess
  - MediaAccess
  - PodcastDownload
- HTTP\_Error

**HTTP\_Request**  
HTTP\_Request Rename Delete

**CONSTRAINTS**

Constraint	Edit	Constraint	View Events
		sourcetype=access_* OR sourcetype=iis*	<a href="#">View Events</a>

**ATTRIBUTES** Add Attribute

Inherited	Attribute	Type	View Values
	_time	timestamp	<a href="#">View Values</a>
	host	string	<a href="#">View Values</a>
	source	string	<a href="#">View Values</a>
	sourcetype	string	<a href="#">View Values</a>
Auto-Extracted	uri	string (required)	<a href="#">View Values</a>
Auto-Extracted	uri_path	string (required)	<a href="#">View Values</a>
Auto-Extracted	status	number (required)	<a href="#">View Values</a>

# Child Objects Inherit Constraints and Attributes

The screenshot shows the Splunk Web Intelligence interface. The main content area displays the 'HTTP\_Success' object. On the left, a tree view shows the object hierarchy under 'EVENTS', including 'HTTP\_Request' and 'HTTP\_Success'. The 'HTTP\_Success' object is selected, and its details are shown on the right. The 'CONSTRAINTS' section lists two constraints: 'Inherited' with the constraint 'sourcetype=access\_\* OR sourcetype=iis\*' and 'Constraint' with the constraint 'status = 2\*'. The 'ATTRIBUTES' section lists ten attributes, all of which are 'Inherited'. The first attribute, 'timestamp', is highlighted with a red box. The 'ATTRIBUTES' table is as follows:

Attribute	Visibility	Required	Type	Value	View Values
Inherited	Hide	Make Required	_time	timestamp	View Values
Inherited	Hide	Make Required	host	a string	View Values
Inherited	Hide	Make Required	source	a string	View Values
Inherited	Hide	Make Required	sourcetype	a string	View Values
Inherited	Hide	Make Optional	uri	a string	View Values
Inherited	Hide	Make Optional	uri_path	a string	View Values
Inherited	Hide	Make Optional	status	# number	View Values
Inherited	Hide	Make Optional	clientip	a ipv4	View Values
Inherited	Hide	Make Optional	referer	a string	View Values
Inherited	Hide	Make Optional	useragent	a string	View Values

.conf2014

# YOUR DATA ADVENTURE

Building Data  
Models

splunk>

# Three Root Object Types

The screenshot displays the Splunk Web Intelligence interface. On the left, a tree view shows the 'Objects' hierarchy, with 'HTTP\_Request' selected. The main panel shows the configuration for 'HTTP\_Request'. It includes a 'CONSTRAINTS' section with a single constraint: 'sourcetype=access\_\* OR sourcetype=ii\*' with 'Edit' and 'View Events' buttons. Below is an 'ATTRIBUTES' section with an 'Add Attribute' button. The attributes are listed in a table:

Inherited	
_time	timestamp
host	string
source	string
sourcetype	string

Auto-Extracted	uri	string (required)
Auto-Extracted	uri_path	string (required)
Auto-Extracted	status	number (required)
Auto-Extracted	clientip	ipv4 (required)
Auto-Extracted	referer	string (required)
Auto-Extracted	useragent	string (required)

At the bottom, a note states: 'Evals, Lookups, Regexs and Geo IPs are processed in the order above, so ensure any dependent attributes are defined first. Drag to rearrange.'

## Event

- Maps to Splunk events
- Requires constraints and attributes

# Three Root Object Types

Web Intelligence  
Webintelligence  
[Back to Data Models](#)

Objects Add Object ▾

EVENTS

- HTTP\_Request
  - ApacheAccessSearch
  - ISAccessSearch
- HTTP\_Success
  - Pageview
  - AssetAccess
    - DocAccess
    - MediaAccess
    - PodcastDownload
- HTTP\_Error
- HTTP\_Redirect

SEARCHES

User

BASE SEARCH

User  
User

[Documentation](#)

[Rename](#) [Delete](#)

[Edit](#) [View Results](#)

```
_time=* host=* source=* sourcetype=*  
uri=* status<600 clientip=* referer=*  
useragent=* (sourcetype = access_*  
OR source = *.log) | eval  
userid=clientip | stats first(_time) as  
earliest, last(_time) as latest,  
list(uri_path) as uri_list by userid
```

ATTRIBUTES Add Attribute ▾

Auto-Extracted	<a href="#">Edit</a>	<a href="#">Delete</a>	earliest	#	string (required)	<a href="#">View Values</a>
Auto-Extracted	<a href="#">Edit</a>	<a href="#">Delete</a>	latest	#	string (required)	<a href="#">View Values</a>
Auto-Extracted	<a href="#">Edit</a>	<a href="#">Delete</a>	uri_list	#	string (required)	<a href="#">View Values</a>

Evals, Lookups, Regexp and Geo IPs are processed in the order above, so ensure any dependent attributes are defined first. Drag to rearrange.

## Event

- Maps to Splunk events
- Requires constraints and attributes

## Search

- Maps to arbitrary Splunk search (may include generating, transforming and reporting search commands)
- Requires search string attributes

# Three Root Object Types

The screenshot shows the Splunk Web Intelligence interface. On the left is a navigation tree with categories: OBJECTS, EVENTS, SEARCHES, and TRANSACTIONS. Under OBJECTS, 'WebSession' is selected. The main panel displays the configuration for 'WebSession'. It includes a 'CONSTRAINTS' section with a table for defining constraints, and an 'ATTRIBUTES' section with a table for defining attributes. The 'ATTRIBUTES' table lists various fields like eventcount, duration, \_time, host, source, sourcetype, uri, uri\_path, status, clientip, referer, useragent, landingpage, and exitpage, each with options to hide, make optional, or make required, and a 'View Values' link.

Inherited	Hide	Make Optional	eventcount	#	number	View Values
Inherited	Hide	Make Optional	duration	#	number	View Values
Inherited	Hide	Make Optional	_time	o	timestamp	View Values
Inherited	Hide	Make Required	host	a	string	View Values
Inherited	Hide	Make Required	source	a	string	View Values
Inherited	Hide	Make Required	sourcetype	a	string	View Values
Inherited	Hide	Make Required	uri	a	string	View Values
Inherited	Hide	Make Required	uri_path	a	string	View Values
Inherited	Hide	Make Required	status	a	string	View Values
Inherited	Hide	Make Required	clientip	a	string	View Values
Inherited	Hide	Make Required	referer	a	string	View Values
Inherited	Hide	Make Required	useragent	a	string	View Values
Eval Expression	Edit	Delete	landingpage	a	string	View Values
Eval Expression	Edit	Delete	exitpage	a	string	View Values

## Event

- Maps to Splunk events
- Requires constraints and attributes

## Search

- Maps to arbitrary Splunk search (may include generating, transforming and reporting search commands)
- Requires search string attributes

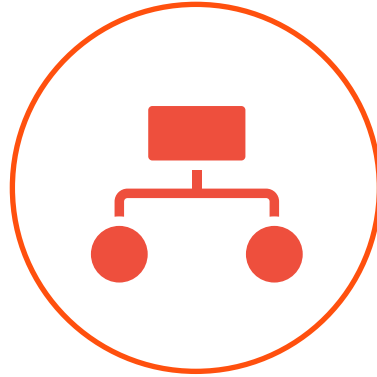
## Transaction

- Maps to groups of Splunk events or groups of Splunk search results
- Requires objects to group, fields/conditions to group by, and attributes

# Anatomy of a Data Model



Data models are comprised of one or more objects



Hierarchical Parent/Child Relationship



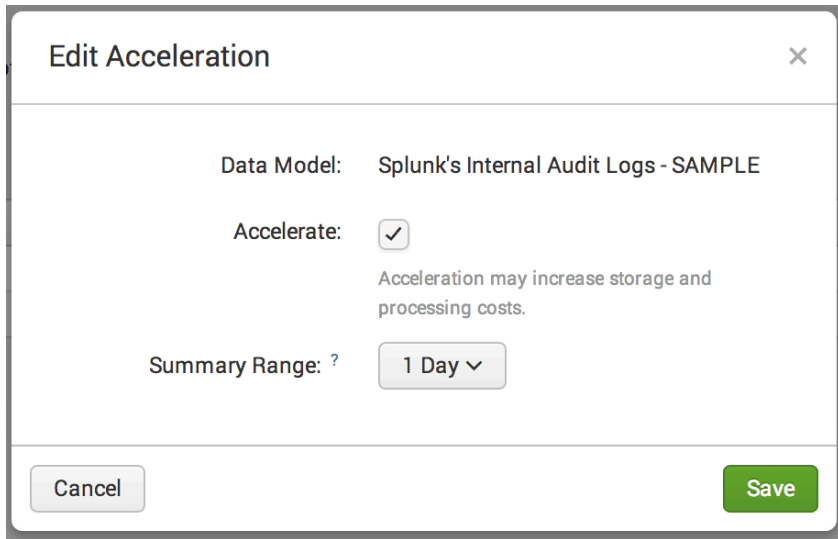
The child objects inherit the constraints (searches, filter on the data) and attributes.nt object

# Where are they Stored?

```
Search Based : 0
},
"objects": [
  {
    "objectName": "Root_Event",
    "displayName": "Root Event",
    "parentName": "BaseEvent",
    "fields": [
      {
        "fieldName": "action",
        "owner": "Root_Event",
        "type": "string",
        "required": false,
        "multivalue": false,
        "hidden": false,
        "editable": true,
        "displayName": "action",
        "comment": "",
        "fieldSearch": ""
      }
    ]
  }
],
```

- Each data model is a separate JSON file
- Lives in [app\_name]/[local|default]/data/models
- Editing this file is not supported!

# Data Model Acceleration



Edit Acceleration

Data Model: Splunk's Internal Audit Logs - SAMPLE

Accelerate:

Acceleration may increase storage and processing costs.

Summary Range: ? 1 Day ▾

Cancel Save

- Data model acceleration is a tool that you can use to speed up data models that represent extremely large datasets
- Data model acceleration summaries take the form of time-series index files (.tsidx) stored on Indexers
- Cannot edit accelerated models

# Data Model Acceleration



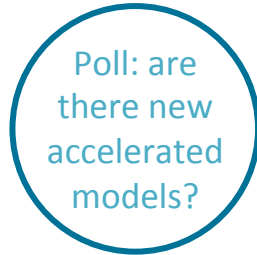
Admin/Power User



Setting written to conf file



splunk> Server



Kick off collection



Non-technical User



Run search using on-disk acceleration



Kick off ad-hoc acceleration and run search

# Best Practices

- Use event objects as much as possible: benefit from data model acceleration
- Minimize object hierarchy depth where possible: Constraint based filtering is less efficient as you move down the tree
- Data model acceleration is most efficient if the root event object being accelerated includes in its initial constraint search the index(es) that should be searched over

# Things to Watch For...

- Data model acceleration only affects the first event object hierarchy in a data model
- Object constraints and attributes cannot contain pipes or subsearches
- Lookups used in attributes must be globally visible (or at least visible to app using the data model)
- No versioning on data models or objects

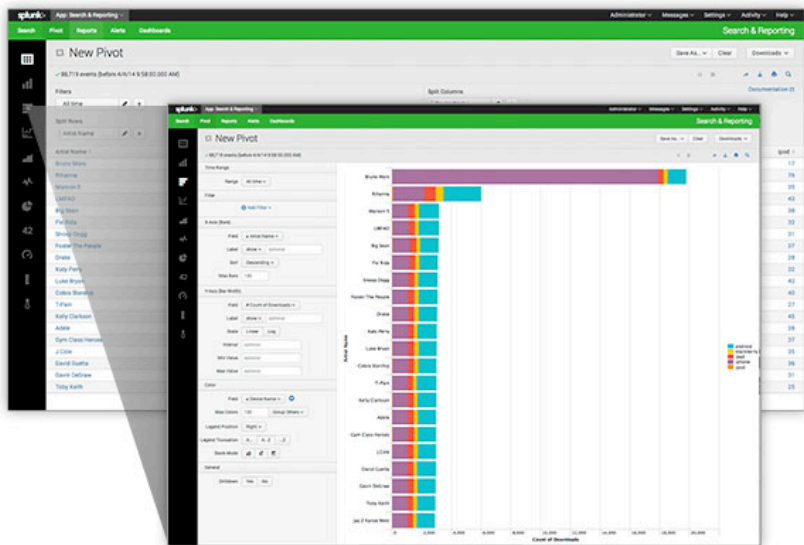
.conf2014

# YOUR DATA ADVENTURE

Pivot

splunk>

# How Does Pivot Work?



- Data models used to define the broad category of event data
- Hierarchically arranged collections of data model objects to subdivide the original datasets
- Define the attributes that you want Pivot to return results on

# Pivot Editor

The screenshot displays the Splunk Pivot Editor interface. At the top, the title is "New Pivot". On the right side of the header, there are buttons for "Save As...", "Clear", and a dropdown menu currently set to "Sales". Below the header, the main content area shows the following configuration:

- Filters:** A dropdown menu set to "All time" with edit and add icons.
- Split Rows:** A button with a "+" sign.
- Split Columns:** A button with a "+" sign.
- Column Values:** A dropdown menu set to "Count of Sales" with edit and add icons.

Below the configuration panels, the pivot result is displayed as a table with one row:

Count of Sales
2465

On the left side of the interface, there is a vertical sidebar with various visualization icons, including a bar chart, a table, a line chart, a pie chart, and a heatmap. The number "42" is visible in the sidebar, likely representing the number of results.

# Search Commands

- Search commands that allow you to utilize data models
  - datamodel
  - tstats
  - pivot

# Pivot

- Required arguments
  - datamodel-name
  - objectname
  - Pivot search
    - Has its own syntax that is different than the Splunk Search Processing Language
    - There are various elements that can be used here such as cell values, rows, columns, filters, limits, row and column formatting, and row sort options
- Command that fuels the Pivot Editor

# Using the Pivot Command on a Custom Dashboard

- Create a search using the Pivot Editor
- Click the Open in Search magnifying glass
- Create a dashboard with multiple user inputs
- Use the |pivot search you created with the Pivot Editor as the search for your dashboard panel
- Update the |pivot search with the filters you wish to provide your users

.conf2014

# YOUR DATA ADVENTURE

Demo

*This demo will show you how to use the power of accelerated data models on a custom dashboard*

splunk>

.conf2014

YOUR DATA  
ADVENTURE

THANK YOU

splunk>