



Google™ Analytics

# Installing and Customizing the Google Analytics Tracking Code

Beta Version

# Contents

What is the tracking code?	3
Basic tracking code	3
Tracking virtual page views	4
Tracking downloaded files	5
Tracking a page in multiple accounts	6
Tracking subdomains	7
Track a visitor across domains using a link	8
Track a visitor across domains using a form	9
E-commerce transactions	10
Adding organic sources	11
Segmenting visitor types	12
Restrict cookie data to a subdirectory	13
Control data collection settings	14
Control session timeout	15
Control campaign conversion timeout	16
Custom campaign fields	17
Using the anchor (#) with campaign data	18
Setting keyword ignore preferences	19
Control the data sampling rate	20
Using Google Analytics and Urchin	21

## What is the tracking code?

The Google Analytics tracking code is a JavaScript code snippet required to be part of the source code of the web pages you would like to track with Google Analytics. To track an entire website, simply add the tracking code to all the web pages on your site.

To sign up for a Google Analytics account, please visit <http://www.google.com/analytics>.

If you are an AdWords advertiser, click on the 'Analytics' tab in your AdWords account to sign up for Google Analytics .

## Basic tracking code

*Start collecting your data effortlessly with Google Analytics.*

Easily copy and paste our basic tracking code into the source code of your site. This JavaScript snippet is the minimum code required for a site to be tracked with Google Analytics.

### basic tracking code

Automatically detects protocol (http:// or https://)

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

For the most accurate results, add the basic tracking code to every page on your website that you want to track.

To minimize page loading time for users with slow connection speeds, we recommend that you place the tracking code just before the closing </body> tag of the page.

## Tracking virtual page views

*Google Analytics allows you to change the actual URI that appears in your reports.*

If your URLs contain query terms or other parameters, you may want to see a more readable URL in the reports. Create an arbitrary entry in Google Analytics reports by sending a string beginning with a forward slash to the **\_trackPageview()** function. You can find data for the pages on your website in the 'Content' section of the reports.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'></script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._initData();
  pageTracker._trackPageview("/my/virtual/url");
</script>
```

This custom URI will appear in your reports.

## Tracking downloaded files

*Track downloaded files and get valuable metrics for them*

Know when key files are downloaded from your site. To create an entry in Google Analytics reports for file downloads, send a string beginning with a forward slash to the **\_trackPageview()** function.

### original link format

```
<a href="mydoc.pdf" >Download a PDF</a>
```

### tracked link format

```
<a href="mydoc.pdf"  
onclick="pageTracker._trackPageview('/mydoc.pdf');">Download a PDF</a>
```

Since this URI string matches the HREF filename, you will be able to see data for this download in the Site Overlay report, too.

## Tracking a page in multiple accounts

*Need to track one page in multiple accounts? Now you can!*

A single page may be significant to several accounts. If you want to track a page in multiple Google Analytics accounts, simply instantiate more than one tracker object.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>
<script type="text/javascript">
  var firstTracker = _gat._getTracker("UA-12345-1");
  firstTracker._initData();
  firstTracker._trackPageview();

  var secondTracker = _gat.getTracker("UA-67890-1");
  secondTracker._initData();
  secondTracker._trackPageview();
</script>
```

First Instance:  
Data sent to account  
number 12345-1

Second Instance:  
Data sent to account  
number 67890-1

NOTE: The tracker objects can be assigned to a variable or variables of your choice. Here, we called them **firstTracker** and **secondTracker** to make it clear, but the naming of these objects is really up to you.

## Tracking subdomains

*Track your web presence without missing a single subdomain.*

If your site spans multiple subdomains, you will need to modify your tracking code to track visitors across these subdomains.

If your site spans multiple subdomains, set your domain name to the root domain using **`_setDomainName()`**. If you are using different root domains, use **`_setDomainName("none")`**.

your domain names

```
www.example.com  
shop.example.com
```

### tracking code - customized for tracking subdomains

```
<script type="text/javascript">  
  var gaJsHost = (("https:" == document.location.protocol)  
  ? "https://ssl." : "http://www.");  
  document.write("\<script src='" + gaJsHost  
  + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );  
</script>  
  
<script type="text/javascript">  
  var pageTracker = _gat._getTracker("UA-12345-1");  
  pageTracker._setDomainName("example.com");  
  pageTracker._initData();  
  pageTracker._trackPageview();  
</script>
```

Notice that the subdomains are omitted, and only the root domain is used.

## Track a visitor across domains using a link

*Comprehensively track a visitor across multiple websites!*

If your visitors navigate between different root domains, you can still track their navigation. This requires maintaining a session by transferring cookies from one domain to another.

To send cookies via URL parameters (HTTP GET), use the **`_link()`** function. This function attaches the tracking cookies to the end of the URL.

### tracking code - customized for tracking a third-party site

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>
<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setDomainName("none");
  pageTracker._setAllowLinker(true);
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

Add these two lines to the basic tracking code.

### modified link to third-party site

```
<a href="http://newsite.com/test.html"
onclick="pageTracker._link('http://newsite.com/test.html'); return
false;">click me</a>
```

Change the links to the third-party website to include the **`_link()`** function.

## Track a visitor across domains using a form

*Does your form send visitors to another domain?  
Track them as one!*

This is the second method you can use to track multiple websites. If your site hosts a form that sends visitors to another domain name, you can send cookies via HTTP POST by using the **\_linkByPost()** function.

### tracking code - customized for tracking a third-party site

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'\>\</script>" );
</script>
<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setDomainName("none");
  pageTracker._setAllowLinker(true);
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

Add these two lines to the basic tracking code.

### modified form

```
<form name="f" method="post" onsubmit="pageTracker._linkByPost(this)">
  ...
</form>
```

Change the onSubmit event in your form to include the **\_linkByPost()** function.

## E-commerce transactions

*Track sales revenue and measure success when you sell!*

Simply use the code below on your receipt page.

```

<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
  ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
  + "google-analytics.com/ga.js' type='text/javascript'\>\</script>" );
</script>
<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._initData();
  pageTracker._trackPageview();
  pageTracker._addTrans (
    "1234",           // order ID - required
    "Mountain View", // affiliation or store name
    "11.99",         // total - required
    "1.29",          // tax
    "5",             // shipping
    "San Jose",      // city
    "California",    // state or province
    "USA"            // country
  );
  pageTracker._addItem(
    "1234",           // order ID - required
    "DD44",           // SKU/code
    "T-Shirt",        // product name
    "Green Medium",  // category or variation
    "11.99",          // unit price - required
    "1"              // quantity - required
  );
  pageTracker._trackTrans ();
</script>

```

Add one transaction.

Use the `_addItem()` function once for each unique item or SKU in the transaction.

Send the data to us.

**NOTE:** The transaction and item values specified in the above code need to be populated with actual values by your shopping cart program. Be sure not to use any currency symbols (such as \$ or £) or commas ( , ) in the monetary fields.

## Adding organic sources

*Track all search engines and optimally identify your organic traffic.*

By default, Google Analytics recognizes many of the Internet's most common search engines, but you can easily customize this list by adding search engines with the **`_addOrganic()`** function. New items are added to the end of the list.

### another search engine

```
www.bigsearch.com?q=queryterm
```

### tracking code - recognizes another search engine

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._addOrganic("bigsearch", "q");
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

Call this function once for each additional search engine you want to track.

Google Analytics currently identifies referrals from the following search engines in your reports:

Google, Yahoo, MSN, AOL, Lycos, Ask, Altavista, Netscape, CNN, Looksmart, About, Mamma, Alltheweb, Gigablast, Voila, Virgilio, Live, Baidu, Alice, Yandex, Najdi, Club-internet, Mama, Seznam, Search, Wp, Onet, Netsprint, Google.interia, Szukacz, Yam, Pchome

Google Analytics adds new recognized search engines to this list regularly, so check with the Help Center for the most current list.

## Segmenting visitor types

*Segmentation is powerful! Quickly segment types of visitors.*

Set a visitor segmentation cookie to help you classify the types of visitors coming to your site. For example: *New Customer vs. Prospect* or *Cat Owner vs. Dog Owner*.

This custom segmentation option is available in addition to the pre-defined segmentation options like language, city, etc.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._initData();
  pageTracker._trackPageview();
  pageTracker._setVar("test_value");
</script>
```

NOTE: The value passed to the **\_setVar()** function will appear in the *User Defined* report in the *Visitors* area of Google Analytics reports. You can also cross-segment other reports by this "User Defined Value".

## Restrict cookie data to a subdirectory

*Control where your Google Analytics first-party cookies are set to (in case you only own a subdirectory on your domain).*

By default, Google Analytics sets the cookie path to /. If you would like to change this, simply send your preferred cookie path to the **\_setCookiePath()** function.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'></script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setCookiePath("/path/of/cookie/");
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

## Control data collection settings

*You can control the amount of information you collect.*

By default Google Analytics will track an optimal set of data elements, such as Flash versions and web browser information. You have the option to turn this detection off by using the functions listed below, but be careful, because once you do so, you will lose the data permanently.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setClientInfo(false); // track browser info
  pageTracker._setAllowHash(false); // cookie integrity checking
  pageTracker._setDetectFlash(false); // detect Flash version
  pageTracker._setDetectTitle(false); // track title in reports
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

## Control session timeout

*Don't like the standard session timeout of 30 minutes?  
You can change it to what works for your business.*

By default, Google Analytics considers a session timeout to be 30 minutes, or 1800 seconds. You have the option to modify the number of seconds by using the **`_setSessionTimeout()`** function. Please use this function with caution because this very important setting is used to compute visits.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setSessionTimeout("3600"); // sets timeout in seconds
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

## Control campaign conversion timeout

*Google Analytics credits a campaign if conversion happens within six months, but you can adapt the timeout to your unique business needs.*

By default, Google Analytics credits a campaign if a conversion occurs within six months, or 15768000 seconds. Change the campaign conversion timeout by sending the desired number of seconds to the **`_setCookieTimeout()`** function.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setCookieTimeout("31536000"); // sets timeout in seconds
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

## Custom campaign fields

*Easily configure Google Analytics to recognize campaigns with custom field names in manually tagged URLs.*

When manually tagging your campaigns, you can use either the default Google Analytics campaign field names or custom names. The following are the default campaign field names:

- utm\_source
- utm\_medium
- utm\_term
- utm\_content
- utm\_campaign

Track campaigns with custom field names by passing your desired names to these campaign functions on your landing pages, enabling Google Analytics to recognize the campaign information in your manually tagged URLs.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>
<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setCampNameKey("ga_campaign"); // name
  pageTracker._setCampMediumKey("ga_medium"); // medium
  pageTracker._setCampSourceKey("ga_source"); // source
  pageTracker._setCampTermKey("ga_term"); // term/keyword
  pageTracker._setCampContentKey("ga_content"); // content
  pageTracker._setCampIdKey("ga_id"); // id number
  pageTracker._setCampNOKey("ga_nooverride"); // don't override
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

NOTE: Your tagged links must contain variables that map to Medium, Source, and Campaign Name.

Please see the Google Analytics Help Center for more information about campaign tracking.

## Using the anchor (#) with campaign data in URL

*You can adapt Google Analytics to accept manually tagged URLs that use the # symbol instead of the default question mark (?).*

Simply set the **\_setAllowAnchor()** to true. Please confirm with your webmaster that using the # instead of the ? does not cause issues with other tools on your site.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'></script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setAllowAnchor(true);
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

## Setting keyword ignore preferences

You can configure Google Analytics to treat certain keywords as direct traffic, such as visitors who type in your domain ([www.mb4r.com](http://www.mb4r.com)) into a search engine.

Use **`_addIgnoredOrganic()`** to treat a keyword as a referral or **`_addIgnoredRef()`** to treat a referral as direct.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'></script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._addIgnoredOrganic("ignore"); // treat keyword as referral
  pageTracker._addIgnoredRef("urchin.com"); // treat referral as direct
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

## Control the data sampling rate

*Traffic too high? Need faster long term data trends?  
Simplify by specifying a sample size that's more  
manageable to analyze.*

By default, Google Analytics tracks every visitor. If you would like to implement a sampling rate instead, you can send your preferred rate to the **`_setSampleRate()`** function.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setSampleRate(50); // sets sampling rate at 50%
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

## Using Google Analytics and Urchin

*Google Analytics and Urchin software are completely compatible!*

Enable tracking of your site with Google Analytics and Urchin software by calling **\_setLocalRemoteServerMode()** in the tracking code. Simple.

```
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol)
    ? "https://ssl." : "http://www.");
  document.write("\<script src='" + gaJsHost
    + "google-analytics.com/ga.js' type='text/javascript'\>\</script>" );
</script>

<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-12345-1");
  pageTracker._setLocalRemoteServerMode() ;
  pageTracker._initData();
  pageTracker._trackPageview();
</script>
```

Using the **\_setLocalRemoteServerMode()** function tells Google Analytics to request the tracking image (**\_\_utm.gif**) from both your own server and Google's server.

NOTE: Remember to remove **utm.js** from your webpages if you are using this feature.