

DAP Code v6.8 Quick Guide

Prepared for:



Introduction

This Quick Guide accompanies the Digital Analytics Program (DAP) Universal Analytics (UA) / GA4 code. Its intent is to allow agencies participating in the GSA DAP program to quickly understand, test and deploy the latest JavaScript file to implement DAP UA/GA4 Code.

Note that as of version 3.0, agencies have the ability to implement the GSA DAP code from within Google Tag Manager (GTM). For more information on the Google Tag Manager implementation, see the “Google Tag Manager Quick Guide”.

If you are new to DAP, please read “[DAP Code Capabilities Summary & Reference](#)”.

For agencies updating to the latest version of the DAP code, please read the latest “[Release Notes](#)”.

DAP Code Deployment Steps

Agencies can choose one of the deployment options for the DAP code: 1) use DAP’s centrally hosted URL (recommended), or 2) host the code locally... The DAP team strongly recommends that agencies use the [DAP centrally hosted URL](#) to allow the DAP team seamlessly push the latest versions of the DAP code to agencies. The DAP code gets frequent updates with new features, fixes, and enhancements, especially with the transition to GA4. ,.

DAP Centrally Hosted URL Implementation Instructions:

- 1) Create the Federated Analytics script block using <https://dap.digitalgov.gov/Universal-Federated-Analytics-Min.js> as shown in the implementation examples below.
- 2) Insert the Federated Analytics script block in the <head> of all pages on your site. The best location for the script block is directly before the closing </head> tag. The script block must be inserted in a way that it appears on every page across the entire site, including sub-domains (e.g.: ed.gov and www.ed.gov and www2.ed.gov,

DAP Code v6.8 Quick Guide

etc.) and on-site search sub-domains (e.g.: find.ed.gov, searchjustice.usdoj.gov, search.nasa.gov).

- 3) Please note that the script tag will not work correctly unless the src is specified with https://. The DAP code central host URL has been configured to only be called securely, and will not function unless being called over SSL.

Local Host Implementation Instructions:

- 1) Download these **three** files and save them locally:
 - [Federated.js.map](#)
 - [Universal-Federated-Analytics-min.js](#)
 - [Universal-Federated-Analytics.js](#)
- 2) Copy the files to the same location as your agency site's other JavaScript files (typically a folder called /scripts or /javascript, etc.). The three files should be on the same server as the site's other files, and all share the same location.
- 3) Insert the Federated Analytics Script Block in the <head> of all pages on your site. The best location for the script block is directly before the closing </head> tag. The script block must be inserted in a way that it appears on every page across the entire site, including sub-domains (e.g.: ed.gov and www.ed.gov and www2.ed.gov, etc.) and on-site search sub-domains (e.g.: find.ed.gov, searchjustice.usdoj.gov, search.nasa.gov)

Note: To load the DAP code asynchronously add the “async” attribute to the Federated Script block, like this:

```
<script async type="text/javascript" id="_fed_an_ua_tag"
src="https://dap.digitalgov.gov/Universal-Federated-Analytics-
Min.js?agency=HHS"></script>
```

Google Tag Manager

DAP 3.0 supports implementing the DAP code from within Google Tag Manager. Follow the simple instructions below (and in the rest of this document) in order to use Google Tag Manager to load the DAP code.

GTM Account Creation/Insertion

If you do not already have a Google Tag Manager container/account, follow these instructions.

1. Go to tagmanager.google.com to create a Google Tag Manager account (or to access an existing account).
2. Create a container for your agency site
3. Add the container snippet to your site after the opening <body> tag.
4. Remove any existing legacy Google Analytics code.

See [this link from Google](#) on creating a new Google Tag Manager container if you need further assistance.

Here is an example of what your Google Tag Manager code may look like. **Do not use this example; use the code that you receive from the Google Tag Manager website for your account.**

```
<!-- Google Tag Manager --> <noscript><iframe
src="//www.googletagmanager.com/ns.html?id=GTM-XYZ123"
height="0" width="0"
style="display:none;visibility:hidden"></iframe></noscript>
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
  new Date().getTime(),event:'gtm.js'});var
f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;
j.src=
'//www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.inser
tBefore(j,f); })(window,document,'script','dataLayer','GTM-
XYZ123');</script> <!-- End Google Tag Manager -->
```

DAP Code Placement in GTM

These steps outline the placement of DAP code within a Google Tag Manager container.

1. Click **New Tag** from the Container screen.
2. Select **“Custom HTML Tag”** as the tag type.
3. Give the tag a name, like **“DAP 6.8 Google Analytics Code”** so that you can easily find it later if needed.
4. Copy and paste the DAP code snippet into the textbox. Note that you may want to make some customizations based on the “DAP Script Implementation Examples/Scenarios” shown below, as well as the “DAP Code Capabilities Summary & Reference” guide.
5. Choose fire on **“All Pages”** as the triggering rule.
6. Click **“Create Tag”** to save.

Note that at this stage, the DAP code will not start to send data to Google Analytics until you **publish** your GTM container.

Here is an example of what your GTM tag may look like:

DAP Code v6.8 Quick Guide

Tag Configuration

Tag Type

< >

Custom HTML

Custom HTML Tag


HTML ?

1

<script async type="text/javascript" id="_fed_an_ua_tag" src="https://dap.digitalgov.gov/Universal-Federated-Analytics-Min.js?agency=HHS"></script>

Triggering

Firing Triggers



All Pages

Page View

DAP Script Implementation Examples/Scenarios

Agencies must use the basic DAP script block implementation at a minimum and can add additional customization as part of the script block as needed. Several examples of customizations are provided in this section (Scenarios A and B) to demonstrate the use of other custom parameters.

DAP Code v6.8 Quick Guide

See the Query String Parameters & Default Values Reference at the bottom of this Quick Guide for the complete list of allowed customization parameters. To get a more in-depth understanding of how these customizations relate to available DAP code features, read the “DAP Code Capabilities Summary & Reference” guide.

Default Scenario: Basic DAP script block implementation (minimum requirement).

- All sites’ script blocks will have these attributes: id="_fed_an_ua_tag" language="text/javascript"
- All script blocks require an src, which will be either the central host URL (shown below) or the local file path, to be specified.
- All script blocks must specify their agency’s standard abbreviation as shown below.

```
<script async type="text/javascript" id="_fed_an_ua_tag"
src="https://dap.digitalgov.gov/Universal-Federated-Analytic
s-Min.js?agency=HHS"></script>
```

If the website is also part of a sub-agency, the sub-agency value should be specified as follows:

```
<script async type="text/javascript" id="_fed_an_ua_tag"
src="https://dap.digitalgov.gov/Universal-Federated-Analytics-
Min.js?agency=HHS&subagency=FEMA"></script>
```

Scenario A (Optional): The agency is implementing the PUA tracker, PGA4 tracker, and a visitor cookie expiration of 6 months.

```
<script async type="text/javascript" id="_fed_an_ua_tag"
src="https://dap.digitalgov.gov/Universal-Federated-
Analytics-Min.js?agency=DHS&pua=UA-000000-0&pga4=G-111111&c
to=6"></script>
```

Scenario B (Optional): The agency is implementing additional Custom Search Parameters (sp), Download Extensions (exts), and Demographics.

Example 1: Adding a single custom search parameter and a single custom download extension.

```
<script async type="text/javascript" id="_fed_an_ua_tag"
src="https://dap.digitalgov.gov/Universal-Federated-Analytics-
Min.js?agency=DHS&sp=lookup&exts=mp3"></script>
```

Example 2: Adding multiple custom search parameters and multiple download extensions.

```
<script async type="text/javascript" id="_fed_an_ua_tag"
src="https://dap.digitalgov.gov/Universal-Federated-Analytics-
Min.js?agency=DHS&sp=lookup,locate&exts=tif,mp3"></ script>
```

Scenario C (Optional): The agency is implementing the PUA tracker, PGA4 tracker, site topic, and site platform.

```
<script async type="text/javascript" id="_fed_an_ua_tag"
src="https://dap.digitalgov.gov/Universal-Federated-Analytics-
Min.js?agency=DHS&pua=UA-000000-
0&pga4=G-111111&sitetopic=Health&siteplatform=Federalist"></sc
ript>
```

Note

1. The code is **not** case sensitive, e.g. Agency, agenCY, agency or agencyY will all work properly.
2. The order of the querystring parameters does not matter, e.g. the sp parameter can come before exts and agency.

Note to Developers: Dynamically Added Links

Please note that this section is intended only for website administrators who have the knowledge that their links are added dynamically and are not part of the page load. The majority of websites participating in DAP are not affected by this scenario.

Links that are added dynamically to an agency website may be added to the site after the DAP tracking code has completely loaded. As a result, these dynamically added links may not be included in the auto-tracking functionality. The solution below outlines a way to re-trigger the auto-tracking functionality once the dynamically loaded links have been completely added.

How to Re-trigger the Auto Tracking Functionality

The **_initAutoTracker()** function is what the Digital Analytics Program uses to generate automatic link event tracking. This function should be called once all links have been added to the page. Note that the below code is an example, to demonstrate a technical concept. Actual execution on agency sites may vary.

In the below example, we assume the agency has an existing function, **addDynamicLinks()**, that normally adds links to the agency site.

```
<script>

function addDynamicLinks() {

    console.log('Here is where links would be added!');

    _initAutoTracker();

}

addDynamicLinks();

</script>
```

Dynamically Inject DAP Library

You can use this alternate format of the DAP script tag when you want to dynamically build and inject the JavaScript library. A good example of this case might be if you want to include the YouTube tracking parameter (yt=true) on pages that match a specific condition. Here is an example:

```
<script
type="text/javascript">
(function() {
    var s =
    document.createElement('script'); var
    params = '?'+'agency=DHS&yt=true'; s.type =
    'text/javascript';
    s.id =
    '_fed_an_ua_tag';
    s.async = true;
    s.src =
    'Universal-Federated-Analytics-Min.js'+params; var x =
    document.getElementsByTagName('script')[0];
    x.parentNode.insertBefore(s, x);
})();
</script>
```

Query String Parameters & Default Values Reference

Query String Parameter	Data Type	Default Value	Description	Default Behavior
agency	String	unspecified:domain.com	Agency Custom dimension value	Agency CD show up as "unspecified:domain.com" in the reporting interface
subagency	String	unspecified:domain.com domain.com	Sub Agency custom dimension value	Sub-agency CD show up as "unspecified:domain.com – domain.com" in the reporting interface
sitetopic	String	unspecified:domain.com	Site Topic custom dimension value	"unspecified:domain.com" in the reporting interface
siteplatform	String	unspecified:domain.com	Site Platform custom dimension value	"unspecified:domain.com" in the reporting interface
sp	Comma separated string		Additional search parameter	By default, these search parameters are automatically tracked: q, querytext, nasalInclude, k, and qt.
exts	Comma separated string		Additional download extensions	By default, these file extensions will be included as downloads "doc docx xls xlsx xlsml ppt pptx exe zip pdf js txt csv dxf dwg dwt rtf dwfx dwg wmv jpg msi 7z gz tgz wma mov avi mp3 mp4 csv mobi epub swf rar"
yt	boolean	false	Enable/Disable YouTube Tracker	YouTube videos will not be tracked.

DAP Code v6.8 Quick Guide

sdor	String	auto	<p>If “auto” cookie will be set as subdomain.domain.com</p> <p>If set to custom value <i.e. sub.domain.com or site.com>, all sub-domains of the current domain will be linked and links between these portals will be considered internal links.</p>	Sub-domains on the same domain will be treated as separate sites.
pua	Tracking ID		Parallel UA Trackers	No additional properties are tracked.
pga4	Measurement ID		Parallel GA4 Trackers	No additional properties are tracked.
autotracker	boolean	true	Enable/Disable AutoTracker	Downloads and outbound links are automatically tracked.
parallelcd	boolean	false	Send Custom Dimensions – Parallel Tracker (true/false)	The following 3 Custom Dimensions will be sent to the Parallel Tracker account.
palagencydim	number (custom dimension slot number)	1	Agency Custom Dimension Slot # (Parallel Tracker)	1
palsubagencydim	number (custom dimension slot number)	2	Sub Agency Custom Dimension Slot # (Parallel Tracker)	2
palversiondim	number (custom dimension slot number)	3	Code Version Custom (Parallel Tracker) Dimension Slot #	3
paltopicdim	number (custom dimension slot number)	4	Site Topic Custom (Parallel Tracker) Dimension Slot #	4
palplatformdim	number (custom dimension slot number)	5	Site Platform Custom (Parallel Tracker) Dimension Slot #	5

DAP Code v6.8 Quick Guide

palscriptsrcdim	number (custom dimension slot number)	6	Script Source Slot # (Parallel Tracker)	6
palurlprotocoldim	number (custom dimension slot number)	7	URL Protocol Slot # (Parallel Tracker)	7
palinteractiontypedim	number (custom dimension slot number)	8	Interaction Type Slot # (Parallel Tracker)	8
cto	Integer (of months)	24	Ga cookie expiration in months	GA cookie expires from user's system after 2 of not being updated. Note: For Chrome it shouldn't be more than 400 days ~ 13 months.