Implementing new Mixpanel Events

**Document created by**: Wic Verhoef

**Date of last revision**: 21 Jan 2016

**Purpose**

This document describes the principles of Mixpanel event specification.

Table of Contents

Quick Summary 1

Mixpanel? 2

Event Parts 2

Which events should be tracked? 3

Event Naming 3

Event Properties 4

More Questions? 6

# Quick Summary

If you are relatively comfortable with the concepts of “Mixpanel”, “Event Property”, this summary is for you. I

1. **More tracking is better than less tracking.** More is better than fewer as switching off some events later on is easier than adding more events. The idea is to **get an MP event behind each and every clickable element of our software**. ~~Less~~ *More* is more.
2. **This event that you are so caringly adding should be distinguishable from other events**. This can be done by naming it something new or by setting one of the constituent properties to denote the difference between this event and the other ones in existence.
3. **Good event names make sense.** At the time of writing there are in excess of 600 MP events. Having event names that resemble the event’s purpose helps a great deal in figuring out what they do.
4. **Be sure that the event can be linked to a user**: in the Connect World, the user’s email address is automatically added. In the Pro/Desktop world, this is not always the case. If the event cannot clearly be linked to a user, the usefulness of the event is rather low.
5. **The Analytic folk are super sensitive about *property* names.** *Super* sensitive. For the sake of brevity, suffice to say that adding brand new property names breaks stuff. We have about 300 property names currently. Some are duplicates (“File ID” vs “file\_id”) and adding more props does not help the situation. Please see the “What properties should I add to events?” section for nice concise rules on the property aspect.

# Mixpanel?

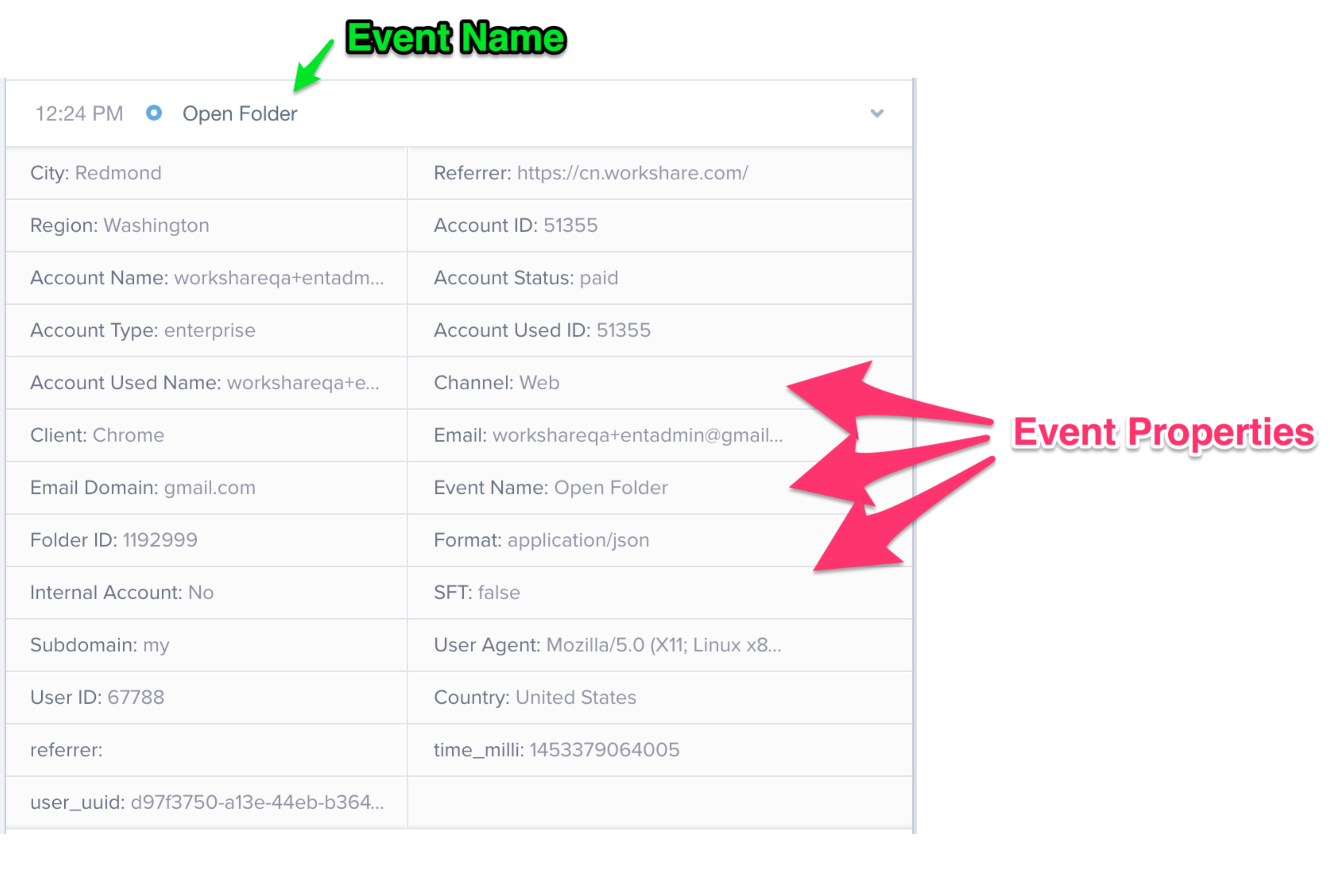
Mixpanel (MP) is an event tracking and analytics service in the Cloud. Every Workshare product sends usage events to Mixpanel. For example, if a user opens a file in Connect or runs a comparison in Professional, these events are sent to Mixpanel and recorded there. By having these events recorded, we can analyse user behaviour to determine if and how they are using our features. This insight is then used to inform product design for existing and future features.

# Event Parts

Each MP event has two distinct parts:

* Event Name: this is the event’s name (yeah). It should be descriptive of the action that the user took to trigger the particular event. Examples of currently-implemented event names are “Open Folder”, “Upload File” and “web.folder.user.view\_folder”
* Event Properties: additional bits of information that’s sent with the event. Examples of properties are “Channel” and “Email”

When it comes to specifying a new MP event, you have complete control over the event name and most of the event properties.



# Which events should be tracked?

More is better than fewer as switching off some events later on is easier than adding more events. The idea is to **get an MP event behind each and every clickable element of our software**. ~~Less~~ *More* is more.

# Event Naming

At the time of writing there are oodles of historically implemented Mixpanel (MP) events. The idea is not (currently) to change the naming of already implemented MP events. This document deals with the naming of new events.

General principles:

1. if this is dealing with the naming of the new WS Desktop app’s events:
   1. as the new WSD is mostly just a rendering of the website, please use the same naming for WSD events as we have for the Web. The only difference should be in the “Channel” property.
2. If it has to do with a brand new development that’s not part of WSD or does not have an existing Web counterpart, please follow the naming convention described below

The pedantic stuff:

* Please stick to lower case and
* separate\_words\_by\_an\_underscore

The Analytics guys are not that sensitive about the event name, but so as to ensure that we all play nicely, there’s a bits&bobs section below this one that talks about the event naming;

**Bits and bobs of naming an event**

For events recorded in the front-end:

* Events recorded in the front-end are triggered by the user performing an action, for example clicking a button on a web page or tapping a button on the mobile app
* Synopsis for front end events is: **<product>.<section>.<event-name>**, where:
  + *Product* is one of “**web**” for Workshare Connect (online), “**mobile**” for the mobile app, “**desktop**” for the desktop sync app, and “**professional**” for any professional suite product (Compare or Protect)
  + *Section* is the section of the user interface where the event was triggered. For Workshare Connect Online, please use sections as per the left navigation bar: “**home**” for any event on the home page, “**groups**” for Groups, “**myfiles**” for My Files, “**fifo**” for Files and Folders, **“groups”** for groups, “**transact**” for Transact, and “**people**” for People
  + *Event name* is your choice, please make it as easy to understand as possible. It can be broken down into further sub-sections if necessary (see home page event example below). When the event corresponds to displaying a screen, please use the “**screen**” descriptor (see people page event example below).
* Examples:
  + web.people.click: triggered when a user clicks on the People navigation bar icon and the People page is displayed
  + web.home.card.click: triggered when a user clicks on a home page card
  + mobile.files.sort: triggered when a user taps on the button to sort files in a folder in the Mobile app.

For events recorded in the back-end:

* Events recorded in the back-end are triggered on our servers, after a user has performed an action in one of our apps. Typically, we record a back-end event when it could be triggered by multiple apps. For example, uploading a file could be done via the web app, mobile app or desktop app.
* Synopsis for back-end events is: **<entity>.<event-name>**, where:
* *Entity* is the entity affected by the event, for example **groups**, **files**, **folders**. Please use plural (i.e. group**s** as opposed to group)
* *Event name* is your choice, please make it as easy to understand as possible. It can be broken down into further sub-sections if necessary.
* Examples:
  + groups.view\_posts: triggered when the user views a group’s posts
  + files.multiple\_dowload: triggered when a user downloads multiple files.

**Note:** these are the most up-to-date conventions that we are following when naming new events. Since we did not have these conventions in place when we started using Mixpanel, many of the older events do not follow the convention.

# Event Properties

**As alluded to before, the Analytics gents are rather precious about property names.**

Each event sent to Mixpanel can have any number of properties with additional information about the event.

For example, the web.home.cards.click event includes a property called Action\_Value which can have the following values:

* Open\_Link: if the user clicked on the main link that appears on the home page card
* View\_File: if the user clicked on the file name link
* View\_Group: if the user clicked on the location (group or shared folder) link

Similarly to the Action\_Value property, you can design any property that captures additional information about the event. Each event already has a number of properties added automatically, so please ensure that you do not repeat information in your custom properties. Properties that are automatically added to Mixpanel events include (not an exhaustive list):

* Email: email address of the user who generated the event
* Channel: where the event was originated: Web, Mobile Web, Mobile App, Desktop, Professional
* Account Name: name of the account the user belongs to (e.g. Workshare)
* Account ID: internal ID of the account
* User Agent: raw identification of the browser used to generate the event. Workshare apps (e.g. Compare, Protect) generate their own User Agent string
* ProductName: name of the product used to generate the event (e.g. Professional, Protect)
* ProductVersion: version of the product used to generate the event (e.g. 9.0.0.2300)
* Internal Account: “Yes” if the event was generated by a Workshare user (either on the Workshare account, or on any of our multiple testing accounts), “No” if a customer account.

The Analytics geeks have this thing they call a data warehouse and this beast is fed off of MP event data. Currently the DWH only use the properties listed below, so please use these when creating new events:

|  |
| --- |
| DateTime |
| App ID |
| App |
| Channel |
| channel |
| Event Name |
| Client |
| Feature |
| InstallType |
| Page Name |
| ProductInstallType |
| ProductName |
| Subdomain |
| action\_name |
| Action\_Type |
| action\_value |
| $app\_version |
| app\_version |
| ProductVersion |
| version |
| account |
| Account ID |
| Account Name |
| Account Used ID |
| Account Used Name |
| Internal Account |
| Account Status |
| Account Type |
| User ID |
| user\_id |
| user\_uuid |
| UserId |
| distinct\_id |
| MachineId |
| MachineLocale |
| Email |
| email |
| user\_email |
| Email Domain |
| mp\_country\_code |
| $city |
| $region |
| LicenceAuthCode |
| Name |
| User Agent |
| Action\_Value |
| File ID |
| Version |
| File\_Type |
| AD email |

If your new, shiny, bespoke MP event has a brand new property that really, really does not fit into any of the 50 above, please come and talk to Wic/Renatobefore adding it.

# More Questions?

Please ask Wic or Renato.