



**CARDINAL**PATH

# R.U.G.A.

**(REALLY UNDERSTANDING GOOGLE ANALYTICS)**

By Kent Clark

For Cardinal Path

<http://www.cardinalpath.com/blog>

<i>Introductions and the __utm.gif</i> .....	3
<i>What is a Pageview?</i> .....	6
<i>Event Basics</i> .....	8
<i>User Defined Variables, Cookies, and the utm.gif</i> .....	10
<i>Really Understanding eCommerce</i> .....	12
<i>Visits and Visitors</i> .....	15
<i>Direct Visitors</i> .....	17
<i>Really Understanding Referrals</i> .....	19
<i>Understanding Campaigns</i> .....	21
<i>_tracksocial</i> .....	24
<i>Extras: Technical Details</i> .....	26

## Introductions and the \_\_utm.gif

Really Understanding Google Analytics is often like translating a language. Visit, pageview, transaction, keyword, these are all terms with clear meaning in English, and a very clear meaning in Google Analytics. The problem is that they don't have the same meaning and the result is that reports end up with numbers that, from an "English language" perspective, just don't make sense.

Think of it as translating a phrase into Chinese (Mandarin, Cantonese, whichever), then back to English.

### Google translate



From: English To: Chinese Translate

Web analytics is confusing

Listen

English to Chinese (Traditional) translation

網絡分析是混亂

Listen Read phonetically

New! Click the words above to view alternate translations. Dismiss

### Google translate



From: Chinese To: English Translate

網絡分析是混亂

Listen Read phonetically

Chinese to English translation

Network analysis is the confusion

Listen

New! Click the words above to view alternate translations. Dismiss

Google Analytics isn't that different. We have terms and phrases like "visit" "pageview" "time on site", which have one meaning in English which we translate into an action online, interpret, and then have fed back to us by Google Analytics with an English language label. And like English to Chinese to English, the results aren't always what they seem to be...

In this series of posts, we're going to dive into the language of Google Analytics, and look at how each function is defined and determined. From the basics of the utm.gif to tracking more complex actions, to give you an idea of what your reports really mean.

## Introducing the \_\_utm.gif

Quick refresher: How does Google Analytics record your data?

You enter a site and execute a javascript which writes data about your actions into first party cookies.



- Said javascript then writes data calculated with a combination of cookie values and other data. This is then written onto a query string at the end of a `__utm.gif` request.
- Since the `__utm.gif` is hosted on Google's servers, Google reads the request and logs the information so that it can process them into reports.

You can look at what query request you're making with some simple tools. Download a plugin such as Live HTTP Headers and try loading a site such as "cardinalpath.com". Or, alternatively, if you're using Chrome, go into developer tools and check "resources". In either, you can see the `__utm.gif` URL (some times there may be two, such as in cases where the page is using GA and Urchin) and it should look something like this (but in one long string):

```
http://www.google-analytics.com/__utm.gif?utmwv=4.8.9
&utmhn=www.cardinalpath.com
&utmcs=UTF-8
&utmsr=1680x1050
&utmssc=32-bit
&utmul=en-us
&utmje=1
&utmfl=10.2 r154
&utmdt=Cardinal Path - Web Analytics and Data Driven Marketing | Cardinal Path
&utmhid=548221997
&utmr=-
&utmp=/
&utmcc=UA-22022050-1
&utmcc=__utma=142955815.2000699856.1300135259.1300297107.1300310649.12;+__ut
mz=142955815.1300208917.6.5.utmcsr=google|utmccn=(organic)|utmcmd=organic|utm
ctr=http://www.cardinalpath.com;
&utmu=qBAAAAAg
```

This gives you an idea of what's being written to Google. Right off the bat you can see a series of UTM codes, and a whole lot of data, which can help you troubleshoot analytics oddities that may come up as a user clicks through your website.

When we talk about Web Analytics we often talk about cookies like they're the be all end all. The last step. But cookies don't calculate, they don't execute, and they certainly don't report. Cookies in GA function a little like RAM: they hold data for computation, gathered by javascript and appended to a request to Google Analytics servers for `__utm.gif`. (I'm leaving the two underscores out from now on)

So you can look at your cookies all you want, but what's going to show in your analytics is that which is sent along with the request for `utm.gif`. (though they should be the same, right? The `UTM.gif` uses a series of query parameters (which we mentioned in the last RUGA post) appended to the image call in order to communicate your data to Google Analytics servers, making it kind of the be all end all for what data is making its way into your analytics reports.

As well as more specific data, each request can contain a series of more general queries, including:

#### **utmcc**

utmcc notes your account. (eg UA-1234567-8) I shouldn't have to explain why GA needs this.

#### **utmcc**



This string denotes the cookie values requested from that page. It includes the data from your utma and utmz cookies, or what not.

#### **utmh**

The most obvious potential problem with this has to do with caching. Most browsers will cache an image so that future requests don't need to keep asking for it from the server. If a .gif request isn't being made of the GA server, then data isn't being gathered. The utmh variable exists to prevent caching. utmh is a random number generated for each utm.gif hit, assuring that each request is original and thus the image does not get cached.

#### **utmwv**

utmwv literally just tracks the version of ga.js that sent the request. Nothing special.

#### **utmt**

There are multiple types of requests that GA can execute. utmt records the type of utm.gif hit, including custom variables, events, ecommerce items or transactions. In some cases this variable won't be sent, in which case the request is defaulted to "page".

## What is a Pageview?

Start simple right? And what is simpler than a pageview? Wait... what exactly is a pageview?

This is something that's been daunting me for a while. Pageviews are unlike a lot of stuff in Analytics in that they aren't counted in cookies, nor really in the utm.gif. A pageview is defined by the loading of a page but we all know that it's not REALLY that. Nothing in analytics is ever REALLY what it seems to be.

Quick refresher: how does GA get your data?

The page load activates the tracking code...

...tracking code writes to the browser cookies...

...the browser cookies provide data for the utm.gif query...

...and the utm.gif query is recorded by Google.

Actually, it ends up there's slightly more to it than that.

Google doesn't know when a page loads. They know when `_trackpageview()` [eg. `_trackPageview('/product334a/learn-more')`] sends a query to their system. From that they can extrapolate that a page was loaded. Then, each time the this code is executed, it creates a time stamp on Google's servers. Simple right?

You wish.

So a page was loaded. Which page? What was it's title? Where were they coming from? What host served it? A pageview itself is meaningless without other data and that's where the very concept of a "pageview" gets confusing.

So let's say Google knows that the tracking code was called, how can they determine a particular pageview?

A pageview comprises a LOT of data in the gif request.

We have the usuals:

Version (utmwv) 4.9.1  
Host id (utmhid) 1234567890  
Account (utmac) UA-10000-1  
Unique Request (utmn) 608131070

And we have the following which are more specific to the request:

Title (utmdt) Analytics Settings – Google Analytics

Referrer (utmrl) (only on the landing page)

`https://www.google.com/accounts/ServiceLoginAuth?continue=https%3A%2F%2Fwww.google.com%2Fanalytics%2Fsettings%2F&followup=https%3A%2F%2Fwww.google.com%2Fanalytics%2Fsettings%2F&service=analytics&nui=1&hl=en`

Page URL (utmp) `blog/what-is-a-pageview-ruga-part-3`



And of course a few which we will get into in later posts:

Encoding (utmcs)	UTF-8
Screen Resolution (utmsr)	1680x1050
Color Depth (utmcc)	24-bit
Language (utmll)	en-us
Java Enabled (utmje)	1
Flash version (utmfl)	10.2 r152
Hostname (utmhn)	www.google.com
utmu	DI
Domain hash(__utma)	173272373
Visitor token(__utma)	96242600
1st visit start(__utma)	February 1, 2011 5:56:12 PM
Previous visit start(__utma)	February 21, 2011 5:58:31 PM
Current visit start(__utma)	April 1, 2011 10:35:37 AM
Visit count(__utma)	3
Domain hash(__utmz)	173272373
Traffic source(__utmz)	1298339911
# of responses(__utmz)	2
utmcsr	google.com
utmccn	(organic)
utmcmd	organic
utmctt	[values from cookies]

Let's take a closer look at the values specific to this request:

utmdt – This is the page title, URL coded.

utmrl – This is the URL of the referring page (populated only by the landing page)

utmp – This is the page request

And there is actually one more hidden in here:

utmt – this is the type of request being made. Normally this would designate an event, transaction, item or custom variable. However, in the case of a pageview this field just doesn't appear – pageview is the default type of gif request.

And there you have a pageview. Not as simple as it seems.

## Event Basics

Finding cookie values a little much? Join us at eMetrics for Web Analytics the Google Way: Introduction and Overview on April 27th, and learn about measuring traffic, goals, ecommerce tracking and more.

See the event page, or register now.

Last week we briefly touched on utmt, mentioning that by default it will not show in the utm.gif request, and when this is the case the call is counted as a pageview. However, when one executes an action that is not a pageview, utmt will be labeled with the corresponding action type. Today we're going to look at the 'event' type.

Events are pretty useful tools, allowing you to define actions other than pageviews. Previously to track a non-pageview action you would need to use a virtual pageview (what a concept) which would record an action (say clicking on a button, a flash object, or menu) as a pageview. Events allow us to do this without inflating our pageview count and gives us a means of defining more specific data about what action has taken place.

## What is an event?

Events are "called" using the `_trackEvent()` method in the source code of a page object, widget, or video. One then specifies the event with a category, action, `opt_label`, and `opt_value`. Only category and action are required, but label and value are important as well.

**category** (required): This is the name of the category of the events you're trying to track.

**action** (required): This is often used to define the type of interaction the user is having with the item that is calling the event.

**label** (optional): Let's you define alternate dimensions for event data,

**value** (optional): This is a numerical entry that you can use to give more definition to the event.

This allows you to define some interesting data about users. Google provides two event tracking libraries intended to illustrate how extended event tracking can be implement via site-wide wrappers, and which also illustrate how flexible the event system can be:

TimeTracker()

MouseOverTracker()

## Why not virtual pageviews?

As I mentioned earlier, previous to events being included in GA, people would often use virtual pageviews for similar tracking tasks. Even with events, people would often use virtual pageviews for events which they wanted to attribute goals to due to, but with GAv5 you can now assign goals to



events and so there really is no reason not to use them. You no longer have to inflate your pageviews in order to track actions!

But what about the utm.gif?

To test your events, click on whatever causes one and watch your requests. You should end up with a GET request for the utm.gif with the utme query which matches the terms defined in your event. Your utme should correspond to the attributes you entered when setting `_trackEvent()`.

Here's your standard utm.gif request data for a basic event:

```
utmwv=4.9.1
utmn=1234567890
utmhn=[site url]
utmt=event
utme=5(object*action*label)(value)
utmcs=ISO-8859-1
utmsr=1680x1050
utmcs=32-bit
utmul=en-us
utmje=1
utmfl=10.2%20r154
utmdt=[title]
utmhid=1182619293
utmrr=-
utmp=[page]
utmact=UA-1234567-8
```

## User Defined Variables, Cookies, and the utm.gif

More utmt variables today, this time it's user defined variables (UDV's, aka. utmt=var). UDV's are Custom Variable's older and lonely brother. Only one of them can be set, they write via a `_setVar` command and into a utmv cookie, and they have their own .gif request. Exciting, I know!

UDV's function is the result of Google's old means of thinking about user defined segmentation and the role of stuff like "visits" and views. Under this theology a UDV defines a visit, as opposed to segmenting a current visit. The problem with this is that, since UDV's are set per visit, and since a visit starts before the `_setVar` has loaded, a UDV change will only propagate into reports at the start of the next visit. Ugly, no?

In Brian Katz's words:

```
The user defined variable, utmv, has a life of two years. It applies to the entire first visit where it is written, as well as to all future visits. It gets overwritten by any other user defined variable that it encounters, but not until after the next visit. So if a new utmv is set on every visit, the first one will be good for Visit One and Visit Two, but the one the visitor encounters on Visit Two will apply to visit Three and the one he encounters on visit Three will apply to visit Four (and so forth.)
```

```
Implementing a custom variable is fairly easy. In your GATC code you can add pageTracker._setVar("whatever");. Alternatively you can apply the same code to actions using an onClick or equivlent, then define specific types of user based on behavior. Of course you have to remember the above: that UDV's define a visit.
```

Since UDV's make their own utm.gif request, the utm.gif variables that they load are a little different.

```
utmwv=4.9.2
utms=9
utmn=1512401513
utmhn=www.analysite.net
utmt=var
utmact=UA-
```

So you have tracking code version, shockwave version, random number, the type of call (a 'var') and your account code.

"But wait," you say, "there's no actual data in there. There's just the utmt type!"

You're paying attention!

Additionally, a UDV writes the utmv cookie with [user defined variable][custom var slot]=[custom var label]=[custom var value]=[session number] (the custom var part may be repeated if there are multiple custom variables)



This then gets fed into the normal utm.gif request's extensible parameter (utm):

```
utms=16  
utm=724652773  
utmhn=www.analysite.net  
utm=8(3!whatever)9(3!1338)11(3!1)  
utmcs=ISO-8859-1
```

and so on and so forth...

And from there it's just Google.

## Really Understanding eCommerce

I found something interesting on the Google Code site today. In one part of it they had “How GIF Requests Are Classified”. Interesting, since that’s what this section of this series has been all about. In it they cite the following:

```
A GIF request is sent to the Analytics servers in the following cases and classified according to the table below. In each of these cases, the GIF request is identified by type in the utmt parameter. In addition, the type of the request also determines which data is sent to the Analytics servers. For example, transaction and item data is only sent to the Analytics servers when a purchase is made. Visitor, page, and system information is only sent when an event is recorded or when a page loads, and the user-defined value is only sent when the _setVar method is called.
```

They then go on to mark pages, events, transactions, and items as “interactions” and var’s as non-interactions, claiming:

```
Requests classified as interaction requests will impact the bounce rate calculations for your page or site. Bounce rate is referred to as a single-page visit to your site, but is strictly defined as a single interaction request during a user session. For this reason, a bounce rate for a page is also affected by ecommerce transactions and event tracking requests. This is because these features co-exist with page tracking and, when they are triggered, they result in additional interaction requests to the Analytics servers.
```

Man, that could have saved me a few posts, eh?

Anyhow, today we’re going to talk about the last two in that list: trans and items. These are added using `_addTrans()` and `_addItem()` functions.

`_addTrans()` initializes a transaction object, which stores all your transaction data (billing address, shipping charges, etc.). These are then associated with items via an order ID. Makes sense right? You can have multiple items within one transaction. Think of this as your shopping cart.

`_addItem()`, on the other hand, tracks information specific to each item in a shopping cart. This can be SKUs, price, quantity, etc.

Specifically, trans tracks:

- order ID – required
- affiliation or store name
- total – required
- tax
- shipping
- city
- state or province

While item tracks:

- order ID (required)
- SKU/code (required)
- product name
- category or variation
- unit price (required)
- quantity (required)

I used an ordered list here for a reason. Each property must be listed in the correct order. If you don't use that property then you will have to list it differently. eg. `_addItem("12345", "46789", "", "", "24.99", "1");`

Remember, according to Google:

If a transaction contains multiple items and the SKU is not supplied for every item, a GIF request is sent only for the last item added to the transaction for which a SKU is provided. In addition, if your inventory has different items with the same SKU, and a visitor purchases both of them, you will receive data for only the most recently added. For this reason, you should make sure that each item you offer has a unique SKU.

So, what should your `_addTrans` and `_addItem`'s look like:

```
_gaq.push(['_addTrans',
'1234',           // order ID - required
'Acme Clothing', // affiliation or store name
'11.99',         // total - required
'1.29',         // tax
'5',            // shipping
'San Jose',     // city
'California',   // state or province
'USA'           // country
// add item might be called for every item in the shopping cart
// where your ecommerce engine loops through each item in the
// cart and
// prints out _addItem for each
_gaq.push(['_addItem',
'1234',           // order ID - required
'DD44',          // SKU/code - required
'T-Shirt',       // product name
'Green Medium', // category or variation
'11.99',         // unit price - required
'1'              // quantity - required
]);
_gaq.push(['_trackTrans']); //submits transaction to the
Analytics servers
```



Associated with an ecommerce transaction is going to be the following values:

- utmtid – your OrderID
- utmtst – affiliation
- utmtto – total value for a unit
- utmttx – total tx
- utmtsp – shipping cost
- utmtci – billing city
- utmtrg – billing reigon
- utmtco – billing country

So, you set your `_addTrans` and `_addItem`, your visitor visits, visits the right pages, and those make calls through `_addTrans` and `_addItem` to Google Analytics, which then writes them into a .gif request and...

...your utm.gif should look something like the following:

Per Transaction:

[http://www.google-analytics.com/\\_\\_utm.gif?utmwv=4.9.2](http://www.google-analytics.com/__utm.gif?utmwv=4.9.2)

```
utms=6
utmn=1684715033
utmhn=store.analysite.net
utmt=tran
utmtid=1304117814682
utmtst=Northern%20Warehouse
utmtto=535.52
utmttx=37.52
utmtsp=29.00
utmtci=Portland
utmtrg=Oregon
utmtco=USA
utmac=UA-9999-9
utmcc=__utma[... ]
utmdu=DRALAAAwE~
```

Per Item:

```
http://www.google-analytics.com/__utm.gif?utmwv=4.9.2
utms=5
utmn=1533116887
utmhn=www.analysite.net
utmt=item
utmtid=1304117435168
utmipc=STATIC-456
utmipn=Tea%20bags
utmiva=military%20issue%2C%20Earl%20grey
utmipr=53
utmigt=4
utmac=UA-9999-9
utmcc=__utma[... ]
utmdu=DRALAAAwE~
```

And there you have it. We're done with request types!

## Visits and Visitors

This is such an important concept in analytics that I actually feel ashamed that I had not, until now.

As such I am going to take a break from our regularly schedule to do some explaining of what those silly metrics like visits are. Today, visits and their co-concept sessions.

How does GA Define a visit?

Note: Since this post was originally written Google has changed the definition of a visit. This section has been rewritten to incorporate that.

So what is a visit? Some one coming to see your website, right? A Session? The period of time where they are visiting. And what is a visitor? It is the some one who is coming to see your website. Simple, but again not.

Again we hit that ever present definition problem, but this time it's because we really don't have a well conceptualized definition for just what a visit is. A visit seems easy for us to **imagine**: Michael Stralker comes over for a game of Scrabble; he came for a visit. But what if he left to get beer, then came back. Is that another visit?

We might argue that a visit is based on the intent: if he left with the intent of ending his visit then came back later for another then it's two visits. If he came, then went out to grab something to enhance his visit, it's the same visit. This makes some sense.

Sadly we have to guess intent online.

To do this Google Analytics makes some pretty drastic assumptions about a visit, but for most sites they should be fairly accurate.

If a user **comes from a new source (imagine Michael goes to the bar, then comes back later) then it's a new visit.**

If a user hasn't done anything in 30 minutes, the visit is over (though this can be changed, see `_setSessionCookieTimeout()`)

So if a user goes away, does something else, then comes back (like Michael goes to the bar) it's a new visit. If the user goes away back to wherever he came from and comes back within 30 minutes (say, Michael runs home quickly), the same. In the past, if the user closes their browser (say, Michael... uhh... goes into a coma?) then opens it again, new visit, but that doesn't quite make the same amount of sense, now does it?

These functions were once governed by two cookies with utmc handling the browser close and utmb handling the 30 minute time out. However, now that visits are no longer defined by a browser close, utmc seems all but useless.

utmb looks like this:

```
24489125.12.10.1305922281266
24489125 = Domain Hash
12 = Number of Pageviews in current session
10 = countdown of outbound links clicked (to be honest I have no idea why)
```



1305922281266 = timestamp of session start

utmcc only ever consisted of the domain hash.

That's it right? Nope. utma actually played a roll here too as a "storage" cookie, recording the session time stamps and session numbers (aka. visits).

So some one visits your site, exits, comes back. What's happening? At this point (if you've been following this series) you should be able to figure it out.

- Visitor visits site and executes javascript.
- utmb&c are written, a time stamp is noted, starting a session.
- Data is noted in utma, which is recorded in the utmcc call of the utm.gif request, and recorded by google.
- Visitor idles for 30 minutes, then clicks a link.
- By the end of the 30 minute span the utmb cookie has expired and a new one must be written, creating a new session.
- Visitor closes browser, utmc is erased.
- Visitor comes back, a new utmc cookie is created, a new session is started.
- 

As with all cookie based tracking, the Achilles heel of this system is that when people clear their cookies it restarts the system, but that isn't as big an issue as one might think given that they would likely be a microscopic percentage of users and we're dealing with numbers in aggregate.

## Direct Visitors

Now we're done with the specifics of how utm.gif requests are classified, we get to move on to the fun stuff. And today is the fun stuff let me tell you. Today we're going to start looking at traffic sources by looking at what it means to be direct traffic.

I want to credit a lot of this post to a coworker of mine, Brian Katz, and his wonderful post on the VKI Studios Blog, Tips Tricks, Traps and Tools: #4.1 of many: Understanding Direct Visits in GA & Web Analytics : UDVs

## Direct Traffic

What does it mean for a visitor to be direct visitor? The standard response is that its they typed the URL in, or entered via a bookmark. However, this definition is woefully inaccurate.

Google Analytics (and some other analytics tools) start with the presumption that direct traffic is traffic that does not have an http referrer, but then they modify it ("muddy it" is how Katz puts it) with the UTMZ cookie so that visits that would normally be considered direct are considered "external" if they have a campaign tag attached or a referrer recorded from a previous visit. This is a good thing, since you should distinguish between direct and external visits at times. However it also gives new meaning to direct traffic: direct is when no external source is already recorded.

This means that, from Brians post:

- a visitor, whose last visit(s) were from external sources, returns directly. GA does not overwrite the traffic source cookie and the traffic source is still credited with the visit.
- a visitor, whose previous visit was recorded as direct, returns via an external visit source will have their External Traffic Source cookie overwritten to record the (new) external traffic source.

## utmz

The utmz cookie is often considered the "campaign" cookie.

Here's a cookie from the Cardinal Path Blog:

```
113869458.1305319923.101.101.utmcsr=cardinalpath.com|utmccn=(referral)|utmcmd=referral|utmcc  
t=/blog/disposable-printers-is-there-a-better-option
```

How do you read this? The first number is the domain hash, the second the time stamp, the third is the session number (101? Wow) and the fourth is the campaign number. Then, on the end, are some familiar looking queries, eh?

- utmccn is the campaign value of the visit. In this case, referral.
- utmcmd is the medium, again referral.
- utmctr is the keywords (if there is one, there isnt here)



- utmct is campaign content

and sometimes you'll get a utmgclid which defines your ad click ID.

## Making traffic direct

Because of this definition, a lot of things that could be considered referrals can be considered direct instead. For instance, maybe a user types in your domain name into Google. Are these people 'effectively' searching for your domain, or should they be considered direct?

Well, you can use the `_addIgnoredOrganic()` to define people searching for a specific keyword as direct traffic.

From Google:

```
_addOrganic()
_addOrganic(newOrganicEngine, newOrganicKeyword, opt_prepend)
```

Adds a search engine to be included as a potential search engine traffic source. By default, Google Analytics recognizes a number of common search engines, but you can add additional search engine sources to the list.

parameters

```
String    newOrganicEngine Engine for new organic source.
String    newOrganicKeyword Keyword name for new organic source.
boolean   opt_prepend If true prepends the new engine to the beginning of
the organic source list. If false adds the new engine to the end of the
list. This parameter's default value is set to false.
```

Katz provides the following option as well:

```
Identifying Direct Visitors
// before var pageTracker = _gat._getTracker...
var isDV = (!document.referrer) &&
(document.cookie.search(/(__utm[cb]=)/g) == -1 ||
document.cookie.match(/(__utm[cb]=)/g).length <= 1
);
...

// after var pageTracker = _gat._getTracker and
// after any pageTracker._set*() function calls ...
if (isDV) pageTracker._setVar('Direct Visitors');
```

That's it.

The snippet checks that there is no referrer. If there is a referrer, this page is either not the landing page or the visit came from an external traffic source. So, the last 2 lines check that there are no visit cookies and confirms we are on a landing page. On subsequent pages there are referrers from the current site but the visit cookies would already have been set.

No UTM.gif report this time, as it doesn't really matter.

## Really Understanding Referrals

Pretty basic one today: referring traffic. We all know what referring traffic is. That's easy, right? But how does Google Analytics determine referring traffic? What is it at a cookie level? Well you should know the header basics:

Here's a sample 'GET':

```
GET / HTTP/1.1
Host: www.cardinalpath.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.13)
Gecko/20101203 Firefox/3.6.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://blog.vkistudios.com/
```

So the browser calls on cardinalpath.com, sending the referrer from blog.vkistudios.com. Simple enough. Meanwhile, Google Analytics calls on document.referrer via javascript and recalls the result.

But what happens after that? First data is written into the utmz cookie. From a couple of weeks ago:

```
utmz
```

The utmz cookie is often considered the "campaign" cookie.

Here's a cookie from the Cardinal Path Blog:

```
113869458.1305319923.101.101.utmcsr=cardinalpath.com|utmccn=(referral)|utmcmd=referral|utmctt=/blog/disposable-printers-is-there-a-better-option
```

How do you read this? The first number is the domain hash, the second the time stamp, the third is the session number (101? Jesus) and the fourth is the campaign number. Then, on the end, are some familiar looking queries, eh?

```
utmccn is the campaign value of the visit. In this case, referral.
utmcmd is the medium, again referral.
utmctr is the keywords (if there is one, there isnt here)
utmctt is campaign content
and sometimes you'll get a utmgclid which defines your ad click ID.
```

Then if no campaign variables are available that would overwrite the referrer data, the referring URL is written into utmr on the utm.gif request, and bam, you have a referred site. Simple right?

But there's more you can do with this, because Google knows that you want to be able to control what counts as referrals, and what counts as direct.

```
_addIgnoredOrganic()
_addIgnoredOrganic(newIgnoredOrganicKeyword)
```



Sets the string as ignored term(s) for Keywords reports. Use this to configure Google Analytics to treat certain search terms as direct traffic, such as when users enter your domain name as a search term. When you set keywords using this method, the search terms are still included in your overall page view counts, but not included as elements in the Keywords reports.

```
_addIgnoredRef()
```

```
_addIgnoredRef(newIgnoredReferrer)
```

Excludes a source as a referring site. Use this option when you want to set certain referring links as direct traffic, rather than as referring sites. For example, your company might own another domain that you want to track as direct traffic so that it does not show up on the "Referring Sites" reports. Requests from excluded referrals are still counted in your overall page view count.

```
_addOrganic()
```

```
_addOrganic(newOrganicEngine, newOrganicKeyword, opt_prepend)
```

Adds a search engine to be included as a potential search engine traffic source. By default, Google Analytics recognizes a number of common search engines, but you can add additional search engine sources to the list.

For more on these functions and their implementation see [Tracking Code: Search Engines and Referrers](#).

So to summarize, header information is read from document.referrer and written into cookies. If there are no campaign variables that would overwrite the utmr query, then it's written into there on the utm.gif and sent away.

## Understanding Campaigns

So last chapter I discussed referrers and mentioned that the utmr tag will be populated by the referrer unless there is a campaign variable to overwrite it. In this one we're going to look at campaign tags and their roles.

Campaign's in Google Analytics are a fairly simple concept. When a visitor enters your site through a tagged link, the tracking code parses the link data and writes it into our old favorite, the utmz cookie – specifically into the utmccn, utmcmc, utmctr, and utmccd (and sometimes utmgclid) – and the utm.gif request.

Available tags are: (source: [Google Analytics Help](#))

Campaign Source (utm\_source) – Required. Use utm\_source to identify a search engine, newsletter name, or other source.

Example: utm\_source=google

Campaign Medium (utm\_medium) – Required. Use utm\_medium to identify a medium such as email or cost-per-click.

Example: utm\_medium=cpc

Campaign Term (utm\_term) – Used for paid search. Use utm\_term to note the keywords for this ad.

Example: utm\_term=running+shoes

Campaign Content (utm\_content) – Used for A/B testing and content-targeted ads. Use utm\_content to differentiate ads or links that point to the same URL.

Examples: utm\_content=logolink or utm\_content=textlink

Campaign Name (utm\_campaign) – Used for keyword analysis. Use utm\_campaign to identify a specific product promotion or strategic campaign.

Example: utm\_campaign=spring\_sale

Fortunately for us, Google provides a [URL building tool](#) for those who would rather not much around with their URLs manually. It's easy to use, though for larger campaigns you may want a campaign building spreadsheet. Fortunately for you [Justin Cutroni has one over at his blog](#).

Tagging links literally just ads information to the URL, then forwards that information to the sites GA code, so there's no need to fiddle with the GA back end. This also means that you can use them with any site that has GA on it, and it will show up in their analytics.

For instance, if we wanted to we could toss some variables into Youtube's analytics with:



## Tool: URL Builder

### Google Analytics URL Builder

Fill in the form information and click the **Generate URL** button below. If you're new to tagging links or this is your first time using this tool, read [How do I tag my links?](#)

If your Google Analytics account has been linked to an active AdWords account, there's no need to tag your AdWords links - [auto-tagging](#) will do it for you automatically.

Step 1: Enter the URL of your website.

Website URL: \*   
(e.g. <http://www.urchin.com/download.htm>)

Step 2: Fill in the fields below. **Campaign Source**, **Campaign Medium** and **Campaign Name** should always be used.

Campaign Source: \*  (referrer: google, citysearch, newsletter4)

Campaign Medium: \*  (marketing medium: cpc, banner, email)

Campaign Term:  (identify the paid keywords)

Campaign Content:  (use to differentiate ads)

Campaign Name\*:  (product, promo code, or slogan)

Step 3

That would result in the following URL:

[http://www.youtube.com/?utm\\_source=HEY\\_GUYS&utm\\_medium=HOW\\_ARE&utm\\_term=YOU&utm\\_content=DOIN&utm\\_campaign=LOVE\\_CP](http://www.youtube.com/?utm_source=HEY_GUYS&utm_medium=HOW_ARE&utm_term=YOU&utm_content=DOIN&utm_campaign=LOVE_CP)

Go ahead, click that URL. Send Youtube a message. (ed note: actually, Youtube's front page doesn't use standard utm based Google analytics tracking. There are no signs of GA cookies or UTM.gif requests. Given the amount of traffic that Youtube get's, I'm not surprised.)

I'll end this with the utm.gif. As always these variables write into the utm.gif request along with other relevant data. In this case some or all of the following:

- utmcmn = "utm\_new\_campaign (1)"
- utmccn = Campaign Name
- utmcmd = Campaign Medium (direct), (organic), (none)
- utmcsr = Campaign Source
- utmcct = Campaign Content
- utmcid = Campaign ID
- utmctm = Campaign Mode (0|1)
- utmcto = Campaign Timeout

Here's an example (using Cardinal Path, since it looks like Youtube wont work)

[http://www.google-analytics.com/\\_\\_utm.gif?utmwv=4.9.4](http://www.google-analytics.com/__utm.gif?utmwv=4.9.4)

```
&utms=1
&utmn=1082105711
&utmhn=www.cardinalpath.com
&utme=8(author)9(cp_admin)
&utmcs=UTF-8
&utmcsr=1680x1050
```



&utmcs=32-bit  
&utmcl=en-us  
&utmje=1  
&utmfl=10.3%20r181  
&utmdt=Cardinal%20Path%20-  
%20Web%20Analytics%20and%20Data%20Driven%20Marketing%20  
%7C%20Cardinal%20Path  
&utmhid=1185577042  
&utmcr=-  
&utmp=%2F%3Futm\_source%3DHEY\_GUYS%26utm\_medium%3DHOW\_ARE%26utm\_term%3DYOU  
%26utm\_content%3DDOIN%26utm\_campaign%3DLOVE\_CP  
&utmacc=UA-22022050-1  
&utmcc=  
\_\_utma%3D23120357.1797963222.1307118009.1307141433.1307145319.6%3B%2B  
\_\_utmz%3D23120357.1307145319.6.2.  
utmcsr%3DHEY\_GUYS  
%7Cutmccn%3DLOVE\_CP  
%7Cutmcmd%3DHOW\_ARE  
%7Cutmctr%3DYOU  
%7Cutmcct%3DDOIN%3B  
&utmu=qR~

## `_tracksocial`

An event by any other name tracks as well, and so `_tracksocial` tracks much as any event does – `utm.gif` call and all.

[Nick Iyengar points out](#) that `tracksocial` doesn't seem to add much that you couldn't already do with events. However, [as he also notes](#), having a dedicated set of reports for social interactions both gives room for social to grow, as well as providing some nice clean reports to work with. It also introduces a whole bunch of `utm.gif` calls and settings for analytics geeks to play with.

In this post we're going to explore some of those.

## Setting It Up

Google provides a wonderful table, which I have reproduced below, illustrating the parameters involved with `_tracksocial`.

<code>network</code> ( <i>String</i> )	The network on which the action occurs (e.g. <i>Facebook, Twitter</i> )
<code>socialAction</code> ( <i>String</i> )	The type of action that happens (e.g. <i>Like, Send, Tweet</i> ).
<code>opt_target</code> ( <i>String</i> )	Optional. The text value that indicates the subject of the action; most typically this is a URL target. If undefined, defaults to <code>document.location.href</code> . For example, if a user clicks the Facebook <i>Like</i> button for a news article, the URL for that news article could be sent as the string. You could also supply any other ID identifying the target, such as an ID from your content management system. As another example, a user could click the <i>Like</i> button next to a blog post, in which case no value need be sent for this parameter because the current document location URL is sent by default.  <b>Note:</b> If you want to leave this value undefined at the same time you want to supply a value for the next parameter ( <code>opt_pageUrl</code> ), you must specify <code>undefined</code> in your code.
<code>opt_pagePath</code> ( <i>String</i> )	Optional. The page (by path, not full URL) from which the action occurred. If undefined, defaults to <code>document.location.pathname</code> plus <code>document.location.search</code> . You will only need to supply a value for this string if you use virtual (or custom) page paths for Analytic reporting purposes. When using this option, use a beginning slash (/) to indicate the page URL. For more information, see " <a href="#">Virtual URLs</a> " in <i>Typical Customizations</i> .

## How it works



\_tracksocial functions by making a gaq.push call, sending a utm.gif to Google Analytics. You can check out the utm.gif call it makes at the [Google Analytics Social Tracking Demo](#). I've also included a sample below.

```
http://www.google-analytics.com/__utm.gif?utmwv=5.1.2b

&utm=4
&utmn=1267893522
&utmhn=analytics-api-samples.googlecode.com
&utmt=social
&utmsn=facebook
&utmsa=unlike
&utmsid=http%3A%2F%2Fanalytics-api-
samples.googlecode.com%2Fsvn%2Ftrunk%2Fsrc%2Ftracking%2Fjavascript%2Fv5%2
Fsocial%2Ffacebook_js_async.html
&utmcs=UTF-8
&utmsr=1680x1050
&utmcs=32-bit
&utmul=en-us
&utmje=1
&utmfl=10.3%20r181
&utmdt=Social%20Demo
&utmhid=126490777
&utmr=http%3A%2F%2Fwww.google.com%2Furl%3Fq%3Dhttp%253A%252F%252Fanal
ytics-api-
samples.googlecode.com%252Fsvn%252Ftrunk%252Fsrc%252Ftracking%252Fjavascr
ipt%252Fv5%252Fsocial%252Ffacebook_js_async.html
&utmp=%2Fsvn%2Ftrunk%2Fsrc%2Ftracking%2Fjavascript%2Fv5%2Fsocial%2Ffa
cebook_js_async.html
&utmcc=UA-12345-6
&utmcc=__utma%3D100125134.1395102488.1312409837.1312409837.1312409837
.1%3B%2B__utmz%3D100125134.1312409837.1.1.utmcsr%3Dgoogle%7Cutmccn%3D(org
anic)%7Cutmcmd%3Dorganic%7Cutmctr%3Dhttp%253A%252F%252Fanalytics-api-
samples.googlecode.com%252Fsvn%252Ftrunk%252Fsrc%252Ftracking%252Fjavascr
ipt%252Fv5%252Fsocial%252Ffacebook_js_async.html%3B
&utm=AAAAAAAAAAAAAAAAAAE~
```

With this request we get the following new parameters, and very little documentation on what they do.

**utms:** Technically not new since I see references to it online going back to April. It appears to be some kind of internal counter, but I don't know what it tracks. Some claim that once it hits 500, GA no longer makes utm.gif requests.

**utmsn:** Social network. Labels the campaign that the request is associated with the "network" string above.

**utmsa:** The social action taken

**utmsid:** The path defined by opt\_pagePath

## Extras: Technical Details

Are we here already, end of the guide? We're done with the main aspects of the utm.gif and on to the tail end of RUGA. The "extras" section, if you will. These are the loose odds and ends that Google Analytics collects for technology reports. These are all pretty simple, and so this will be a short post.

The sequence for collecting this information is much the same as what we've already covered. The GATC executes on page load, javascript collects a bunch of data about the clients system, and those are then reported to Google Analytics through a utm.gif request. This doesn't write into cookies, it just goes straight into the query request.

Below is an example of the utm.gif call that is made when you load cardinalpath.com:

```
http://www.google-analytics.com/__utm.gif?  
  
  utmwv=4.9.4  
  &utms=3  
  &utmhn=426445686  
  &utmhn=www.cardinalpath.com  
  &utmcs=UTF-8  
  &utmsr=1680x1050  
  &utmcs=32-bit  
  &utmhl=en-us  
  &utmje=1  
  &utmfl=10.3%20r181  
  ...
```

You can see here:

- utmwv = tracking code version
- utmcs = character encoding
- utmsr = "Screen Resolution"
- utmsc = "Screen Colors"
- utmul = "Language (lang|lang-CO|-)"
- utmje = "JavaScript Enabled? (0|1)"
- utmfl = "Flash version"

Sometimes you'll also get utmjv, which tells your JavaScript version.

This data is processed on Google's side, populated into reports and made available to you. You can find most of it under "Visitors > Technology" in the V5 interface.

And that actually concludes this series... for now.



# THE RUGGA CHEAT SHEET

Variable	Description
utmacc	Account String.
utmcc	Cookie values.
utmccn	Starts a new campaign session
utmcr	Indicates a repeat campaign visit.
utmcs	Language encoding for the browser.
utmdt	Page title
utme	Extensible Parameter
utmfl	Flash Version
utmhn	Host Name
utmipc	Product Code
utmipn	Product Name
utmipr	Unit Price
utmigt	Quantity
utmiva	Variations on an item
utmje	Is Java-enabled (1 0)
utmn	Unique ID
utmp	Page request of the current page.
utmr	Referral
utmssc	Screen color depth
utmsr	Screen resolution
utmt	Type of request (event,transaction, item, or custom variable)
utmtci	Billing City
utmtco	Billing Country
utmtid	Order ID
utmttg	Billing region
utmtsp	Shipping cost.
utmtst	Affiliation.
utmtto	Total
utmttx	Tax
utmul	Browser language
utmwv	Tracking code version

utma = 123456789.123456789.1234567890.1234567890.1234567980.1

Domain Hash	Visitor ID	Initial Visit	Previous Session	Current Session	Session #
-------------	------------	---------------	------------------	-----------------	-----------

utmb = 123456789.9.10.1234567890

Domain Hash	Time Stamp	Outbound Link Countdown	Domain Hash
-------------	------------	-------------------------	-------------

Page Views (this session)

utmz = 123456789.1234657890.1.4.utmcsr=X|utmccn=Y|utmcmd=Z|

Domain Hash	Time Stamp	Session #	Campaign #
-------------	------------	-----------	------------

### Pageviews:

\_trackpageview(URL) - tracks the cited url as a pageview

### Traffic Sources:

\_addIgnoredOrganic(Keyword) - Defines ignored term(s) in Keywords reports, treating them as direct traffic.

\_addIgnoredRef(Referrer) - Registers traffic from a specific referring source as direct traffic.

\_addOrganic(engine, Keyword, prepend[true|false]) - Registers traffic from a source as search engine traffic.

### Ecommerce:

\_addTrans(orderID, affiliation, total, tax, shipping, city, state, country) - initializes a transaction object. Associated with items via an order ID.

\_addItem(orderID, SKU, name, category, price, quantity) - tracks information specific to each item in a shopping cart.