| Parameter | Analytics Variable | Report Populated | Description |
| --- | --- | --- | --- |
| aamlh | None | None | Audience Manager Location Hint (used in Experience Cloud Shared Profile integration) |
| aamb | None | None | Audience Manager Blob (used in Experience Cloud Shared Profile integration) |
| aid | None | None | Analytics visitor ID |
| AQB | None | None | Indicates the beginning of an image request. |
| AQE | None | None | Indicates the end of an image request, meaning the request was not truncated. |
| bh | None | Visitor Profile | Technology | Browser Height | Browser window height (in pixels) |
| bw | None | Visitor Profile | Technology | Browser Width | Browser window width (in pixels) |
| c | None | Visitor Profile | Technology | Monitor Color Depths | Color quality (in bits) |
| c.*[key]* | s.contextData | None | Key-values pairs are specified in one of the following formats:  <my.a>red</my.a>  or:  <my><a>red</a></my>  Each of these examples result in a context data value of my.a = red. Multiple key-value pairs can be specified.  In the query string, this context data variable would appear asc.my.a=red |
| c1-c75 | s.prop1-s.prop75 | All Custom Traffic reports | Traffic variables used in custom traffic reporting |
| cc | s.currencyCode | None | The type of currency used on the site |
| cdp | s.cookieDomainPeriods | None | Indicates the number of periods in a domain for cookie tracking; manually set. |
| ce | s.charSet | None | The character encoding of the image request |
| cl | s.cookieLifetime (s\_vi cookie lifetime in seconds) | None | The lifetime of the visitor cookie. |
| ch | s.channel | Site Content | Site Sections | The Site Sections variable used in traffic reporting |
| cp | Hit Type | Hit Type | Indicates whether the behavior is a result of direct interaction foreground or information the device is sending without direct interaction background. |
| ct | None | Visitor Profile | Technology | Connection Types | Connection Type (Modem, LAN, etc; can only populate in IE browsers) |
| D | dynamicVariablePrefix | None | See [Dynamic Variables](https://marketing.adobe.com/resources/help/en_US/sc/implement/dynvars_overview.html#concept_B016789733A94070A9EAB209EEC05262). |
| events or ev | s.events | Site Traffic | Purchases, Shopping Cart, Custom Events | The commerce and custom events that occurred on the page; used in conversion reports |
| g | None | None | The current URL of the page, up to 255 bytes. |
| -g | None | None | URLs longer than 255 bytes are split, with the first 255 bytes appearing in the g parameter, with the remaining bytes appearing later in the query string in the -g= query parameter. |
| h1-h5 | s.hier1-s.hier5 | Site Content | Hierarchy reports | Hierarchy variables; used in traffic reporting |
| hp | None | Visitor Profile | Visitor Home Page | Indicates if current page is browser's home page (Y or N; can only populate in IE browsers) |
| j | None | Visitor Profile | Technology | Javascript Version | Shows the current Javascript version installed (generally 1.x) |
| k | None | Visitor Profile | Technology | Cookies | Are cookies supported in the browser (Y, N or U) |
| l1-l3 | s.list1-s.list3 | Custom Conversion | A delimited list of values that are passed into a variable, then reported as individual line items for reporting. |
| mid | None | None | Experience Cloud Visitor ID |
| ndh | None | None | Indicates whether the image request originated from JS file (1 or 0) |
| ns | s.visitorNameSpace | None | Specifies what domain the cookies are set on |
| oid | s.objectID | Site Content | Links | ClickMap | Object identifier for last page; used in ClickMap |
| ot | None | Site Content | Links | ClickMap | Object tag name for last page; used in ClickMap |
| p | None | Visitor Profile | Technology | Netscape Plug-Ins | Semicolon delimited browser plug-in names |
| pageName (or gn) | s.pageName | Site Content | Pages | The page's designated name in reporting |
| pageType (or gt) | s.pageType | Site Content | Pages Not Foun | Indicates whether it is a 404 page or not (Either 'error' or blank) |
| pccr | None | None | Only occurs for new visitors; prevents infinite redirects (Always true) |
| pe | s.linkType | Site Content | Links | Exit Links, File Downloads, Custom Links | Determines the type of custom link hit fired |
| pev1 | None | Site Content | Links | Exit Links, File Downloads, Custom Links | URL the custom link hit occurred on |
| pev2 | None | Site Content | Links | Exit Links, File Downloads, Custom Links | Custom link friendly name |
| pev3 | None | All video reports | Used to track milestones in legacy video reporting; deprecated with v15 |
| pf | None | None | For Adobe use only. Do not alter. |
| pid | None | Site Content | Links | ClickMap | Page identifier for last page; used in ClickMap |
| pidt | None | Site Content | Links | ClickMap | Page identifier type for last page; used in ClickMap |
| products (or pl) | s.products | Products | Products | Products variable used in conversion reporting |
| purchaseID (or pi) | s.purchaseID | None | Used to deduplicate purchases, preventing revenue inflation |
| r | s.referrer | All traffic sources reports | Referring URL |
| s | None | Visitor Profile | Technology | Monitor Resolutions | Screen resolution (width x height) |
| server (or sv) | s.server | Site Content | Servers | The page's server; used in traffic reporting |
| state | s.state | Visitor Profile | Visitor State | Specifies the state as defined by the variable. |
| t | (automatic, sent with every hit that does not have a custom timestamp) | None | The t parameter is in the following format:  dd/mm/yyyy hh:mm:ss D OFFSET  Where D is a number in the range 0-6 specifying the day of the week, and OFFSETrepresents:  offset from GMT in hours \* 60 \* - 1  For example:  23/09/2016 14:00:00 1 420 |
| ts | timestamp | None | The custom timestamp calculated and sent in with the hit. Typically used for offline tracking. |
| v | None | Visitor Profile | Technology | Java | Java enabled (Y or N) |
| v0 | s.campaign | Campaigns | Tracking Codes | The campaign variable used in conversion reporting |
| v1-v75 | s.eVar1-s.eVar75 | All Custom Conversion reports | Conversion variables used in custom conversion reporting |
| vid | s.visitorID | None | The visitor's unique ID as set in the [visitorID](https://marketing.adobe.com/resources/help/en_US/sc/implement/visitorID.html" \l "concept_CD273CC915CC4ABD8F52E4209FF9557E" \o "Visitors can be identified by the visitorID variable or by IP address/User Agent.) variable. |
| vmk | s.vmk | None | Visitor migration key; used to migrate from third-party to first-party cookies. Deprecated. |
| vvp | s.variableProvider | None | Used in Genesis integrations |
| xact | s.transactionID | None | The transaction ID used to link online data to offline data |
| zip | s.zip | Visitor Profile | Visitor ZIP/Postal Code | Determines the zip code as defined by the variable |
| /5/ (for mobile protocol) or /1/ (for non-mobile protocol) in the image request URL. | None | None | Controls the order in which cookies and other methods are used to identify visitors. |

CONFIGURATION VARIABLES

Configuration variables set in the AppMeasurement.js.

Configuration variables control the way data is captured and processed in reporting. The most-common configuration variables that are typically set in the main global JavaScript (AppMeasurement.js). These variables can be set within the Reports & Analytics page-level code and links when appropriate.

Not all of these variables appear in the code by default when you generate code through the Admin Tool > Code Manager. Some of these configuration variables may not be applicable to your site's implementation needs.

Some of the goals of using these configuration variables are:

* Track multiple sites/domains.
* Use any currency on purchases.
* Capture data indifferent languages.
* Link tracking (number of downloaded files, links to external sites.
* Track custom links for unique purposes.

1. S.account

If sending to multiple report suites (multi-suite tagging), s.account may be a comma-separated list of values. The report suite ID is determined by Adobe.

Max Size : 40 Bytes

The report suite is the most fundamental level of segmentation in reporting. You can set as many report suites as your contract allows. Each report suite refers to a dedicated set of tables that are populated in Adobe's collection servers. A report suite is identified by the s\_account(Typed in console) variable in your JavaScript code.

The s\_account variable is normally declared inside the JavaScript file (s\_code.js). You can declare the s\_account variable on the HTML page, which is a common practice when the value of s\_account may change from page to page. Because the s\_account variable has a global scope, it should be declared immediately before including Adobe's JavaScript file. If s\_account does not have a value when the JavaScript file is loaded, no data is sent to Analytics.

Syntax and Possible Values

The report suite ID is an alphanumeric string of ASCII characters, no more than 40 bytes in length. The only non-alphanumeric character allowed is a hyphen. Spaces, periods, commas and other punctuation are not allowed. The *s\_account* variable may contain multiple report suites, all of which receive data from that page.

1. var s\_account="reportsuitecom[,reportsuite2[,reportsuite3]]"

All values of *s\_account* must be provided or approved by Adobe.

Examples

1. var s\_account="mycompanycom"
2. var s\_account="mycompanycom,mycompanysection"

Configuring the Variable in Analytics

The friendly name associated with each report suite ID can be changed by Adobe Customer Care. The friendly name can be seen inAnalytics in the site drop-down box in the top, left section of the screen.

1. s.dynamicAccountSelection

The dynamicAccountSelection variable lets you dynamically select the report suite based on the URL of each page.

dynamicAccountSelection does not work with custom link tracking.

Both dynamicAccountList and dynamicAccountMatch are ignored if the dynamicAccountSelection variable is not declared or set to 'false.'

## Syntax and Possible Values

1. s.dynamicAccountSelection=[true|false]

Only 'true' and 'false' are allowed as values of dynamicAccountSelection.

## Examples

1. s.dynamicAccountSelection=true
2. s.dynamicAccountSelection=false

3. s.dynamicAccountList

AppMeasurement for JavaScript can dynamically select a report suite to which it sends data. The dynamicAccountList variable contains the rules used to determine the destination report suite.

This variable is used in conjunction with the *dynamicAccountSelection* and *dynamicAccountMatch* variables. The rules in *dynamicAccountList*are applied if *dynamicAccountSelection* is set to 'true,' and they apply to the section of the URL specified in *dynamicAccountMatch*.

If none of the rules in *dynamicAccountList* matches the URL of the page, the report suite identified in *s\_account* is used. The rules listed in this variable are applied in a left-to-right order. If the page URL matches more than one rule, the left-most rule is used to determine the report suite. As a result, your more generic rules should be moved to the right of the list.

In the following examples, the page URL is http://www.mycompany.com/path1/?prod\_id=12345, *dynamicAccountSelection* is set to 'true,' and*s\_account* is set to "mysuitecom."

| DynamicAccountList Value | DynamicAccountMatch Value | Report Suite to Receive Data |
| --- | --- | --- |
| mysuite2=www2.mycompany.com;mysuite1=mycompany.com" | window.location.host | mysuite1 |
| "mysuite1=path4,path1;mysuite2=path2" | window.location.pathname | mysuite1, mysuite2 |
| "mysuite1=path5" | window.location.pathname | mysuitecom, mysuite1 |
| "myprodsuite=prod\_id" | window.location.search?window.location.search:"?") | myprodsuite |

The dynamicAccountList variable is a semicolon-separated list of name=value pairs (rules). Each piece of the list should contain the following items:

* one or more report suite ID (separated by commas)
* an equals sign
* one or more URL filters (comma-separated)

1. s.dynamicAccountList=rs1[,rs2]=domain1.com[,domain2.com/path][;...]

Only standard ASCII characters should be used in the string (no spaces).

## Examples

1. s.dynamicAccountList="mysuite2=www2.mycompany.com;mysuite1=mycompany.com"
2. s.dynamicAccountList="ms1,ms2=site1.com;ms1,ms3=site3.com"

# 4. s.dynamicAccountMatch

The dynamicAccountMatch variable uses the DOM object to retrieve the section of the URL to which all rules in dynamicAccountList are applied.

This variable is only valid when dynamicAccountSelection is set to 'True.' Since the default value is window.location.host, this variable is not required for Dynamic Account Selection to work.

The rules found in dynamicAccountList are applied to the value of dynamicAccountMatch. If dynamicAccountMatch only containswindow.location.host (default), the rules in dynamicAccountList apply only to the domain of the page.

## Syntax and Possible Values

The dynamicAccountMatch variable is usually populated by the Adobe consultant who provides the AppMeasurement for JavaScript file. However, the values listed below may be applied at any time.

1. s.dynamicAccountMatch=[DOM object]

| Description | Value |
| --- | --- |
| Domain (default) | window.location.host |
| Path | window.location.pathname |
| Query String | (window.location.search?window.location.search:"?") |
| Domain and Path | window.location.host+window.location.pathname |
| Path and Query String | window.location.pathname+(window.location.search?window.location.search:"?") |
| Full URL | window.location.href |

## Examples

1. s.dynamicAccountMatch=window.location.pathname
2. s.dynamicAccountMatch=window.location.host+window.location.pathname

# 5. s.dynamicVariablePrefix

The dynamicVariablePrefix variable allows deployment to flag variables, which should be populated dynamically.

Cookies, request headers, and image query string parameters are available to be populated dynamically.

Syntax and Possible Values

1. s.prop1="D=User-Agent”

OR USE CUSTOM FLAG FOR DYNAMIC VARIABLES

1. s.dynamicVariablePrefix=".."

Examples

1. s.prop1="D=User-Agent”

OR USE CUSTOM FLAG FOR DYNAMIC VARIABLES

1. s.dynamicVariablePrefix=".."
2. s.prop1="..User-Agent"

# 6. s.charSet

The charSet variable translates the character set of the Web page into UTF-8.

If the charSet variable contains an incorrect value, the data in all other variables are translated incorrectly. If JavaScript variables on your pages (e.g. pageName, prop1, or channel) contain only ASCII characters, charSet does not need to be defined. However, if the variables on your pages contain non-ASCII characters, the charSet variable must be populated.

The charSet variable is used to identify the character set of the page. For more information on character sets, see the [Multi-byte Character Sets](https://marketing.adobe.com/resources/help/en_US/whitepapers/multibyte/) white paper before using the charSet variable.

## Syntax and Possible Values

The charSet variable may only contain one of a predefined set of values, as listed in [Multi-byte Character Sets](https://marketing.adobe.com/resources/help/en_US/whitepapers/multibyte/).

1. s.charSet="character\_set"

## Examples

1. s.charSet="ISO-8859-1"
2. s.charSet="SJIS"

# **7. s.currencyCode**

The currencyCode variable determines the conversion rate to be applied to revenue.

All monetary amounts are stored in a currency of your choice. If that currency is the same as that specified in currencyCode, or currencyCodeis empty, no conversion is applied.

| **Max Size** | **Debugger Parameter** | **Reports Populated** | **Default Value** |
| --- | --- | --- | --- |
| N/A | cc | Conversion > Purchases > Revenue  All Conversion reports showing Revenue or monetary values | "USD" |

If your site allows visitors to purchase in multiple currencies, you should use the currencyCode variable to make sure revenue is stored in the appropriate currency. For example, if the base currency for your report suite is USD, and you sell an item for 40 Euros, you should populate the currencyCode with "EUR" on the HTML page. As soon as data collection receives the data, it uses the current conversion rate to convert the 40 Euros to its USD equivalent.

Populating the currencyCode variable on the HTML page instead of in the JavaScript file is recommend if you sell in multiple currencies. If you want to use your own conversion rates rather than the conversion rates used by Adobe, set the currencyCode to equal the base currency of your report suite. You then convert all revenue before sending it into Analytics.

Currency conversion applies to both revenue and any currency events. These are events that are used to sum values similar to revenue, such as tax and shipping. The revenue and currency events are specified in the products string. For more information on products, see [events](https://marketing.adobe.com/resources/help/en_US/sc/implement/events.html#concept_FFD115543D54401B98FE683BD7D5B3FE). For more details on how currencies are managed, see [Multi-Currency Support](https://marketing.adobe.com/resources/help/en_US/whitepapers/currency/)..

## Syntax and Possible Values

1. s.currencyCode="currency\_code"

Only the currency codes listed in [Multi-Currency Support](https://marketing.adobe.com/resources/help/en_US/whitepapers/currency/) are allowed.

## Examples

1. s.currencyCode="GBP"
2. s.currencyCode="EUR"

# **8. s.trackDownLoadLinks**

Set trackDownloadLinks to 'true' if you would like to track links to downloadable files on your site.

If trackDownloadLinks is 'true,' linkDownloadFileTypes is used to determine which links are downloadable files.

The trackDownloadLinks variable should only be set to 'false' if there are no links to downloadable files on your site, or you don't care to track the number of clicks on downloadable files. If trackDownloadLinks is 'true,' when a file download link is clicked, data is immediately sent to Analytics. The data that is sent with a download link includes the link download URL, and visitor click map data for that link. IftrackDownloadLinks is 'false,' then visitor click map data for links to downloadable files on your site is likely to be under reported.

## Syntax and Possible Values

The trackDownloadLinks variable is expected to be either 'true' or 'false.'

## Examples

1. s.trackDownloadLinks=true
2. s.trackDownloadLinks=false

# **9. s.trackExternalLinks**

If trackExternalLinks is 'true,' linkInternalFilters and linkExternalFilters are used to determine whether any link clicked is an exit link.

The trackExternalLinks variable should only be set to 'false' if there are no exit links on your site, or if you don't care to track the number of clicks on those exit links. An exit link is any link that takes a visitor off of your site. If trackExternalLinks is 'true,' then when you click an exit link, tracking data is immediately sent. The data that is sent with an exit link includes the link URL, link name, and visitor click map data for that link. If trackExternalLinks is 'false,' then visitor click map data for exit links on your site is likely to be under reported.

## Syntax and Possible Values

The trackExternalLinks variable is expected to be either 'true' or 'false.'

1. s.trackExternalLinks=true|false

## Examples

1. s.trackExternalLinks=true
2. s.trackExternalLinks=false

# **10. s.trackInlineStats**

The trackInlineStats variable determines whether ClickMap data is gathered.

If trackInlineStats is 'true,' data about the page and link clicked are stored in a cookie called s\_sq. If 'false,' s\_sq will have a value of "[[B]]," which is considered null.

## Syntax and Possible Values

1. s.trackInlineStats=true|false

The trackInlineStats variable is expected to be either 'true' or 'false.'

## Examples

1. s.trackInlineStats=true
2. s.trackInlineStats=false

# **11. s.linkDownloadFileTypes**

The linkDownloadFileTypes variable is a comma-separated list of file extensions.

If your site contains links to files with any of these extensions, the URLs of these links will appear in the File Downloads report.

The linkDownloadFileTypes variable is only relevant when trackDownloadLinks is set to 'True.'

Only left-mouse-clicks on a link are counted in the File Downloads report. All file downloads that start automatically when a page loads, or that are only downloaded after a redirect, are not counted in the File Downloads report. When you right-click a file and select the "Save Target As..." option, it is not counted in the File Downloads report.

The linkDownloadFileTypes variable may be used to track clicks to RSS feeds. If you have links to RSS feeds with a .xml or other extension, appending ",xml" to the linkDownloadFileTypes list allows you to see how often each RSS link is clicked.

## Syntax and Possible Values

Only include file extensions (no spaces).

1. s.linkDownloadFileTypes="type1[,type2[,type3[...]]]"

Any file extension may be included in the list. Be careful not to include a common file extension, such as htm or aspx, inlinkDownloadFileTypes. Doing so causes an extra image request to be sent for each click, which will be billed as a primary server call.

## Examples

1. s.linkDownloadFileTypes="exe,zip,wav,mp3,mov,mpg,avi,wmv,doc,pdf,xls"
2. s.linkDownloadFileTypes="exe,zip,wav,mp3,mov,mpg,avi,wmv,doc,pdf,xls,xml"

12. s.campaign :

The campaign variable identifies marketing campaigns used to bring visitors to your site. The value of campaign is usually taken from a query string parameter.

**Parameters**

| **Max Size** | **Debugger Parameter** | **Reports Populated** | **Default Value** |
| --- | --- | --- | --- |
| 255 Bytes | v0 | Conversion > Campaigns > Tracking Code | "" |

Every element in a marketing campaign should have an associated unique tracking code. For example, a paid search engine keyword may have a tracking code of 112233. When someone clicks the keyword with the 112233 tracking code and is routed to the corresponding website, the campaign variable records the tracking code.

There are two main ways to populate the campaign variable:

* The getQueryParam plug-in, used in the JavaScript file, retrieves a query string parameter from the URL. For more information on thegetQueryParam plugin, see [Implementation Plug-ins](https://marketing.adobe.com/resources/help/en_US/sc/implement/impl_plugins.html#concept_021F5E4A6BD745AE91E85E7138BE930F).
* Assign a value to the campaign variable in the HTML on the Web page.

With either method of populating the campaign variable, the Back button traffic may inflate the actual number of click-throughs from a campaign element.

For example, a visitor enters your site by clicking a paid search keyword. When the visitor arrives on the landing page, the URL contains a query string parameter identifying the tracking code for the keyword. The visitor then clicks a link to another page, but then immediately clicks the Back button to return to the landing page. When the visitor arrives a second time on the landing page, the URL with the query string parameter identifies the tracking code again. And a second click-through is registered, thereby falsely inflating the number of click-throughs.

To avoid this inflation of click-throughs, Adobe recommends using the getValOnce plugin to force each campaign click-through to be counted only once per session. For more information on the getValOnce plugin, see [Implementation Plug-ins](https://marketing.adobe.com/resources/help/en_US/sc/implement/impl_plugins.html#concept_021F5E4A6BD745AE91E85E7138BE930F).

## Syntax and Possible Values

1. s.campaign="112233"

The campaign variable has the same limitations as all other variables. Adobe recommends limiting the value to standard ASCII characters.

## Case Sensitivity

eVars are case insensitive, but they are displayed in the capitalization of the first occurrence. For example, if the first instance of eVar1 is set to "Logged In," but all subsequent instances are passed as "logged in," reports always show "Logged In" as the value of the eVar.

## Examples

1. s.campaign="112233"
2. s.campaign=s.getQueryParam('cid');

Prop

* + Expire on hit (a hit is a row in a data feed). Cannot persist further than the hit. This means that if you want to attribute an event to a prop you need to do it in the same analytics server call
  + Only has linear allocation. So all metrics (events in data feed) that you apply to a prop reports will use linear allocation
  + Pathing can be enable on any props. This means that you can see the sequential order of the values that have been sent. For best result, the prop needs to be specified in all analytics server calls. If a server call is missing the value then it will count as an exit from the path.
  + You can also use a prop as a list prop to send multiple values but I prefer to use list variables as you can assign different allocations
* Evar
  + The expiration can be set to what you want: hit, visit, never, after x days, when a specific event is sent etc....
  + The allocation can be set to:
    - Most recent (last touch): the most recent values sent to this eVar will persist (see expiration), until expiration end or new value is set. When an eVar is set (via server call) it will be in its corresponding evar column in data feed. And the persisting value will be set in post\_evar. If using processing rules then only post\_evar will be set. To see instances of the evar chec post\_event\_list and check for corresponding numeric id of the evar.
    - First touch: the first value send via sever call will persist until expiration. Even if you send new value it will not overwrite first value sent. Reset at expiration
    - Linear: all values send via server calls will get credit for the event when sent.