transaction

Synopsis

Displays a flow of correlated events from a single or multiple log sources.

Syntax

`transaction ("STEP_I_QUERY", "CORRELATION_I_ID", "STEP_I_NAME"->"STEP_II_QUERY", "CORRELATION_II_ID", "STEP_II_NAME"->...->"STEP_N_QUERY", "CORRELATION_N_ID", "STEP_N_NAME")`

Required Arguments

Each step is represented by the following 3 arguments:

`STEP_I_QUERY = a search query that isolate all relevant events of a transaction step`

`CORRELATION_I_ID = the log field which is used to correlate an event of a step and the next step's event(s)`

`STEP_I_NAME = the name of the transaction step that will be presented in the results`

Optional Arguments

Transaction Time / *STEP_NAME->STEP_NAME* Time (calculates the time of a transaction or between 2 steps of the transaction)

transaction eventscount / *STEP_NAME eventscount* (calculates the events count of a transaction / step of the transaction)

limit time to X minutes/hours (limits the maximal time allowed from first event to last event in a given transaction - only events within the time limitation will be correlated)

use unique key (events used to open a transaction with the same key will be joined to the same transaction)

transaction fullstate = OPEN/CLOSE/PARTIAL CLOSE (OPEN = transactions the don't contain the closing events, CLOSE = transactions that contain closing events, TIME CLOSE = transaction which were closed because of a time limitation specified by "limit time to X hours" or closing events but missing some events internally)

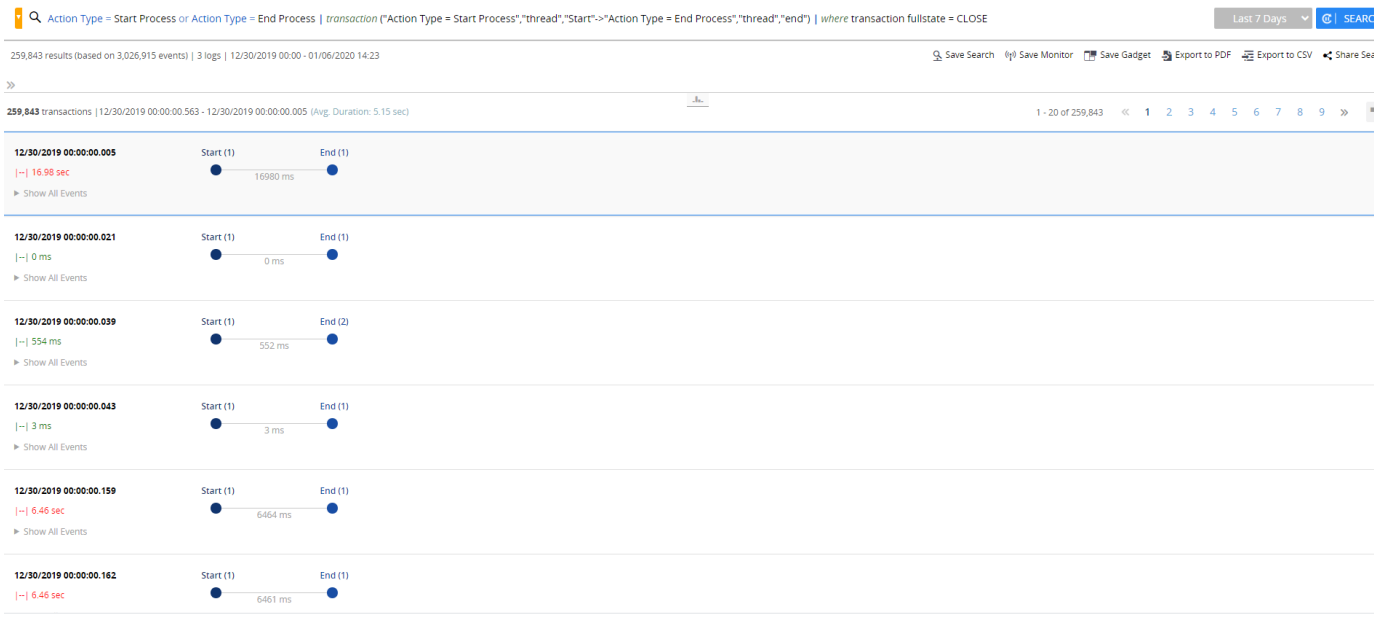
Description

Shows the un-formatted amount of time between the first and last event in a group. Should be formatted and displayed in time format.

Note: By default, the Search colors matching events, step(s) which could not be correlated will be grayed. The time of each specific transaction (marked in green = faster or equals to the average time, marked in red = slower than the average time) and the time taken between each two transaction steps is presented on the mapped transactions.

To see a transaction's events, click the 'Show All Events' link.

Example:



Examples

Example 1:

```
* in log.ORDER_FLOW | transaction
("requesting","TXID","Request"->"authorized","TXID","Authorization"->"dispensing","TXID","Dispense"->"ready
transaction","TXID","Ready"->"end of","TXID","Completed")
```

Displays the correlated transaction from the log ORDER_FLOW based on the correlation ID (log field) - TXID.

Example 2:

```
* in log.LOG_1, LOG_2, LOG_3 | transaction ("start transaction in log.LOG_1","TXID","Start"->"processing
transaction in log.LOG_2","TXID","Processing"->"transaction completed in log.LOG_3","TXID","End")
```

Displays the correlated transaction from the logs LOG_1, LOG_2, LOG_3 based on the correlation IDs (log fields) - TXID.

Example 3:

```
* in log.ORDER_FLOW | transaction
("requesting","TXID","Request"->"authorized","TXID","Authorization"->"dispensing","TXID","Dispense"->"ready
transaction","TXID","Ready"->"end of","TXID","Completed") | avg transaction Time, max transaction Time, min
transaction Time | display avg as Average Tx Time in time format, min as Fastest Tx Time in time format,
max as Slowest Tx Time in time format
```

Displays the average, minimum and maximum transaction time.

Example 4:

```
* in log.ORDER_FLOW | transaction
("requesting","TXID","Request"->"authorized","TXID","Authorization"->"dispensing","TXID","Dispense"->"ready
transaction","TXID","Ready"->"end of","TXID","Completed") | count | interval 5 minute | show count as
Transactions Over Time
```

Displays the number of transactions that were correlated in 5 minutes time bucketing.

Example 5:

```
* in log.ORDER_FLOW | transaction
("requesting","TXID","Request"->"authorized","TXID","Authorization"->"dispensing","TXID","Dispense"->"ready
transaction","TXID","Ready"->"end of","TXID","Completed") | where transaction time > 500 | order by
transaction time desc
```

Displays all transactions that their total time to be completed took more than 500 milliseconds (result will be sorted in a descending order).

Example 6:

```
* in log.ORDER_FLOW | transaction
("requesting","TXID","Request"->"authorized","TXID","Authorization"->"dispensing","TXID","Dispense"->"ready
transaction","TXID","Ready"->"end of","TXID","Completed") | avg request->authorization time, max
request->authorization time, min request->authorization time | display avg as Average Request>Authorization
in time format, max as Slowest Request>Authorization in time format, min as Fastest Request>Authorization
in time format
```

Displays the average, minimum and maximum time of the time taken between the transaction's steps Request to Authorization.

Example 7:

```
* in log.ORDER_FLOW | transaction
("requesting","TXID","Request"->"authorized","TXID","Authorization"->"dispensing","TXID","Dispense"->"ready
transaction","TXID","Ready"->"end of","TXID","Completed") | where transaction contains exception
```

Displays only transactions that contain exception in one or more of it's log events.

Example 8:

```
* in log.ORDER_FLOW | transaction
("requesting","TXID","Request"->"authorized","TXID","Authorization"->"dispensing","TXID","Dispense"->"ready
transaction","TXID","Ready"->"end of","TXID","Completed") | avg authorization eventscount
```

Displays the average number of events in the 'Authorization' transaction step.

Example 9:

```
* in log.ops | transaction ("start","TXID_1+TXID_2","Start"->"end","TXID_1+TXID_2","End")
```

Displays transactions which were correlated by using a combination of 2 log fields TXID_1 and TXID_2.

Example 10:

```
* in log.ORDER_FLOW | transaction ("requesting","TXID","Request"->"end of","TXID","Completed") use unique
key
```

Displays the correlated transaction from the log ORDER_FLOW based on the correlation ID (log field) - TXID (events used to open a transaction with the same key will be joined to the same transaction).

Example 11:

```
* in log.ORDER_FLOW | transaction ("requesting","TXID","Request"->"end of","TXID","Completed") limit time
to 10 minutes
```

Displays the correlated transaction from the log ORDER_FLOW based on the correlation ID (log field) - transactions that are not closed within 10 minutes will be considered as TIME CLOSED and not CLOSED