

# **Module 3 - Analytics implementation and configuration**

This toolkit is designed for Professional Developer Exam Aspirants. There are **six** Modules. Study Each module per week to stick to schedule. Technical Parts of applications are depicted in Videos, you can learn more about them from experience League. You can visit [Get prep page](#) to understand the contents and anticipate the learning journey.

This is Professional Exam, Developer toolkit Module 3. This module contains four sections.

## **3.1 Analytics Tool Guide**

The Analytics Tools Guide includes information about product features, use cases, task instructions, and best practices for the following Analytics tools:

| <b>Tool</b>                                 | <b>Description</b>  |
|---|---|
| <a href="#"><u>Analysis Workspace</u></a>   | Analysis Workspace is a flexible browser tool that allows you to quickly build analyses and share insights. Using the drag-and-drop interface, you can craft your analysis, add visualizations to bring data to life, curate a dataset, share and schedule projects with anyone in your organization.   |
| <a href="#"><u>Analytics dashboards</u></a> | Analytics dashboards and their mobile scorecards allow executive users to view a broad rendering of important summary data quickly and easily on their own mobile devices. Curators add visualizations to mobile scorecard projects and share them with executives.<br>Scorecards provide a way to target and measure KPIs and provide a clear indication of how well organizations are working to achieve their targets. |
| <a href="#"><u>Activity Map</u></a>         | Activity Map is an Adobe Analytics application that is designed to rank link activity using visual overlays and provide a dashboard of real-time analytics to monitor audience engagement of your web pages.<br>Activity Map lets you set up different views to visually identify the acceleration of customer activity, quantify marketing initiatives, and act on audience needs and behaviors."                        |
| <a href="#"><u>Report Builder</u></a>       | Report Builder is an add-in for Microsoft Excel. Report Builder lets you build customized requests from Adobe Analytics data that are inserted into your Excel worksheets. Requests can dynamically reference cells within your worksheet, and you can update and customize how Report Builder presents the data.   |
| <a href="#"><u>Analytics APIs</u></a>       | Analytics APIs allow you to directly call Adobe's servers to perform almost any action that you can perform in the user interface. You can create reports to explore, get insights, or answer important questions about your data. You  |

| Tool                                    | Description  |
|---|--|
|   | can also manage components of Adobe Analytics, such as the creation of segments or calculated metrics.   |
| <a href="#">Reports &amp; Analytics</a> | Reports & Analytics is a tool with dozens of pre-built reports and visualizations. These are now available within Analysis Workspace. <b>Effective December 31, 2023, Adobe intends to discontinue Reports &amp; Analytics and its accompanying reports and features. Adobe recommends using Analysis Workspace for most reporting needs.</b> For additional information, see <a href="#">Analysis Workspace</a> . |

For questions about which tool to use, see [Which Adobe Analytics tool should I use?](#).

## [3.2 Quick tip – Passing Data into Launch Rules](#)

### Quick tip – Passing Data into Launch Rules

SEPTEMBER 18, 2018 ~ JAN EXNER

If you haven't yet heard it: In Launch, by Adobe, the `_satellite.track()` method accepts two parameters: an Event name, plus payload data.

We used to work around this in the past when calling DCRs in DTM, storing the data somewhere in the DOM. My favourite method was the one first shown to me by my colleague [Pedro Monjo](#): create some weirdly-named object within the `_satellite` object, then attach your data there.

But now, we can pass the data straight into the call, just like that.

This is what we all always wanted!

And here is how you use it.

# Payload

The situation: say you're tracking when a visitor "moves to the next page" within your single-page application. Due to the way SPAs work, you have decided to explicitly call a Rule using `_satellite.track()`.

You want to obviously track *which* page is loaded, so you want to pass a page name into the call, plus maybe some other information — site section, page language, page author, content tags if the page has some, and so on.

This is what it would look like in your code:

```
_satellite.track('pseudoPageLoad', pageData);
```

That was easy!

You can also hard-code the payload, if you want, like so:

```
_satellite.track('pseudoPageLoad', {'pageName': 'Homepage', 'language': 'es'});
```

# Inside

Inside the Rule, you likely want to pass the data into variables. As if by magic, your payload can be accessed, both in code and as a Data Element.

In code, you have access to `event.detail`, which holds a structure that reflects what you sent. And if you want to use it directly in an Action, you can access `%event.detail%`.

If we take the hard-coded example from above, then you can use the following in the field to set Page Name: `%event.detail.pageName%`, and for — say — `eVar2`, you type `%event.detail.language%` into the field.

Note that `event.detail` contains exactly what you pass into `_satellite.track()`. If you pass a structure — like in the example above — you get a structure. If you pass a simple string, you'll get a string.

Awesome, isn't it?

Btw: I am (pain-) fully aware that [Jenn Kunz](#) has written about exactly the same thing (see [Direct Call Rules in Launch have a new power: passing additional info in satellite.track](#)), but I refuse to throw away a perfectly valid article, just because it is no longer useful. Worse: this was originally scheduled for Sep 4, but after my blunder with the other article, I swapped it. Sigh.

Oh: Jenn is right, this works in DTM, too.

### 3.3 [transactionID](#)

The `transactionID` variable uniquely identifies a transaction so the hit can tie to data uploaded through Data Sources. This variable is valuable in cases where you want to use data from other channels and link it to data collected with AppMeasurement.

#### **NOTE**

Make sure that Transaction ID Storage is enabled in a report suite before using this variable. See [General Account Settings](#) in the Admin user guide for more information.

When you set `transactionID` on a hit, Adobe takes a “snapshot” of all Analytics variables set or persisted at that point in time. Data uploaded through Data Sources with a matching transaction ID is permanently tied to those variable values.

By default, Adobe remembers all transaction ID values (linked and unlinked) for up to 90 days. If your offline interaction process is longer than 90 days, contact Customer Care to have this limit extended.

#### Transaction ID using the Web SDK

Transaction ID is [mapped for Adobe Analytics](#) under the XDM field `commerce.order.transactionID`.

#### Transaction ID using the Adobe Analytics extension

You can set transaction ID either while configuring the Analytics extension (global variables) or under rules.

1. Log in to [Adobe Experience Platform Data Collection](#) using your AdobeID credentials.
2. Click the desired tag property.
3. Go to the Rules tab, then click the desired rule (or create a rule).
4. Under Actions, click an existing Adobe Analytics - Set Variables action or click the '+' icon.
5. Set the Extension drop-down list to Adobe Analytics, and the Action Type to Set Variables.
6. Locate the Transaction ID section.

You can set transaction ID to any string value, including data elements.

### s.transactionID in AppMeasurement and the Analytics extension custom code editor

The `s.transactionID` variable is a string containing a unique identifier for a transaction. Valid values include alphanumeric characters up to 100 bytes in length. Its default value is an empty string.

```
s.transactionID = "ABC123";
```

Copy

Toggle Text Wrapping

If you have more than one transaction ID for a hit, you can delimit each with a comma. Multiple transaction IDs are still subject to the 100-byte limit.

```
s.transactionID = "ABC123,XYZ456";
```

Copy

Toggle Text Wrapping

#### **NOTE**

If you integrate multiple offline channels using this variable, make sure that different channels don't overlap transaction IDs. For example, if you have a call center transaction ID value of `1234` and a sales lead transaction ID value of `1234`, they can conflict and cause unexpected results. Make sure that transaction IDs contain unique formats per offline channel and differentiate them if necessary. For example, set your call center transaction ID to `call_1234` and your sales lead transaction ID `lead_1234` in both Data Sources and AppMeasurement.

## **3.4 How analytics data is collected**

How does data get from your page into the reports in Adobe Analytics? Click the link above to view a video showing you how.

