

# Suggestions for Additions to Tableau Log Events

**Author:** Sara Alspaugh

**Affiliation:** University of California, Berkeley

**Contact:** [alspaugh@eecs.berkeley.edu](mailto:alspaugh@eecs.berkeley.edu)

**Last modified:** 22 December 2014

## Introduction

This document describes the results of an assessment we performed on a set of Tableau logs that we collected, which documented sessions in which we asked a small number of users to perform exploratory data analysis on some pre-defined tasks with pre-defined questions. Our goal is to use the Tableau event logs to try to understand user analysis activity, by reconstructing user actions and the Tableau interface state from the event stream. We realize that the log files were probably designed for some other purposes, such as debugging the system, and so it is not surprising that some of the information we would like to see in the logs are not present. The purpose of this document is to explain what changes we would like to see in the logs to better support understanding user behavior.

## Explanation of contents

There are three main sections to this document following this one. The first section, **Events that are missing information**, contains a list of currently logged Tableau events that are missing information and a description of what information needs to be added. The second section, **Interactions that are not logged**, contains a list of information that is not currently logged, but might be useful to log. The third section, **Miscellaneous suggestions**, contains some other suggestions that would make it easier to make use of the Tableau logs.

The information contained in this document pertains to Tableau version 8.2.

The events that we focus on are those that are keyed `command-pre` (i.e., `"k": "command-pre"`) because we have surmised that these events are the only ones that reflect user interaction that is relevant to our purposes. We believe these events to be almost identical to `"k": "command-post"` events with the exception of the timestamp (`ts`) and key (`k`), so if one of these events is changed, it may be desirable that the other is changed as well.

We have noted that one consistently missing piece of information is the content of the visualization after a user takes an action that would change the visualization.

This is a work-in-progress document. Not all Tableau events or interactions have been evaluated for completeness for our purposes. Events may be added to the list -- when they are, they will be marked as "NEW", accompanied by the date added.

### **Summary of assumptions**

- All relevant events are contained in the log.txt file contained in the Tableau log directory.
- The events are logged from Tableau version 8.2 or versions with a comparable logging format
- The only type of events which reflect user interaction of interest are "k": "command-pre" events.
- The "k": "command-pre" events are all nearly identical to "k": "command-post" events in terms of parameter information; only things like timestamp (ts) and key (k) information differ.
- There is no "verbose mode" or other means to induce Tableau to log more information than is done by default.
- It is acceptable to log the visualization produced in response to relevant interactions, though this is not usually done.

## **Contents**

### **Events that are missing information**

1. Create calculation
2. Change aggregation
3. Drag and drop field
4. Add field to sheet via right click
5. Add field to sheet via double click
6. Remove fields
7. Clear shelf
8. Show me
9. Undo
10. Duplicate sheet
11. Delete sheet
12. Clear sheet
13. Copy data from sheet
14. Paste
15. Drop sheet on dashboard

### **Interactions that are not logged**

1. Switch sheet
2. Menu open

### **Miscellaneous suggestions**

- It may be helpful to at some point generate a guide that describes what physical interactions with Tableau generate which events (as identified by their key).

## Events that are missing information

### Create calculation

**Event key:** `tabui:create-calculation-ui`

**Example event:**

```
{"ts":"2014-09-03T18:13:31.666","pid":13594,"tid":"6a7af","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:create-calculation-ui","args":"tabui:create-calculation-ui use-selector=\"true\"\"}}
```

**Interaction description:** A calculated field, based on functions of other fields, is created. The corresponding event notes that the calculation dialog was used but not how the field was calculated. This calculated field is also assigned an ID (based at least in part on the timestamp at the time of creation) that is used to refer to it in later events, such as `Calculation_6530903181331686`, which is, for example, referred to in the following `tabui:drop-ui` event:

```
{"ts":"2014-09-03T18:16:04.019","pid":13594,"tid":"6a7af","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:drop-ui","args":"tabui:drop-ui drag-description=\"\" drag-source=\"drag-drop-schema\" drop-target=\"drag-drop-viz\" field-encodings=[{\"fn\":\"[On_Time_On_Time_Performance_2001_9 Extract].[sum:Calculation_6530903181331686:qk]\", \"encoding-type\":\"invalid-encoding\"}] is-copy=\"false\" is-dead-drop=\"false\" is-right-drag=\"false\" shelf-drag-source-position={\"is-override\": false} shelf-drop-context=\"none\" shelf-drop-target-position={\"shelf-type\":\"rows-shelf\", \"shelf-pos-index\": 2, \"is-override\": false} target-sheet=\"Sheet 1\"\"}}
```

**Missing information to add:**

- The name given to the calculated field
- The formula used to compute the field

### Change aggregation

**Event key:** `tabui:change-aggregation-ui`

**Example event:**

```
{"ts":"2014-12-21T21:54:43.829","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:change-aggregation-ui","args":"tabui:change-aggregation-ui aggregation=\"count\"\"}}
```

**Interaction description:** An aggregated field (one that is labeled as AGG(Field), for example) is right-clicked, "Measure (Aggregation)" is highlighted and then an aggregation is selected. The event does not note what field the aggregation was changed on or what the aggregation was previously.

**Missing information to add:**

- The field for which the aggregation was changed
- The original aggregation that was changed from

## Drag and drop field

**Event key:** tabui:drop-ui

**Example event:**

```
{ "ts": "2014-12-22T14:41:13.896", "pid": 96633, "tid": "91cc3c", "sev": "info", "req": "-","sess": "-", "site": "-", "user": "-", "k": "command-pre", "v": { "name": "tabui:drop-ui", "args": "tabui:drop-ui drag-description=\"\" drag-source=\"drag-drop-schema\" drop-target=\"drag-drop-viz\" field-encodings=[{ \"fn\": \"[textscan.41978.672928009300].[none:columns:ok]\", \"encoding-type\": \"invalid-encoding\"}] is-copy=\"false\" is-dead-drop=\"false\" is-right-drag=\"false\" shelf-drag-source-position={ \"is-override\": false } shelf-drop-context=\"none\" shelf-drop-target-position={ \"shelf-type\": \"rows-shelf\", \"shelf-pos-index\": 0, \"is-override\": false } target-sheet=\"Sheet 26\"\" } }
```

**Interaction description:** A field is dragged and dropped from one location to the other. The corresponding event describes a lot of information but one thing that is missing is a description of what visualization is generated in response to the action.

**Missing information to add:**

- What visualization is shown in response to the current visualization

## Add field to sheet via right click

**Event key:** tabui:add-to-sheet-ui

**Example event:**

```
{ "ts": "2014-12-22T12:33:23.211", "pid": 96633, "tid": "91cc3c", "sev": "info", "req": "-","sess": "-", "site": "-", "user": "-", "k": "command-pre", "v": { "name": "tabui:add-to-sheet-ui", "args": "tabui:add-to-sheet-ui schema-item-type=\"categorical\"\" } }
```

**Interaction description:** A field in the left-hand side bar (i.e., the schema) is right-clicked and then "Add to Sheet" is selected. The event does not say which field was added to the sheet, what sheet it was added to, where it was placed, or what visualization was shown in response.

**Missing information to add:**

- What field was added to the sheet
- What sheet the field was added to
- Where the field was placed on the sheet
- What visualization was shown

## Add field to sheet via double click

**Event key:** `tabdoc:add-to-sheet`

**Example event:**

```
{"ts":"2014-12-22T12:36:00.900","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabdoc:add-to-sheet","args":"tabdoc:add-to-sheet\nfn=\"[textscan.41978.672928009300].[county]\\\"\"}}
```

**Interaction description:** A field in the left-hand side bar (i.e., the schema) is double-clicked. This automatically adds the field to the sheet. The event does not note what sheet the field was added to, where it was placed on the sheet, or what visualization was shown.

**Missing information to add:**

- The sheet that the field was added to
- Where the field was placed on the sheet
- What visualization was shown

## Remove fields

**Event key:** `tabdoc:remove-fields`

**Example event:**

```
{"ts":"2014-12-21T21:42:10.882","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabdoc:remove-fields","args":"tabdoc:remove-fields"}}
```

**Interaction description:** A field on a sheet is right-clicked and "Remove" is selected from the menu. The corresponding event does not note what field was removed, where it was removed from, or what visualization was shown in response.

**Missing information to add:**

- The name of the field that is removed
- What sheet the field is removed from.
- What visualization was shown

**Clear shelf**

**Event key:** tabdoc:remove-fields-from-shelf

**Example event:**

```
{"ts":"2014-12-22T13:55:51.088","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabdoc:remove-fields-from-shelf","args":"tabdoc:remove-fields-from-shelf shelf-type=\"filter-shelf\"\"}}
```

**Interaction description:** The small downward pointing arrow next to a shelf name is clicked and "Clear Shelf" is selected from the menu. The event does not note which sheet this occurs on or what visualization was shown in response.

**Missing information to add:**

- The sheet fields were removed from
- (Optionally) which fields were removed
- What visualization was shown

**Show me**

**Event key:** tabdoc:show-me

**Example event:**

```
{"ts":"2014-12-22T13:36:16.641","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabdoc:show-me","args":"tabdoc:show-me show-me-command-type=\"filled-maps\" worksheet=\"Sheet 26\"\"}}
```

**Interaction description:** The "Show Me" tab is used to select a chart type. Upon selection, the visualization on the current sheet might change, including what fields are on what shelves. The event does not note the updated fields that are then present on the sheet or where they are located.

**Missing information to add:**

- What fields are present on the sheet after the *Show Me* command (including the aggregation applied to the fields, if relevant)
- The locations of the fields on the sheet after the *Show Me* command

## Undo

**Event key:** `tabdoc:undo`

**Example event:**

```
{"ts":"2014-12-22T13:47:41.749","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabdoc:undo","args":"tabdoc:undo"}}
```

**Interaction description:** The *back* button is selected to undo the last action on the stack. The event does not note what action was undone. This is not strictly necessary but would be useful.

**Missing information to add:**

- What action was undone.

## Duplicate sheet

**Event key:** `tabui:duplicate-sheet-ui`

**Example event:**

```
{"ts":"2014-12-21T21:49:49.086","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:duplicate-sheet-ui","args":"tabui:duplicate-sheet-ui"}}
```

**Interaction description:** A sheet tab along the bottom of the interface is right-clicked and "Duplicate Sheet" is selected from the menu. The corresponding event does not note what sheet was duplicated or what name the new sheet was given.

**Missing information to add:**

- What sheet was duplicated
- The name given to the new duplicate sheet

## Delete sheet

**Event key:** `tabui:delete-sheet-ui`

**Example event:**

```
{"ts":"2014-12-21T21:46:44.098","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:delete-sheet-ui","args":"tabui:delete-sheet-ui"}}
```

**Interaction description:** A sheet tab along the bottom of the interface is right-clicked and "Delete Sheet" is selected from the menu. The corresponding event does not note what sheet was deleted.

**Missing information to add:**

- The name of the deleted sheet

## Clear sheet

**Event key:** `tabui:clear-sheet-ui`

**Example event:**

```
{"ts":"2014-12-22T13:51:55.184","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:clear-sheet-ui","args":"tabui:clear-sheet-ui"}}
```

**Interaction description:** The "Clear Sheet" button (i.e., with the red "X") at the top of the interface is clicked, removing all fields from the current sheet. The event does not note what sheet was cleared.

**Missing information to add:**

- What sheet was cleared

## Copy data from sheet

**Event key:** `tabui:copy-data`

**Example event:**

```
{"ts":"2014-12-22T12:22:02.121","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:copy-data","args":"tabui:copy-data"}}
```

**Interaction description:** The keyboard shortcut for copy (i.e., command-c on OSX) is used when a sheet is selected. The event does not specify what sheet was selected when this occurs.

**Missing information to add:**

- The sheet that was selected to copy data from



## Paste

**Event key:** `tabui:paste`

**Example event:**

```
{"ts":"2014-12-22T12:21:32.646","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabui:paste","args":"tabui:paste"}}
```

**Interaction description:** The keyboard shortcut for paste (i.e., command-v on OS X) is used. The event does not specify what was pasted or where it was pasted.

**Missing information to add:**

- What was pasted
- Where it was pasted to

## Drop sheet on dashboard

**Event key:** `tabdoc:drop-on-dashboard`

**Example event:**

```
{"ts":"2014-12-22T11:37:55.455","pid":96633,"tid":"91cc3c","sev":"info","req":"-","sess":"-","site":"-","user":"-","k":"command-pre","v":{"name":"tabdoc:drop-on-dashboard","args":"tabdoc:drop-on-dashboard add-as-floating=\"false\" drop-location={\"x\": 379,\"y\": 309} worksheet=\"Total errors by county\" zone-type=\"viz\""}}
```

**Interaction description:** When a dashboard sheet is active, a visualization sheet is dragged from the left side-bar to the main area and dropped. The dashboard the sheet is added to is not noted.

**Missing information to add:**

- Which dashboard the sheet is added to

## Interactions that are not logged

### Switch sheet

**Interaction description:** A sheet is selected via clicking on the tabs at the bottom of the interface or using keyboard shortcuts to navigate between tabs. There is currently no event for this as far as we can tell. This information would suffice in place of additional parameters for many of the other events that act on the

current sheet and do not specify what sheet was acted on, because we would know what the current sheet is.

**Missing information to add:**

- An event describing what sheet was switched to

## **Menu open**

**Interaction description:** A user clicks on an interface element that pops up a menu list of possible options. Usually, when a user clicks on one of the options, an event is logged. However, it would be useful to know that a user opened a menu even if they selected no action, because this would allow the detection of "menu surfing".

**Missing information to add:**

- An event describing what menu was opened