This documentation does not apply to the most recent version of Splunk. Click **here (/Documentation/Splunk/latest/SearchReference/ListOfDataTypes)** for the latest version.

List of data types

This topic is out of date.

This page lists the data types used to define the syntax of the search language. Learn more about the commands used in these examples by referring to the Command quick reference (http://docs.splunk.com/Documentation/Splunk/6.0.2/SearchReference/ListOfSearchCommands).

after-opt

Syntax: timeafter=<int>(s|m|h|d)?
Description: the amount of time to add to endtime (ie expand the time region forward in time)

anovalue-action-option

Syntax: action=(annotate|filter|summary)
Description: If action is ANNOTATE, a new field is added to the event containing the anomalous value that indicates the anomaly score of the value If action is FILTER, events with anomalous value(s) are isolated. If action is SUMMARY, a table summarizing the anomaly statistics for each field is generated.

anovalue-pthresh-option

Syntax: pthresh=<num>
Description: Probability threshold (as a decimal) that has to be met for a value to be deemed anomalous

associate-improv-option

Syntax: improv=<num>
Description: Minimum entropy improvement for target key. That is, entropy(target key) - entropy(target key given reference key/value) must be greater than or equal to this.

associate-option

Syntax: <associate-supcnt-option>|<associate-supfreq-option>|<associate-improv-option>
Description: Associate command options

associate-supcnt-option

Syntax: supcnt=<int>
Description: Minimum number of times the reference key=reference value combination must be appear. Must be a non-negative integer.

associate-supfreq-option

Syntax: supfreq=<num>
Description: Minimum frequency of reference key=reference value combination, as a fraction of the number of total events.

before-opt

Syntax: timebefore=<int>(s|m|h|d)?
Description: the amount of time to subtract from starttime (ie expand the time region backwards in time)

bucket-bins

Syntax: bins=<int>
Description: Sets the maximum number of bins to discretize into. Given this upper-bound guidance, the bins will snap to human sensible bounds.
Example: bins=10

bucket-span

Syntax: span=(<span-length>|<log-span>)
Description: Sets the size of each bucket.
Example: span=2d
Example: span=5m
Example: span=10

bucket-start-end

Syntax: (start=|end=)<num>
Description: Sets the minimum and maximum extents for numerical buckets.

bucketing-option

Syntax: <bucket-bins>|<bucket-span>|<bucket-start-end>
Description: Discretization option.

by-clause

Syntax: by <field-list>
Description: Fields to group by.
Example: BY addr, port
Example: BY host

cmp

Syntax: =|!=|<|<=|>|>=
Description: None

collapse-opt

Syntax: collapse=<bool>
Description: whether to collapse terms that are a prefix of another term and the event count is the same
Example: collapse=f

collect-addinfo

Syntax: No syntax
Description: None

collect-addtime

Syntax: addtime=<bool>
Description: whether to prefix a time into each event if the event does not contain a _raw field. The first found field of the following times is used: info_min_time, _time, now() defaults to true

collect-arg

Syntax: <collect-addtime> | <collect-index> | <collect-file> | <collect-spool> | <collect-marker> | <collect-testmode>
Description: None

collect-file

Syntax: file=<string>
Description: name of the file where to write the events to. Optional, default "<random-num>_events.stash" The following placeholders can be used in the file name $timestamp$, $random$ and will be replaced with a timestamp, a random number respectively

collect-index

Syntax: index=<string>
Description: name of the index where splunk should add the events to. Note: the index must exist for events to be added to it, the index is NOT created automatically.

collect-marker

Syntax: marker=<string>
Description: a string, usually of key-value pairs, to append to each event written out. Optional, default ""

collect-spool

Syntax: spool=<bool>
Description: If set to true (default is true), the summary indexing file will be written to Splunk's spool directory, where it will be indexed automatically. If set to false, file will be written to $SPLUNK_HOME/var/run/splunk.

collect-testmode

Syntax: testmode=<bool>
Description: toggle between testing and real mode. In testing mode the results are not written into the new index but the search results are modified to appear as they would if sent to the index. (defaults to false)

comparison-expression

Syntax: <field><cmp><value>
Description: None

connected-opt

Syntax: connected=<bool>
Description: Relevant iff fields is not empty. Controls whether an event that is not inconsistent and not consistent with the fields of a transaction, opens a

new transaction (connected=t) or is added to the transaction. An event can be not inconsistent and not consistent if it contains fields required by the transaction but none of these fields has been instantiated in the transaction (by a previous event addition).

contingency-maxopts

Syntax: (maxrows|maxcols)=<int>
Description: Maximum number of rows or columns. If the number of distinct values of the field exceeds this maximum, the least common values will be ignored. A value of 0 means unlimited rows or columns.

contingency-mincover

Syntax: (mincolcover|minrowcover)=<num>
Description: Cover only this percentage of values for the row or column field. If the number of entries needed to cover the required percentage of values exceeds maxrows or maxcols, maxrows or maxcols takes precedence.

contingency-option

Syntax: <contingency-maxopts>|<contingency-mincover>|<contingency-usetotal>|<contingency-totalstr>
Description: Options for the contingency table

contingency-totalstr

Syntax: totalstr=<field>
Description: Field name for the totals row/column

contingency-usetotal

Syntax: usetotal=<bool>
Description: Add row and column totals

convert-auto

Syntax: auto("(" (<wc-field>)? ")")?
Description: Automatically convert the field(s) to a number using the best conversion. Note that if not all values of a particular field can be converted using a known conversion type, the field is left untouched and no conversion at all in done for that field.
Example: ... | convert auto(*delay) as *delay_secs
Example: ... | convert auto(*) as *_num
Example: ... | convert auto(delay) auto(xdelay)
Example: ... | convert auto(delay) as delay_secs
Example: ... | convert auto
Example: ... | convert auto()
Example: ... | convert auto(*)

convert-ctime

Syntax: ctime"("<wc-field>?")"
Description: Convert an epoch time to an ascii human readable time. Use timeformat option to specify exact format to convert to.
Example: ... | convert timeformat="%H:%M:%S" ctime(_time) as timestr

convert-dur2sec

Syntax: dur2sec"("<wc-field>?")"
Description: Convert a duration format "D+HH:MM:SS" to seconds.
Example: ... | convert dur2sec(*delay)
Example: ... | convert dur2sec(xdelay)

convert-function

Syntax: <convert-auto>|<convert-dur2sec>|<convert-mstime>|<convert-memk>|<convert-none>|<convert-num>|<convert-rmunit>|<convert-rmcomma>|<convert-ctime>|<convert-mktime>
Description: None

convert-memk

Syntax: memk"(" <wc-field>? ")"
Description: Convert a {KB, MB, GB} denominated size quantity into a KB
Example: ... | convert memk(VIRT)

convert-mktime

Syntax: mktime"("<wc-field>?")"
Description: Convert an human readable time string to an epoch time. Use timeformat option to specify exact format to convert from.
Example: ... | convert mktime(timestr)

convert-mstime

Syntax: mstime"(" <wc-field>? ")"
Description: Convert a MM:SS.SSS format to seconds.

convert-none

Syntax: none"(" <wc-field>? ")"

Description: In the presence of other wildcards, indicates that the matching fields should not be converted.
Example: ... | convert auto(*) none(foo)

convert-num

Syntax: num"("<wc-field>? ")"
Description: Like auto(), except non-convertible values are removed.

convert-rmcomma

Syntax: rmcomma"("<wc-field>? ")"
Description: Removes all commas from value, e.g. '1,000,000.00' -> '1000000.00'

convert-rmunit

Syntax: rmunit"(" <wc-field>? ")"
Description: Looks for numbers at the beginning of the value and removes trailing text.
Example: ... | convert rmunit(duration)

copyresults-dest-option

Syntax: dest=<string>
Description: The destination file where to copy the results to. The string is interpreted as path relative to SPLUNK_HOME and (1) should point to a .csv file and (2) the file should be located either in etc/system/lookups/ or etc/apps/<app-name>/lookups/

copyresults-sid-option

Syntax: sid=<string>
Description: The search id of the job whose results are to be copied. Note, the user who is running this command should have permission to the job pointed by this id.

correlate-type

Syntax: type=cocur
Description: Type of correlation to calculate. Only available option currently is the co-occurrence matrix, which contains the percentage of times that two fields exist in the same events.

count-opt

Syntax: count=<int>
Description: The maximum number of results to return
Example: count=10

crawl-option

Syntax: <string>=<string>
Description: Override settings from crawl.conf.
Example: root=/home/bob

daysago

Syntax: daysago=<int>
Description: Search the last N days. (equivalent to startdaysago)

debug-method

Syntax: optimize|roll||logchange|validate|delete|sync|sleep|rescan
Description: The available commands for debug command

dedup-consecutive

Syntax: consecutive=<bool>
Description: Only eliminate events that are consecutive

dedup-keepempty

Syntax: keepempty=<bool>
Description: If an event contains a null value for one or more of the specified fields, the event is either retained (if keepempty=true) or discarded

dedup-keepevents

Syntax: keepevents=<bool>
Description: Keep all events, remove specific values instead

default

Syntax: No syntax
Description: None

delim-opt

Syntax: delim=<string>

Description: A string used to delimit the original event values in the transaction event fields.

email_address

Syntax: <string>
Description: None
Example: bob@smith.com

email_list

Syntax: <email_address> (, <email_address>)*
Description: None
Example: "bob@smith.com, elvis@presley.com"

end-opt

Syntax: endswith=<transam-filter-string>
Description: A search or eval filtering expression which if satisfied by an event marks the end of a transaction
Example: endswith=eval(speed_field > max_speed_field/12)
Example: endswith=(username=foobar)
Example: endswith=eval(speed_field > max_speed_field)
Example: endswith="logout"

enddaysago

Syntax: enddaysago=<int>
Description: A short cut to set the end time. endtime = now - (N days)

endhoursago

Syntax: endhoursago=<int>
Description: A short cut to set the end time. endtime = now - (N hours)

endminutesago

Syntax: endminutesago=<int>
Description: A short cut to set the end time. endtime = now - (N minutes)

endmonthsago

Syntax: endmonthsago=<int>
Description: A short cut to set the start time. starttime = now - (N months)

endtime

Syntax: endtime=<string>
Description: All events must be earlier or equal to this time.

endtimeu

Syntax: endtime=<int>
Description: Set the end time to N seconds since the epoch. (unix time)

erex-examples

Syntax: ""<string>(, <string>)*""
Description: None
Example: "foo, bar"

eval-bool-exp

Syntax: (NOT|!)? (<eval-compare-exp>|<eval-function-call>) ((AND|OR|XOR) <eval-expression>)*
Description: None

eval-compare-exp

Syntax: (<field>|<string>|<num>) (<|>|<=|>=|!=|=|==|LIKE) <eval-expression>
Description: None

eval-concat-exp

Syntax: ((<field>|<string>|<num>) (. <eval-expression>)*)|((<field>|<string>) (+ <eval-expression>)*)
Description: concatenate fields and strings
Example: first_name." ".last_nameSearch

eval-expression

Syntax: <eval-math-exp> | <eval-concat-exp> | <eval-compare-exp> | <eval-bool-exp> | <eval-function-call>
Description: A combination of literals, fields, operators, and functions that represent the value of your destination field. The following are the basic operations you can perform with eval. For these evaluations to work, your values need to be valid for the type of operation. For example, with the exception of addition, arithmetic operations may not produce valid results if the values are not numerical. For addition, Splunk can concatenate the two operands if they are both strings. When concatenating values with '.', Splunk treats both values as strings regardless of their actual type.

eval-field

Syntax: <field>
Description: A field name for your evaluated value.
Example: velocity

eval-function

Syntax: abs|case|cidrmatch|coalesce|exact|exp|floor|if|ifnull|isbool|isint|isnotnull|isnull|isnum|isstr|len|like|ln|log|lower|match|max|md5|min|mvcount|mvindex|mvfilter|now
Description: Function used by eval.
Example: md5(field)
Example: typeof(12) + typeof("string") + typeof(1==2) + typeof(badfield)
Example: searchmatch("foo AND bar")
Example: sqrt(9)
Example: round(3.5)
Example: replace(date, "^(\d{1,2})/(\d{1,2})/", "\2/\1/")
Example: pi()
Example: nullif(fielda, fieldb)
Example: random()
Example: pow(x, y)
Example: mvfilter(match(email, "\.net$") OR match(email, "\.org$"))
Example: mvindex(multifield, 2)
Example: null()
Example: now()
Example: isbool(field)
Example: exp(3)
Example: floor(1.9)
Example: coalesce(null(), "Returned value", null())
Example: exact(3.14 * num)
Example: case(error == 404, "Not found", error == 500, "Internal Server Error", error == 200, "OK")
Example: cidrmatch("123.132.32.0/25", ip)
Example: abs(number)
Example: isnotnull(field)
Example: substr("string", 1, 3) + substr("string", -3)
Example: if(error == 200, "OK", "Error")
Example: len(field)
Example: log(number, 2)
Example: lower(username)
Example: match(field, "^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$")
Example: max(1, 3, 6, 7, "f"^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$")oo", field)
Example: like(field, "foo%")
Example: ln(bytes)
Example: mvcount(multifield)
Example: urldecode("http%3A%2F%2Fwww.splunk.com%2Fdownload%3Fr%3Dheader")
Example: validate(isint(port), "ERROR: Port is not an integer", port >= 1 AND port <= 65535, "ERROR: Port is out of range")
Example: tostring(1==1) + " " + tostring(15, "hex") + " " + tostring(12345.6789, "commas")
Example: trim(" ZZZZabcZZ ", " Z")

eval-function-call

Syntax: <eval-function> "(" <eval-expression> ("," <eval-expression>)* ")"
Description: None

eval-math-exp

Syntax: (<field>|<num>) ((+|-|*|/|%) <eval-expression>)*
Description: None
Example: pi() * pow(radius_a, 2) + pi() * pow(radius_b, 2)

evaled-field

Syntax: "eval("<eval-expression>")"
Description: A dynamically evaled field

event-id

Syntax: <int>:<int>
Description: a splunk internal event id

eventtype-specifier

Syntax: eventtype=<string>
Description: Search for events that match the specified eventtype

eventtypetag-specifier

Syntax: eventtypetag=<string>
Description: Search for events that would match all eventtypes tagged by the string

extract-opt

Syntax: (segment=<bool>)|(auto=<bool>)|(reload=<bool>)|(limit=<int>)|(maxchars=<int>)|(mv_add=<bool>)|(clean_keys=<bool>)
Description: Extraction options. "segment" specifies whether to note the locations of key/value pairs with the results (internal, false). "auto" specifies whether to perform automatic '=' based extraction (true). "reload" specifies whether to force reloading of props.conf and transforms.conf (false). "limit" specifies how many automatic key/value pairs to extract (50). "kvdelim" string specifying a list of character delimiters that separate the key from the value "pairdelim" string specifying a list of character delimiters that separate the key-value pairs from each other "maxchars" specifies how many characters to look into the event (10240). "mv_add" whether to create multivalued fields. Overrides MV_ADD from transforms.conf "clean_keys" whether to clean keys. Overrides CLEAN_KEYS from transforms.conf
Example: reload=true
Example: auto=false

extractor-name

Syntax: <string>
Description: A stanza that can be found in transforms.conf
Example: access-extractions

fields-opt

Syntax: fields=<string>? (,<string>)*
Description: DEPRECATED: The preferred usage of transaction is for list of fields to be specified directly as arguments. E.g. 'transaction foo bar' rather than 'transaction fields="foo,bar"' The 'fields' constraint takes a list of fields. For search results to be members of a transaction, for each field specified, if they have a value, it must have the same value as other members in that transaction. For example, a search result that has host=mylaptop can never be in the same transaction as a search result that has host=myserver, if host is one of the constraints. A search result that does not have a host value, however, can be in a transaction with another search result that has host=mylaptop, because they are not inconsistent.
Example: fields=host,cookie

grouping-field

Syntax: <field>
Description: By default, the typelearner initially groups events by the value of the grouping-field, and then further unifies and merges those groups, based on the keywords they contain. The default grouping field is "punct" (the punctuation seen in _raw).
Example: host

grouping-maxlen

Syntax: maxlen=<int>
Description: determines how many characters in the grouping-field value to look at. If set to negative, the entire value of the grouping-field value is used to initially group events
Example: maxlen=30

host-specifier

Syntax: host=<string>
Description: Search for events from the specified host

hosttag-specifier

Syntax: hosttag=<string>
Description: Search for events that have hosts that are tagged by the string

hoursago

Syntax: hoursago=<int>
Description: Search the last N hours. (equivalent to starthoursago)

increment

Syntax: <int:increment>(s|m|h|d)?
Description: None
Example: 1h

index-expression

Syntax: \"<string>\"|<term>|<search-modifier>
Description: None

index-specifier

Syntax: index=<string>
Description: Search the specified index instead of the default index

input-option

Syntax: <string>=<string>
Description: Override settings from inputs.conf.
Example: root=/home/bob

join-options

Syntax: usetime=<bool> | earlier=<bool> | overwrite=<bool> | max=<int>

Description: Options to the join command. usetime indicates whether to limit matches to sub results that are earlier or later (depending on the 'earlier' option which is only valid when usetime=true) than the main result to join with, default = false. 'overwrite' indicates if fields from the sub results should overwrite those from the main result if they have the same field name (default = true). max indicates the maximum number of sub results each main result can join with. (default = 1, 0 means no limit).
Example: max=3
Example: usetime=t earlier=f
Example: overwrite=f
Example: usetime=t

keepevicted-opt

Syntax: keepevicted=<bool>
Description: Whether to output evicted transactions. Evicted transactions can be distinguished from non-evicted transactions by checking the value of the 'evicted' field, which is set to '1' for evicted transactions

key-list

Syntax: (<string>)*
Description: a list of keys that are ANDed to provide a filter for surrounding command

kmeans-cnumfield

Syntax: cfield=<field>
Description: Controls the field name for the cluster number for each event

kmeans-distype

Syntax: dt=(l1norm|l2norm|cityblock|sqeuclidean|cosine)
Description: Distance metric to use (L1/L1NORM equivalent to CITYBLOCK). L2NORM equivalent to SQEUCLIDEAN

kmeans-iters

Syntax: maxiters=<int>
Description: Maximum number of iterations allowed before failing to converge

kmeans-k

Syntax: k=<int>(-<int>)?
Description: Number of initial clusters to use. Can be a range, in which case each value in the range will be used once and summary data given.

kmeans-options

Syntax: <kmeans-reps>|<kmeans-iters>|<kmeans-tol>|<kmeans-k>|<kmeans-cnumfield>|<kmeans-distype>|<kmeans-showlabel>
Description: Options for kmeans command

kmeans-reps

Syntax: reps=<int>
Description: Number of times to repeat kmeans using random starting clusters

kmeans-showlabel

Syntax: showlabel=<bool>
Description: Controls whether or not the cluster number is added to the data.

kmeans-tol

Syntax: tol=<num>
Description: Algorithm convergence tolerance

lit-value

Syntax: <string>|<num>
Description: None

lmaxpause-opt

Syntax: maxpause=<int>(s|m|h|d)?
Description: the maximum (inclusive) time between two consecutive events in a contiguous time region

log-span

Syntax: (<num>)?log(<num>)?
Description: Sets to log based span, first number if coefficient, second number is base coefficient, if supplied, must be real number >= 1.0 and < base base, if supplied, must be real number > 1.0 (strictly greater than 1)
Example: 2log5
Example: log

logical-expression

Syntax: (NOT)? <logical-expression>)|<comparison-expression>|(<logical-expression> OR? <logical-expression>)
Description: None

max-time-opt

Syntax: max_time=<int>
Description: None
Example: max_time=3

maxevents-opt

Syntax: maxevents=<int>
Description: The maximum number of events in a transaction. If the value is negative this constraint is disabled.

maxinputs-opt

Syntax: maxinputs=<int>
Description: Determines how many of the top results are passed to the script.
Example: maxinputs=1000

maxopenevents-opt

Syntax: maxopenevents=<int>
Description: Specifies the maximum number of events (which are) part of open transactions before transaction eviction starts happening, using LRU policy.

maxopentxn-opt

Syntax: maxopentxn=<int>
Description: Specifies the maximum number of not yet closed transactions to keep in the open pool before starting to evict transactions, using LRU policy.

maxpause-opt

Syntax: maxpause=<int>(s|m|h|d)?
Description: The maxpause constraint requires there be no pause between a transaction's events of greater than maxpause. If value is negative, disable the maxpause constraint.

maxsearchesoption

Syntax: maxsearches=<int>
Description: The maximum number of searches to run. Will generate warning if there are more search results.
Example: maxsearches=42

maxspan-opt

Syntax: maxspan=<int>(s|m|h|d)?
Description: The maxspan constraint requires the transaction's events to span less than maxspan. If value is negative, disable the maxspan constraint.

memcontrol-opt

Syntax: <maxopentxn-opt> | <maxopenevents-opt> | <keepevicted-opt>
Description: None

metadata-delete-restrict

Syntax: (host::|source::|sourcetype::)<string>
Description: restrict the deletion to the specified host, source or sourcetype.

metadata-type

Syntax: hosts|sources|sourcetypes
Description: controls which metadata type that will be returned

minutesago

Syntax: minutesago=<int>
Description: Search the last N minutes. (equivalent to startminutesago)

monthsago

Syntax: monthsago=<int>
Description: Search the last N months. (equivalent to startmonthsago)

multikv-copyattrs

Syntax: copyattrs=<bool>
Description: Controls the copying of non-metadata attributes from the original event to extract events (default = true)

multikv-fields

Syntax: fields <field-list>
Description: Filters out from the extracted events fields that are not in the given field list

multikv-filter

Syntax: filter <field-list>
Description: If specified, a table row must contain one of the terms in the list before it is extracted into an event

multikv-forceheader

Syntax: forceheader=<int>
Description: Forces the use of the given line number (1 based) as the table's header. By default a header line is searched for.

multikv-multitable

Syntax: multitable=<bool>
Description: Controls whether or not there can be multiple tables in a single _raw in the original events? (default = true)

multikv-noheader

Syntax: noheader=<bool>
Description: Allow tables with no header? If no header fields would be named column1, column2, ... (default = false)

multikv-option

Syntax: <multikv-copyattrs>|<multikv-fields>|<multikv-filter>|<multikv-forceheader>|<multikv-multitable>|<multikv-noheader>|<multikv-rmorig>
Description: Multikv available options

multikv-rmorig

Syntax: rmorig=<bool>
Description: Controls the removal of original events from the result set (default=true)

mvlist-opt

Syntax: mvlist=<bool>|<field-list>
Description: Flag controlling whether the multivalued fields of the transaction are (1) a list of the original events ordered in arrival order or (2) a set of unique field values ordered lexigraphically. If a comma/space delimited list of fields is provided only those fields are rendered as lists

outlier-action-opt

Syntax: action=(remove|transform)
Description: What to do with outliers. RM | REMOVE removes the event containing the outlying numerical value. TF | TRANSFORM truncates the outlying value to the threshold for outliers and prefixes the value with "000"

outlier-option

Syntax: <outlier-type-opt>|<outlier-action-opt>|<outlier-param-opt>|<outlier-uselower-opt>
Description: Outlier options

outlier-param-opt

Syntax: param=<num>
Description: Parameter controlling the threshold of outlier detection. For type=IQR, an outlier is defined as a numerical value that is outside of param multiplied the inter-quartile range.

outlier-type-opt

Syntax: type=iqr
Description: Type of outlier detection. Only current option is IQR (inter-quartile range)

outlier-uselower-opt

Syntax: uselower=<bool>
Description: Controls whether to look for outliers for values below the median

prefix-opt

Syntax: prefix=<string>
Description: The prefix to do typeahead on
Example: prefix=source

quoted-str

Syntax: "" <string> ""
Description: None

readlevel-int

Syntax: 0|1|2|3
Description: How deep to read the events, 0 : just source/host/sourcetype, 1 : 0 with _raw, 2 : 1 with kv, 3 2 with types (deprecated in 3.2)

regex-expression

Syntax: (\")?<string>(\")?
Description: A Perl Compatible Regular Expression supported by the pcre library.
Example: ... | regex _raw="(?<!\d)10.\d{1,3}\.\d{1,3}\.\d{1,3}(?!\d)"

rendering-opt

Syntax: <delim-opt> | <mvlist-opt>

Description: None

result-event-opt

Syntax: events=<bool>
Description: Option controlling whether to load the events or results of a job. (default: false)
Example: events=t

savedsearch-identifier

Syntax: savedsearch="<user-string>:<application-string>:<search-name-string>"
Description: The unique identifier of a savedsearch whose artifacts need to be loaded. A savedsearch is uniquely identified by the triplet {user, application, savedsearch name}.
Example: savedsearch="admin:search:my saved search"

savedsearch-macro-opt

Syntax: nosubstitution=<bool>
Description: If true, no macro replacements are made.

savedsearch-opt

Syntax: <savedsearch-macro-opt>|<savedsearch-replacement-opt>
Description: None

savedsearch-replacement-opt

Syntax: <string>=<string>
Description: A key value pair to be used in macro replacement.

savedsplunk-specifier

Syntax: (savedsearch|savedsplunk)=<string>
Description: Search for events that would be found by specified search/splunk

savedsplunkoption

Syntax: <string>
Description: Name of saved search
Example: mysavedsearch

script-arg

Syntax: <string>
Description: An argument passed to the script.
Example: to=bob@mycompany.com

script-name-arg

Syntax: <string>
Description: The name of the script to execute, minus the path and file extension.
Example: sendemail

search-modifier

Syntax: <sourcetype-specifier>|<host-specifier>|<source-specifier>|<savedsplunk-specifier>|<eventtype-specifier>|<eventtypetag-specifier>|<hosttag-specifier>|<tag-specifier>
Description: None

searchoption

Syntax: search=\"<string>\"
Description: Search to run map on
Example: search="search starttimeu::$start$ endtimeu::end"

searchtimespandays

Syntax: searchtimespandays=<int>
Description: None

searchtimespanhours

Syntax: searchtimespanhours=<int>
Description: The time span operators are always applied from the last time boundary set. Therefore, if an endtime operator is closest to the left of a timespan operator, it will be applied to the starttime. If you had 'enddaysago::1 searchtimespanhours::5', it would be equivalent to 'starthoursago::29 enddaysago::1'.

searchtimespanminutes

Syntax: searchtimespanminutes=<int>
Description: None

searchtimespanmonths

Syntax: searchtimespanmonths=<int>
Description: None

select-arg

Syntax: <string>
Description: Any value sql select arguments, per the syntax found at http://www.sqlite.org/lang_select.html (http://www.sqlite.org/lang_select.html). If no "from results" is specified in the select-arg it will be inserted it automatically. Runs a SQL Select query against passed in search results. All fields referenced in the select statement must be prefixed with an underscore. Therefore, "ip" should be references as "_ip" and "_raw" should be referenced as "__raw". Before the select command is executed, the previous search results are put into a temporary database table called "results". If a row has no values, "select" ignores it to prevent blank search results.

selfjoin-options

Syntax: overwrite=<bool> | max=<int> | keepsingle=<int>
Description: The selfjoin joins each result with other results that have the same value for the join fields. 'overwrite' controls if fields from these 'other' results should overwrite fields of the result used as the basis for the join (default=true). max indicates the maximum number of 'other' results each main result can join with. (default = 1, 0 means no limit). 'keepsingle' controls whether or not results with a unique value for the join fields (and thus no other results to join with) should be retained. (default = false)
Example: max=3
Example: keepsingle=t
Example: overwrite=f

server-list

Syntax: (<string>)*
Description: A list of possibly wildcarded servers changes in the context of the differences. Try it see if it makes sense. * - header=[true | false] : optionally you can show a header that tries to explain the diff output * - attribute=[attribute name] : you can choose to diff just a single attribute of the results.

sid-opt

Syntax: <string>
Description: The search id of the job whose artifacts need to be loaded.
Example: 1233886270.2

single-agg

Syntax: count|<stats-func>(<field>)
Description: A single aggregation applied to a single field (can be evaled field). No wildcards are allowed. The field must be specified, except when using the special 'count' aggregator that applies to events as a whole.
Example: avg(delay)
Example: sum({date_hour * date_minute})
Example: count

slc-option

Syntax: (t=<num>|(delims=<string>)|(showcount=<bool>)|(countfield=<field>)|(labelfield=<field>)|(field=<field>)|(labelonly=<bool>)|(match=(termlist|termset|ngramset)))
Description: Options for configuring the simple log clusters. "T=" sets the threshold which must be > 0.0 and < 1.0. The closer the threshold is to 1, the more similar events have to be in order to be considered in the same cluster. Default is 0.8 "delims" configures the set of delimiters used to tokenize the raw string. By default everything except 0-9, A-Z, a-z, and '_' are delimiters. "showcount" if yes, this shows the size of each cluster (default = true unless labelonly is set to true) "countfield" name of field to write cluster size to, default = "cluster_count" "labelfield" name of field to write cluster number to, default = "cluster_label" "field" name of field to analyze, default = _raw "labelonly" if true, instead of reducing each cluster to a single event, keeps all original events and merely labels with them their cluster number "match" determines the similarity method used, defaulting to termlist. termlist requires the exact same ordering of terms, termset allows for an unordered set of terms, and ngramset compares sets of trigram (3-character substrings). ngramset is significantly slower on large field values and is most useful for short non-textual fields, like 'punct'
Example: t=0.9 delims=" ;:" showcount=true countfield="SLCCNT" labelfield="LABEL" field=_raw labelonly=true

sort-by-clause

Syntax: ("-"|"+")<sort-field> ","
Description: List of fields to sort by and their sort order (ascending or descending)
Example: - time, host
Example: -size, +source
Example: _time, -host

sort-field

Syntax: <field> | ((auto|str|ip|num) "(" <field> ")")
Description: a sort field may be a field or a sort-type and field. sort-type can be "ip" to interpret the field's values as ip addresses. "num" to treat them as numbers, "str" to order lexigraphically, and "auto" to make the determination automatically. If no type is specified, it is assumed to be "auto"
Example: host
Example: _time
Example: ip(source_addr)
Example: str(pid)
Example: auto(size)

source-specifier

Syntax: source=<string>
Description: Search for events from the specified source

sourcetype-specifier

Syntax: sourcetype=<string>
Description: Search for events from the specified sourcetype

span-length

Syntax: <int:span>(<timescale>)?
Description: Span of each bin. If using a timescale, this is used as a time range. If not, this is an absolute bucket "length."
Example: 2d
Example: 5m
Example: 10

split-by-clause

Syntax: <field> (<tc-option>)* (<where-clause>)?
Description: Specifies a field to split by. If field is numerical, default discretization is applied.

srcfields

Syntax: (<field>|<quoted-str>) (<field>|<quoted-str>) (<field>|<quoted-str>)*
Description: Fields should either be key names or quoted literals

start-opt

Syntax: startswith=<transam-filter-string>
Description: A search or eval filtering expression which if satisfied by an event marks the beginning of a new transaction
Example: startswith=eval(speed_field < max_speed_field/12)
Example: startswith=(username=foobar)
Example: startswith=eval(speed_field < max_speed_field)
Example: startswith="login"

startdaysago

Syntax: startdaysago=<int>
Description: A short cut to set the start time. starttime = now - (N days)

starthoursago

Syntax: starthoursago=<int>
Description: A short cut to set the start time. starttime = now - (N hours)

startminutesago

Syntax: startminutesago=<int>
Description: A short cut to set the start time. starttime = now - (N minutes)

startmonthsago

Syntax: startmonthsago=<int>
Description: A short cut to set the start time. starttime = now - (N months)

starttime

Syntax: starttime=<string>
Description: Events must be later or equal to this time. Must match time format.

starttimeu

Syntax: starttimeu=<int>
Description: Set the start time to N seconds since the epoch. (unix time)

stats-agg

Syntax: <stats-func>("(" (<evaled-field> | <wc-field>)? ")")?
Description: A specifier formed by a aggregation function applied to a field or set of fields. As of 4.0, it can also be an aggregation function applied to a arbitrary eval expression. The eval expression must be wrapped by "{" and "}". If no field is specified in the parenthesis, the aggregation is applied independently to all fields, and is equivalent to calling a field value of * When a numeric aggregator is applied to a not-completely-numeric field no column is generated for that aggregation.
Example: count({sourcetype="splunkd"})
Example: max(size)
Example: stdev(*delay)
Example: avg(kbps)

stats-agg-term

Syntax: <stats-agg> (as <wc-field>)?

Description: A statistical specifier optionally renamed to a new field name.
Example: count(device) AS numdevices
Example: avg(kbps)

stats-c

Syntax: count
Description: The count of the occurrences of the field.

stats-dc

Syntax: distinct-count
Description: The count of distinct values of the field.

stats-first

Syntax: first
Description: The first seen value of the field.

stats-func

Syntax: <stats-c>|<stats-dc>|<stats-mean>|<stats-stdev>|<stats-var>|<stats-sum>|<stats-min>|<stats-max>|<stats-mode>|<stats-median>|<stats-first>|<stats-last>|<stats-perc>|<stats-list>|<stats-values>|<stats-range>
Description: Statistical aggregators.

stats-last

Syntax: last
Description: The last seen value of the field.

stats-list

Syntax: list
Description: List of all values of this field as a multi-value entry. Order of values reflects order of input events.

stats-max

Syntax: max
Description: The maximum value of the field (lexicographic, if non-numeric).

stats-mean

Syntax: avg
Description: The arithmetic mean of the field.

stats-median

Syntax: median
Description: The middle-most value of the field.

stats-min

Syntax: min
Description: The minimum value of the field (lexicographic, if non-numeric).

stats-mode

Syntax: mode
Description: The most frequent value of the field.

stats-perc

Syntax: perc<int>
Description: The n-th percentile value of this field.

stats-range

Syntax: range
Description: The difference between max and min (only if numeric)

stats-stdev

Syntax: stdev|stdevp
Description: The {sample, population} standard deviation of the field.

stats-sum

Syntax: sum
Description: The sum of the values of the field.

stats-values

Syntax: values

Description: List of all distinct values of this field as a multi-value entry. Order of values is lexigraphical.

stats-var

Syntax: var|varp
Description: The {sample, population} variance of the field.

subsearch

Syntax: [<string>]
Description: Specifies a subsearch.
Example: [search 404 | select url]

subsearch-options

Syntax: maxtime=<int> | maxout=<int> | timeout=<int>
Description: controls how the subsearch is executed.

tc-option

Syntax: <bucketing-option>|(usenull=<bool>)|(useother=<bool>)|(nullstr=<string>)|(otherstr=<string>)
Description: Options for controlling the behavior of splitting by a field. In addition to the bucketing-option: usenull controls whether or not a series is created for events that do not contain the split-by field. This series is labeled by the value of the nullstr option, and defaults to NULL. useother specifies if a series should be added for data series not included in the graph because they did not meet the criteria of the <where-clause>. This series is labeled by the value of the otherstr option, and defaults to OTHER.
Example: otherstr=OTHERFIELDS
Example: usenull=f
Example: bins=10

time-modifier

Syntax: <starttime>|<startdaysago>|<startminutesago>|<starthoursago>|<startmonthsago>|<starttimeu>|<endtime>|<enddaysago>|<endminutesago>|<endhoursago>|<endmonthsago>|<endtimeu>|<searchtimespanhours>|<searchtimespanminutes>|<searchtimespandays>|<searchtimespanmonths>|<daysago>|<minutesago>|<hoursago>|<monthsago>
Description: None

time-opts

Syntax: (<timeformat>)? (<time-modifier>)*
Description: None

timeformat

Syntax: timeformat=<string>
Description: Set the time format for starttime and endtime terms.
Example: timeformat=%m/%d/%Y:%H:%M:%S

timescale

Syntax: <ts-sec>|<ts-min>|<ts-hr>|<ts-day>|<ts-month>|<ts-subseconds>
Description: Time scale units.

timestamp

Syntax: (MM/DD/YY)?:(HH:MM:SS)?|<int>
Description: None
Example: 10/1/07:12:34:56
Example: -5

top-opt

Syntax: (showcount=<bool>)|(showperc=<bool>)|(rare=<bool>)|(limit=<int>)|(countfield=<string>)|(percentfield=<string>)
Description: Top arguments: showcount: Whether to create a field called "count" (see countfield option) with the count of that tuple. (T) showperc: Whether to create a field called "percent" (see percentfield option) with the relative prevalence of that tuple. (T) rare: When set and calling as top or common, evokes the behavior of calling as rare. (F) limit: Specifies how many tuples to return, 0 returns all values. (10) countfield: Name of new field to write count to (default is "count") percentfield: Name of new field to write percentage to (default is "percent")

transaction-name

Syntax: <string>
Description: The name of a transaction definition from transactions.conf to be used for finding transactions. If other arguments (e.g., maxspan) are provided as arguments to transam, they overrule the value specified in the transaction definition.
Example: purchase_transaction

transam-filter-string

Syntax: "<search-expression>" | (<quoted-search-expression>) | eval(<eval-expression>)
Description: Where: \i\ <search-expression> is a valid search expression that does not contain quotes\i\ <quoted-search-expression> is a valid search expression that contains quotes\i\ <eval-expression> is a valid eval expression that evaluates to a boolean
Example: eval(distance/time < max_speed)
Example: "user=mildred"

Example: ("search literal")
Example: (name="foo bar")

trend_type

Syntax: (sma|ema|wma)<num>
Description: The type of trend to compute which consist of a trend type and trend period (integer between 2 and 10000)
Example: sma10

ts-day

Syntax: days
Description: Time scale in days.

ts-hr

Syntax: hours
Description: Time scale in hours.

ts-min

Syntax: minutes
Description: Time scale in minutes.

ts-month

Syntax: months
Description: Time scale in months.

ts-sec

Syntax: seconds
Description: Time scale in seconds.

ts-subseconds

Syntax: us|ms|cs|ds
Description: Time scale in microseconds("us"), milliseconds("ms"), centiseconds("cs"), or deciseconds("ds")

txn_definition-opt

Syntax: <maxspan-opt> | <maxpause-opt> | <maxevents-opt> | <field-list> | <start-opt> | <end-opt> | <connected-opt>
Description: None

value

Syntax: <lit-value>|<field>
Description: None

where-clause

Syntax: where <single-agg> <where-comp>
Description: Specifies the criteria for including particular data series when a field is given in the tc-by-clause. This optional clause, if omitted, default to "where sum in top10". The aggregation term is applied to each data series and the result of these aggregations is compared to the criteria. The most common use of this option is to select for spikes rather than overall mass of distribution in series selection. The default value finds the top ten series by area under the curve. Alternately one could replace sum with max to find the series with the ten highest spikes.
Example: where max < 10
Example: where count notin bottom10
Example: where avg > 100
Example: where sum in top5

where-comp

Syntax: <wherein-comp>|<wherethresh-comp>
Description: A criteria for the where clause.

wherein-comp

Syntax: (in|notin) (top|bottom)<int>
Description: A where-clause criteria that requires the aggregated series value be in or not in some top or bottom grouping.
Example: notin top2
Example: in bottom10
Example: in top5

wherethresh-comp

Syntax: (<|>)( )?<num>
Description: A where-clause criteria that requires the aggregated series value be greater than or less than some numeric threshold.
Example: < 100
Example: > 2.5

x-field

Syntax: <field>
Description: Field to be used as the x-axis

y-data-field

Syntax: <field>
Description: Field that contains the data to be charted

y-name-field

Syntax: <field>
Description: Field that contains the values to be used as data series labels

(/Documentation/Splunk/6.0.2/SearchReference/SQLtoSplunk)

PREVIOUS
Splunk SPL for SQL users
(/Documentation/Splunk/6.0.2/SearchReference/SQLtoSplunk)

Function
(/Documentation/Splunk/6.0.2/SearchReferen

This documentation applies to the following versions of Splunk® Enterprise: 4.3 (/Category:V:Splunk:4.3), 4.3.1 (/Category:V:Splunk:4.3.1), 4.3.2 (/Category:V:Splunk:4.3.2), 4.3.3 (/Category:V:Splunk:4.3.3), 4.3.4 (/Category:V:Splunk:4.3.4), 4.3.5 (/Category:V:Splunk:4.3.5), 4.3.6 (/Category:V:Splunk:4.3.6), 4.3.7 (/Category:V:Splunk:4.3.7), 5.0 (/Category:V:Splunk:5.0), 5.0.1 (/Category:V:Splunk:5.0.1), 5.0.2 (/Category:V:Splunk:5.0.2), 5.0.3 (/Category:V:Splunk:5.0.3), 5.0.4 (/Category:V:Splunk:5.0.4), 5.0.5 (/Category:V:Splunk:5.0.5), 5.0.6 (/Category:V:Splunk:5.0.6), 5.0.7 (/Category:V:Splunk:5.0.7), 5.0.8 (/Category:V:Splunk:5.0.8), 5.0.9 (/Category:V:Splunk:5.0.9), 5.0.10 (/Category:V:Splunk:5.0.10), 5.0.11 (/Category:V:Splunk:5.0.11), 5.0.12 (/Category:V:Splunk:5.0.12), 5.0.13 (/Category:V:Splunk:5.0.13), 5.0.14 (/Category:V:Splunk:5.0.14), 5.0.15 (/Category:V:Splunk:5.0.15), 6.0 (/Category:V:Splunk:6.0), 6.0.1 (/Category:V:Splunk:6.0.1), 6.0.2 (/Category:V:Splunk:6.0.2), 6.0.3 (/Category:V:Splunk:6.0.3), 6.0.4 (/Category:V:Splunk:6.0.4), 6.0.5 (/Category:V:Splunk:6.0.5), 6.0.6 (/Category:V:Splunk:6.0.6), 6.0.7 (/Category:V:Splunk:6.0.7), 6.0.8 (/Category:V:Splunk:6.0.8), 6.0.9 (/Category:V:Splunk:6.0.9), 6.0.10 (/Category:V:Splunk:6.0.10), 6.0.11 (/Category:V:Splunk:6.0.11), 6.1 (/Category:V:Splunk:6.1), 6.1.1 (/Category:V:Splunk:6.1.1), 6.1.2 (/Category:V:Splunk:6.1.2), 6.1.3 (/Category:V:Splunk:6.1.3), 6.1.4 (/Category:V:Splunk:6.1.4), 6.1.5 (/Category:V:Splunk:6.1.5), 6.1.6 (/Category:V:Splunk:6.1.6), 6.1.7 (/Category:V:Splunk:6.1.7), 6.1.8 (/Category:V:Splunk:6.1.8), 6.1.9 (/Category:V:Splunk:6.1.9), 6.1.10 (/Category:V:Splunk:6.1.10), 6.2.0 (/Category:V:Splunk:6.2.0), 6.2.1 (/Category:V:Splunk:6.2.1), 6.2.2 (/Category:V:Splunk:6.2.2), 6.2.3 (/Category:V:Splunk:6.2.3), 6.2.4 (/Category:V:Splunk:6.2.4), 6.2.5 (/Category:V:Splunk:6.2.5), 6.2.6 (/Category:V:Splunk:6.2.6), 6.2.7 (/Category:V:Splunk:6.2.7), 6.2.8 (/Category:V:Splunk:6.2.8), 6.2.9 (/Category:V:Splunk:6.2.9) View the Article History (/index.php?title=Documentation:Splunk:SearchReference:ListOfDataTypes:4.1&action=history) for its revisions.