Adobe Consulting Plugin

getAndPersistValue

v2.0

Contents

Adobe Consulting Plugin – getAndPersistValue

General Details

Plugin Name: getAndPersistValue
Latest Version: v2.0
Latest Version Save Date: 2018-04-16

Purpose of This Plugin

What does this plugin do?

The getAndPersistValue plugin allows you to store a value in a cookie that can be retrieved later during a visit.

Why should I use this plugin?

You should use the getAndPersistValue plugin if you want to persistently and automatically set an Analytics variable equal to the same value in every server call after the variable was initially set.

Why shouldn't I use this plugin?

From a conversion perspective, the built-in persistence of eVars eliminates the need to set them equal to the same value in every server call after they were initially set.   In general, if you have no need to automatically set an Analytics variable equal to the same value in every server call after the variable was initially set, then you have no need for the getAndPersistValue plugin.

Prerequisites

You must have AppMeasurement (i.e. the base Adobe Analytics Code) to run the getAndPersistValue plugin

How to Implement

- Copy + Paste the code (see below) to anywhere within the Plugins section of the AppMeasurement file

NOTE: Adding the cyan-highlighted comments/version numbers of the code to the AppMeasurement file will help Adobe Consulting with troubleshooting any potential implementation issues.

- Run the getAndPersistValue function as needed within the doPlugins function (see example calls below)

Arguments to Pass In

vtp: (required) The value that is meant to persist from page to page after it has been written to the cookie specified in the cn argument

cn: (optional) The name of the cookie that will store the value to persist (as set within the vtp argument). If the cn argument is not set, then it will default to the name of "s_gapv"

ex: (optional) The number of days that the value is set to persist before it expires. If the argument is set equal to 0 (i.e. zero, without quotes) or is not set at all, the value will stop persisting at the end of the visit (i.e. after 30 minutes of inactivity)

Returns

If the variable/value to be passed in via the vtp argument has been set on the current page before the getAndPersistValue plugin is called, then the plugin will first set the cookie specified in the cn argument equal to the value of the vtp argument and then return the (new) value of the cookie.

The plugin will only return the value of the cookie if the variable in the vtp argument hasn't been set on the page.

Cookies

The getAndPersistValue sets a cookie with a name that you specify via the cn argument. The length of the cookie depends on the value that is persisting (i.e. the vtp argument's value). The cookie will expire depending on the value of the ex argument

Example Calls

Example #1

The following code will set s.eVar21 equal to the value of "hello". The code will then set the ev21gapv cookie, which will expire in 28 days, equal to the value of s.eVar21 (i.e. "hello"). The code will then (re)set s.eVar21 equal to the value of the ev21gapv cookie.

```
s.eVar21 = "hello";
s.eVar21 = s.getAndPersistValue(s.eVar21,"ev21gapv",28);
```

Example #2

Assume that s.eVar21 has **not** been set on the current page yet but was set equal to "hello" on a previous page within the last 28 days. The following code will only set s.eVar21 equal to the value of the ev21gapv cookie (i.e. "hello"). It does not reset the ev21gapv cookie since s.eVar21 was not set on the current page before the function was called.

```
s.eVar21 = s.getAndPersistValue(s.eVar21,"ev21gapv",28);
```

Example #3

Assume that s.eVar21 has not been set on the current page yet but was set equal to "hello" on a previous page within the last 28 days. The following code will set only **s.prop35** equal to the value of the ev21gapv cookie (i.e. "hello"). It will not set eVar21.

```
s.prop35 = s.getAndPersistValue(s.eVar21,"ev21gapv",28);
```

Example #4

The following code will set s.eVar21 equal to the value of "howdy". The code will then set (or reset) the ev21gapv cookie, which will expire in 28 days, equal to the value of s.eVar21 (i.e. "howdy"). The code will then set s.**prop35** equal to the value of the ev21gapv cookie (i.e. "howdy").

```
s.eVar21 = "howdy";
s.prop35 = s.getAndPersistValue(s.eVar21,"ev21gapv",28);
```

Example #5

Assume that s.eVar21 has not been set on any pages within the last 28 days. The following code will set s.eVar21 equal to nothing since the ev21gapv cookie would have expired 28 days after it was last set.

```
s.eVar21 = s.getAndPersistValue(s.eVar21,"ev21gapv",28);
```

Example #6

The following code will set eVar30 equal to "shopping". It will then set the s_gapv cookie, which will expire at the end of the browser session, equal to the value of s.eVar30 (i.e. "shopping"). It will then set s.eVar30 equal to the value of the s_gapv cookie (i.e. the getAndPersistValue call returns the value of the s_gapv cookie, which in this case is "shopping").

```
s.eVar30 = "shopping";
s.eVar30 = s.getAndPersistValue(s.eVar30);
```

If s.eVar30 is not set to an explicit value on any additional pages seen during the session but **is** set (in doPlugins) via the following code…

```
s.eVar30 = s.getAndPersistValue(s.eVar30);
```

…s.eVar30 will be set equal to "shopping" (i.e. the persisted value of the s_gapv cookie)

S Object Replacement

When instantiating the main AppMeasurement library object with a name other than "s", change the following portion of the plugin code from this…

```
s.getAndPersistValue=function(vtp, cn, ex)
```

…to this:

```
[objectname].getAndPersistValue=function(vtp, cn, ex)
```

Version History

2.0 (2018-04-16)

- Point Release (smaller code size)
- Passing 0 into the "ex" argument now forces expiration after 30 minutes of inactivity rather than expiration at the end of the "session".

1.0 (2016-01-18)

- First official 1.0 version (previous versions/changes undocumented)

Code to Deploy

```
/******************************************* BEGIN CODE TO DEPLOY *******************************************/
/* Adobe Consulting Plugin: getAndPersistValue v2.0 (Requires AppMeasurement) */
s.getAndPersistValue=function(vtp,cn,ex){var b=new Date;cn=cn?cn:"s_gapv";(ex=ex?ex:0)?b.setTime(b.getTime()+864E5*ex):
b.setTime(b.getTime()+18E5);vtp||(vtp=this.c_r(cn));this.c_w(cn,vtp,b);return vtp};
/******************************************** END CODE TO DEPLOY ********************************************/
```