

DUMPS ARENA

Adobe Analytics Architect Master

Adobe AD0-E207

Version Demo

Total Demo Questions: 7

Total Premium Questions: 73

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Analytics Strategy	8
Topic 2, Analytics Architecture	11
Topic 3, Analytics Implementation	36
Topic 4, Analytics Administration	18
Total	73

QUESTION NO: 1

A company uses tracking codes in the format `email_spring_2025`, `paidsearch_brand_2025`, and `affiliate_partnerA_2025`.

Analysts need a new reporting dimension named Marketing Tactic that groups both historical and future Tracking Code values into Email, Paid Search, and Affiliate. The original Tracking Code values must remain available for analysis.

Which approach should the Architect recommend?

- A. Create a Tracking Code classification named Marketing Tactic and configure Classification Rule Builder rules for each prefix
- B. Create Processing Rules that overwrite Tracking Code with the applicable marketing tactic
- C. Create a virtual report suite for each marketing tactic
- D. Create a calculated metric for each marketing tactic

ANSWER: A

Explanation:

Create a classification for the Tracking Code dimension and use Classification Rule Builder to assign Marketing Tactic values from the code prefixes. A classification adds a reporting relationship to the existing Tracking Code values without replacing or modifying the original collected values. Once classification processing completes, the grouped classification can be used for historical and future Tracking Code reporting.

Processing Rules can alter incoming data but do not retroactively populate a new grouping for historical values. A virtual report suite restricts the data available to users; it does not create a derived classification dimension.

QUESTION NO: 2

An Architect must support two recurring data-delivery requirements:

- A data science team needs hit-level data, including visitor identifiers and the sequence of internal-search interactions.
- A regional business team needs a scheduled weekly aggregate report of search terms, orders, and revenue.

Which two Adobe Analytics extraction methods should the Architect use? (Choose two.)

- A. Data Feeds for the data science team
- B. Data Warehouse for the regional business team
- C. Analysis Workspace CSV exports for the data science team
- D. Data Sources for the regional business team
- E. Processing Rules for the regional business team

ANSWER: A B

Explanation:

Data Feeds provide hit-level files that can be processed downstream to retain visitor-level identifiers, hit sequence, and individual search interactions. Data Warehouse is appropriate for a scheduled aggregate report because it can produce a report with selected dimensions, metrics, date ranges, and granularity for delivery on a recurring schedule.

An Analysis Workspace CSV export is generally a manual report export rather than a managed data-delivery design. Data Sources import data into Adobe Analytics and do not extract Analytics reporting data.

QUESTION NO: 3

While preparing for a new analytics implementation for a site, you begin conducting stakeholder interviews. Part of the conversation includes defining KPIs, including custom success events.

Which are two examples of custom success events on a site? (Choose two.)

- A. cart adds
- B. internal searches
- C. checkouts
- D. form completions
- E. purchases

ANSWER: B D

Explanation:

“internal searches” and “form completions” are appropriate examples of custom success events because they represent meaningful, business-defined visitor actions that an implementation can measure using Adobe Analytics custom event variables. During stakeholder interviews, these actions can be identified as KPIs when they indicate engagement, self-service success, lead generation, or progression toward a business outcome. For example, an internal-search event can be sent whenever a visitor submits a site-search query, enabling reporting on search usage and its relationship to conversion. A form-completion event can be sent when a visitor successfully submits a lead, registration, support, or other business form, allowing the organization to measure completed actions rather than merely form views.

Adobe Analytics supports custom events through the `event1` through `event1000` variables. An architect defines each event's name and behavior in the report suite's Success Events settings, then ensures the tagging implementation sends the designated event when the relevant action occurs. This makes custom events flexible enough to align reporting directly with the organization's stated KPIs. See Adobe's [Success events documentation](#) and its [events implementation overview](#).

QUESTION NO: 4

During a website audit, an Architect finds that the checkout pages on the site have been redesigned to include a tag management system that deploys Adobe Analytics.

The rest of the site has Adobe Analytics hard-coded onto the site. Both implementations use the same AppMeasurement version and are on the same domain.

How will this approach affect the reporting?

- A. eVars will not persist as a user moves to the checkout section.
- B. As visitors move to checkout, they will be tracked as a different visitor.
- C. Checkout will be considered a landing page.
- D. There will be inconsistencies in the data due to challenging upkeep.

ANSWER: D

Explanation:

There will be inconsistencies in the data due to challenging upkeep. Using both hard-coded AppMeasurement and a tag-management deployment can technically send data to the same Adobe Analytics report suite, but it creates two separate implementation locations that must remain aligned over time. The Architect must ensure that each deployment uses identical tracking-server settings, report-suite configuration, variable mappings, link-tracking behavior, consent handling, page-view timing, and any custom plug-ins or processing assumptions. A future change made in only one implementation path can

cause checkout hits to be collected differently from the rest of the site, reducing consistency in dimensions, metrics, and analysis.

Adobe recommends managing implementation settings consistently through its supported implementation tools and maintaining a documented solution design reference so variables and tracking behavior are governed uniformly. The fact that both areas use the same AppMeasurement version and domain helps support continuity, but it does not eliminate the operational risk of maintaining parallel implementations. See Adobe's [Analytics implementation documentation](#) and its guidance on creating a [solution design reference](#).

QUESTION NO: 5

A company has a website with many forms.

The company wants the ability to report on:

- Form errors per visit
- Form errors per form
- Successful form submissions

Which three variables should be used to meet these requirements? (Choose three.)

- A. Event for "Forms Submitted with Errors"
- B. s.pageType="errorPage"
- C. Event for "Visit Number"
- D. eVar for "Form Type"
- E. Event for "Forms Submitted Successfully"

ANSWER: A D E

Explanation:

"Event for \"Forms Submitted with Errors\"", "eVar for \"Form Type\"", and "Event for \"Forms Submitted Successfully\"" provide the required implementation pattern. A custom success event for forms submitted with errors records each error-submission occurrence. That metric can be reported alongside the standard Visits metric to calculate or compare form errors per visit, and it can be used as a metric in reporting without treating visit number itself as an implementation event. Adobe Analytics custom events are designed to capture these site actions and can be configured as counters when each qualifying form attempt should increment the total.

The form-type eVar supplies the persistent dimensional context needed to break the error event down by the specific form. The implementation should set the eVar on the form interaction or submission hit where the error event is sent. Its expiration and allocation should be configured to match the organization's attribution needs, such as retaining the form identity through the relevant submission flow. Finally, a separate custom success event records completed submissions independently, enabling direct reporting of successful form submissions and supporting form-funnel analysis. Adobe documents custom-event configuration at [Custom events](#) and explains eVar dimensional attribution at [eVars](#).

QUESTION NO: 6

An Architect is using dynamic variables in an Adobe Analytics implementation. eVar32 is used to collect the current page URL.

Which syntax should be used?

- A. s.eVar32="D=g"
- B. s.eVar32="D=cp"

C. s.eVar32=*D=host'

D. s eVar32="D=url"

ANSWER: B

Explanation:

s.eVar32="D=cp" is the appropriate syntax because Adobe Analytics dynamic variables use the D= prefix to copy a value that Adobe has already populated for the same server call. The cp identifier represents the current page URL value. Assigning D=cp to eVar32 therefore copies the current page URL into that eVar without requiring the implementation to separately read and format the browser location value.

This approach is especially useful when the desired eVar value should match Adobe Analytics' page URL collection value on that hit. The assignment must use valid JavaScript property notation, including the period between s and eVar32, and matching quotation marks around the dynamic-variable expression. Dynamic variables are resolved by Adobe Analytics during processing, so the value is evaluated as part of the beacon rather than treated as the literal text D=cp.

Adobe documents the supported dynamic-variable syntax and the available source identifiers in its [Dynamic variables documentation](#). The meaning and collection behavior of the current page URL are detailed in the [Page URL documentation](#).

QUESTION NO: 7

For internal search terms, a company wants to give credit to the original keyword used to find a product. On the first visit, a customer searches for "Mobile" and views the Nebulous Pro.

During the second visit, the customer refines this search to "5G Mobile" and views the Nebulous Pro again. The customer then purchases the Nebulous Pro for \$200 on the third visit. The company wants "Mobile" to receive credit.

Which configurations should the Architect apply?

- A. Configure an eVar as Merchandising with Original Value (First) allocation and set the expiration to 3rd Visit.
- B. Configure an event to capture the revenue and enable the participation and set the expiration of the visit to 90 days.
- C. Configure an event as Merchandising event to capture the revenue and set the expiration of the visit to 30 days.
- D. Configure an eVar as a Merchandising Variable with Original Value (First) allocation and set the expiration to the purchase event.

ANSWER: D

Explanation:

Configure an eVar as a Merchandising Variable with Original Value (First) allocation and set the expiration to the purchase event. A merchandising eVar is designed to associate a merchandising attribute, such as an internal search term, with products and then persist that association for downstream success events. Using Original Value (First) allocation preserves the first qualifying value set for the relevant persistence period. Therefore, once the visitor searches for "Mobile" and views the Nebulous Pro, that original term remains the value eligible for merchandising attribution even when the visitor later searches for "5G Mobile" and views the same product again.

Setting expiration to the purchase event keeps the merchandising value available across the customer's subsequent visits until the transaction occurs. When the Nebulous Pro is purchased on the third visit, the purchase revenue can consequently be attributed to "Mobile," satisfying the stated first-touch internal-search credit requirement. This configuration also avoids relying on a fixed time-based visit duration, which would not directly align eVar persistence with the business event that concludes the attribution journey. Adobe describes merchandising eVars and their product-level attribution behavior in the [Merchandising eVars documentation](#); allocation and expiration behavior is covered in the [conversion variable administration documentation](#).