

Seamless Marketo-Google Analytics Integration

Google Analytics is the ubiquitous digital analytics platform used by marketers to measure and optimize their effort against defined objectives. Marketo is a leading marketing automation software found on thousands of websites. The best of both solutions can be leveraged to gain a 360-degree view of your lead generation workflow. This cheat-sheet explains how to do it.

About the author

Stéphane Hamel is a seasoned consultant and distinguished thought leader in the field of digital analytics. Named Most Influential Industry Contributor by the Digital Analytics Association (2013), he has made significant contributions to the industry. Strong of twenty-five years' experience empowering organizations to analyze and optimize their online channels, he has cemented his position as a leading voice for digital analytics and online optimization. He is frequently called upon to speak at events around the world and an often-quoted media contributor on the topic of digital intelligence and analysis. Stéphane hold an MBA specialized in eBusiness, teaches a graduate class on digital analytics, and serve on the advisory for a number of organizations.

Stéphane Hamel, MBA - "data is the raw material of my craft"
Digital Analytics Thought Leader
immeria.net & digitalanalyticsmaturity.org
Email: shamel@immeria.net
LinkedIn: linkedin.com/in/shamel
G+: [google.com/+StephaneHamel-immeria](https://plus.google.com/+StephaneHamel-immeria)
Twitter: [@SHamel67](https://twitter.com/SHamel67)
Skype: [SHamel67](https://www.skype.com/people/SHamel67)

Content

High-level Use Cases	2
Track Marketo Form Submission in Google Analytics	2
End-to-End Lead Tracking	4
Marketo leads via REST API	6
Extend the concept to Marketo Lead workflows	7
Additional References	7

V1.0 – Last modified Monday, August 24, 2015.

High-level Use Cases

Visitors fill out Marketo Forms to download white papers, subscribe to newsletter, request trials or services, etc.:

- Use a generic and consistent method to track successful form submissions - which, in some cases, don't end up with a "thank you" page.

Once the lead is received, various rules and human intervention are required to qualify the lead and either reject it, proceed to enable a trial, or initiate the sales & billing process after discussing with the client:

- Track key events affecting the lead, such as a status change – and enable "post back" to Google Analytics using events or ecommerce transactions.

Solution Outline

- The [Marketo Forms 2.0](#) onSuccess() handler can be used to trigger a Google Tag Manager custom event, and from there, track the "thank you" as a Google Analytics virtual page view or event;
- A hidden form field can hold the [Google Analytics Client ID](#). Whenever Marketo receives a lead, a Webhook uses the [Google Analytics Measurement Protocol](#) to register an event against Google Analytics;

- Any relevant workflow can be modified to include a Webhook to log a Google Analytics event with updated status and other relevant information, or even an ecommerce transaction with all the details;
- For cases where a lead is generated using the [Marketo REST API](#), the concept can still be applied successfully.

Metrics & Analysis

Through this implementation, you will gain the following metrics and unlock new analysis capabilities:

- Events or virtual page views for successfully submitted leads;
- Ability to track leads as goals and visualize their conversion funnels;
- A custom dimension to track the specific form ID being used;
- A custom dimension to track the user-level Marketo lead id;
- Offline conversion and ecommerce tracking;
- Ability to optimize your marketing & retargeting based on specific user behavior.

Track Marketo Form Submission in Google Analytics

When creating your Marketo form you can elect to have a "Thank You Page" or create a more sophisticated workflow. By default, the user is just presented with a notice in an overlay. The technique presented here leverage the Marketo Forms 2.0 API in order to have a generic and consistent method of tracking successful forms submissions.

Following those simple steps will allow you to track a virtual page view or an event in Google Analytics.

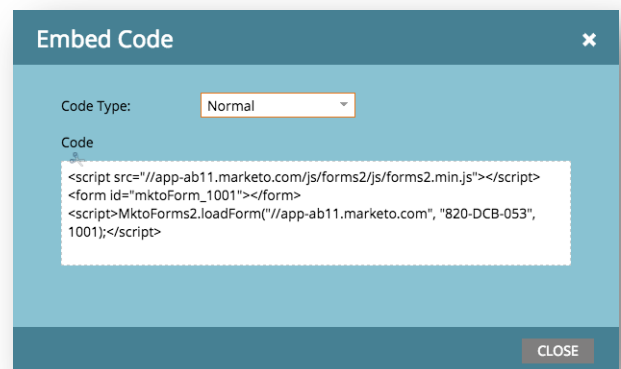
① *Google Tag Manager (GTM) is the preferred method of deploying and managing Google Analytics (GA) tags and demonstrated throughout this document.*

1. Create your Marketo Form

Under the Marketo Design Studio, proceed as usual to create your.

2. Integrate the Marketo Form to your Website

Embed the form as indicated under Embed Code.



3. Track Form Submissions

When the Marketo Form is submitted successfully, a `dataLayer.push()` is used to tell GTM about this successful user interaction. We will also pass the form id so we can keep track of the exact form that was submitted.

CHEAT-SHEET: END-TO-END MARKETO AND GOOGLE ANALYTICS INTEGRATION

The default JavaScript embed code looks as follow:

```
<script src="//app-ab11.marketo.com/js/forms2/js/forms2.min.js"></script>
<form id="mktoForm_1001"></form>
<script>MktoForms2.loadForm("//app-ab11.marketo.com",
"820-DCB-053", 1001);</script>
```

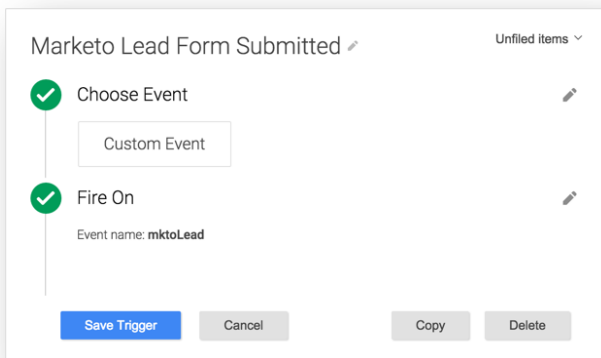
Modified code:

```
<script src="//app-ab11.marketo.com/js/forms2/js/forms2.min.js"></script>
<form id="mktoForm_1001"></form>
<script>
MktoForms2.loadForm("//app-ab11.marketo.com", "820-DCB-053", 1001, function(form){
  form.onSuccess(function(values, followUpUrl){
    dataLayer.push({event: 'mktoLead',mktoFormId:
form.getId()});
    // rest of your success handler...
    return true;
  });
});
</script>
```

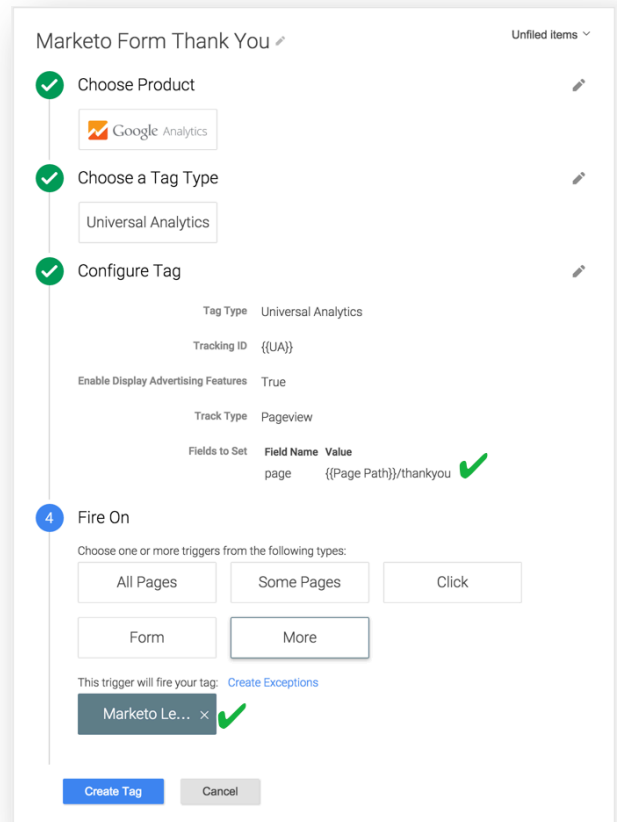
4. Track an event or virtual page view

When GTM receives this custom event, a UA virtual pageview or event can be used to indicate the lead was successfully submitted.

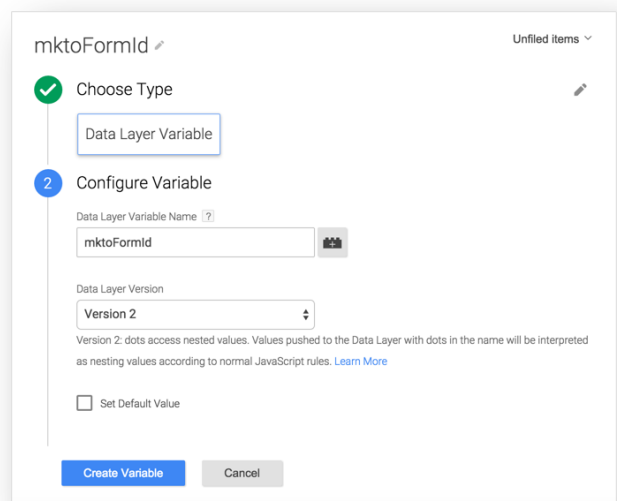
- Create a GTM trigger for “custom event equals mktoLead” named “Marketo Lead Form Submitted”



- As mentioned previously, by default Marketo forms don't show a “thank you” page and do not refresh the current page. In this case, you might want to track a GA Virtual Pageview. If there's a “thank you” page set for this form, a regular pageview should be tracked there, otherwise:
 - Create a GA Pageview tag
 - Under “More settings/Fields to Set”, set the “page” field with the value of the variable {{Page Path}} followed by an indication such as “/thankyou”
 - Set this to Fire On “Marketo Lead Form Submitted” custom event trigger

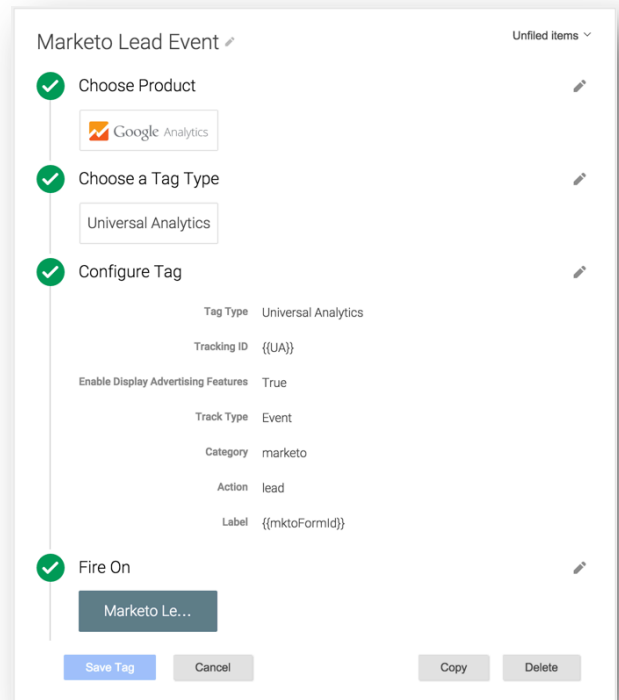


- Alternatively, you can fire a GA event:
 - Create a new Variable named “mktoFormId” of type dataLayer, with Data Layer Variable Name “mktoFormId”



CHEAT-SHEET: END-TO-END MARKETO AND GOOGLE ANALYTICS INTEGRATION

- Create a GA Event tag
 - Set the Category to “marketo”
 - Set the Action to “lead”
 - Set the Label to your Variable named “{{mktoFormId}}”
- Set this to Fire On “Marketo Lead Form Submitted”



End-to-End Lead Tracking

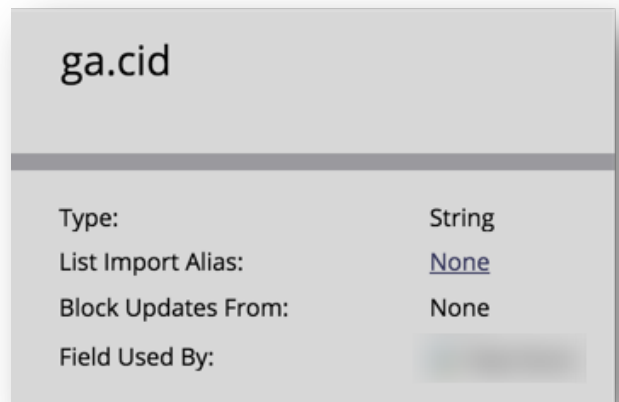
Following these steps will:

- a) send the Google Analytics client id to Marketo and;
- b) allow Marketo to send back the Marketo Unique Code (lead reference) as an event back to Google Analytics.

1. Create a hidden field to hold the Google Client ID value

Under Marketo Admin/Field Management, create a new field

- Name it ga.cid, and set the type to String



2. Create a GA Custom Dimension

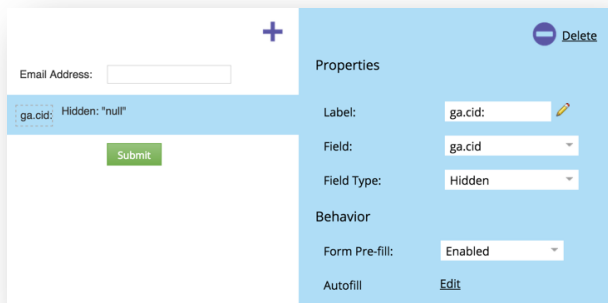
- Under GA Admin/Property/Custom Definitions/Custom Dimensions, create a new entry for Marketo Lead with a “Hit” scope
- Take note of the Index generated by GA, this will be used with the Webhook defined in step 5

CHEAT-SHEET: END-TO-END MARKETO AND GOOGLE ANALYTICS INTEGRATION

① You could set the scope to "Session" or even "User" if you expect to receive a single lead per session, or even for a given user.

3. Add the ga.cid field to your Marketo form

Add the ga.cid field to your Marketo form and hide it



The screenshot shows the Marketo form editor interface. On the left, a form preview shows an 'Email Address' field and a 'Submit' button. A 'ga.cid' field is added below, set to 'Hidden: null'. On the right, the 'Properties' panel shows the 'Label' as 'ga.cid', 'Field' as 'ga.cid', and 'Field Type' as 'Hidden'. The 'Behavior' panel shows 'Form Pre-fill' as 'Enabled' and 'Autofill' as 'Edit'.

4. Send the Google Analytics Client ID to Marketo

Modify the form initialization script to automatically populate the ga.cid hidden field with the GA CID value.

① Do not read the value of the `_ga` cookie directly, always use the proper `tracker.get()` technique.

Modified code:

```
<script src="//app-ab11.marketo.com/js/forms2/js/forms2.min.js"></script>
<form id="mktoForm_1001"></form>
<script>
MktoForms2.loadForm("//app-ab11.marketo.com", "820-DCB-053", 1001, function(form) {
  // Set the value of the GA client id
  function getgacid() {
    try {
      var tracker = ga.getAll()[0];
      return tracker.get('clientId');
    } catch (e) {
      return 'n/a';
    }
  }
  form.vals({
    'gacid': getgacid()
  });
});
</script>
```

4b. Combining GA event with GA client id

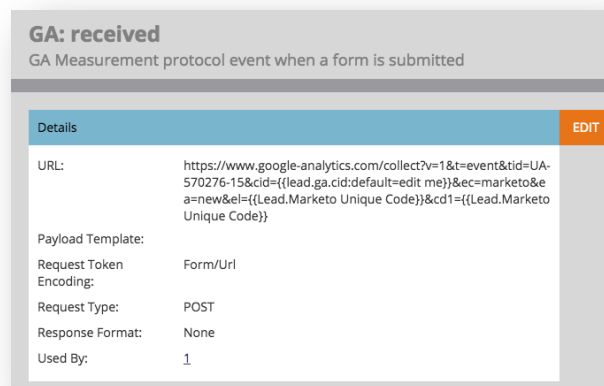
In this step, we combine the form initialization with the `onSuccess()` handler previously mentioned.

```
<script src="//app-ab11.marketo.com/js/forms2/js/forms2.min.js"></script>
<form id="mktoForm_1001"></form>
<script>
MktoForms2.loadForm("//app-ab11.marketo.com", "820-DCB-053", 1001, function(form){
  form.onSuccess(function(values, followUpUrl){
    dataLayer.push({event: 'mktoLead',mktoFormId: form.getId()});
    // rest of your success handler...
    return true;
  });
  // Set the value of the GA client id
  function getgacid() {
    try {
      var tracker = ga.getAll()[0];
      return tracker.get('clientId');
    } catch (e) {
      return 'n/a';
    }
  }
  form.vals({
    'gacid': getgacid()
  });
});
</script>
```

5. Create a Webhook

We want to register a Google Analytics event whenever a form is received and send the Marketo Lead id to GA. We need to leverage the GA Measurement Protocol to register an event of Category: "marketo", event Action: "new", event Label: the unique ID of this lead.

- Under Admin/Integration/Webhooks, create a new webhook named "GA: received"



The screenshot shows the Marketo Webhook configuration page for a webhook named "GA: received". The description is "GA Measurement protocol event when a form is submitted". The "Details" section shows the following configuration:

Details	EDIT
URL:	https://www.google-analytics.com/collect?v=1&t=event&tid=UA-570276-15&cid={{lead.ga.cid:default=edit me}}&ec=marketo&ea=new&el={{Lead.Marketo Unique Code}}&cd1={{Lead.Marketo Unique Code}}
Payload Template:	
Request Token	Form/Url
Encoding:	
Request Type:	POST
Response Format:	None
Used By:	1

The URL of the web hook is a GA Measurement Protocol request with the following components:

```
https://www.google-analytics.com/collect?v=1&t=event&tid=UA-999999-9&cid={{lead.ga.cid:default=edit me}}&ec=marketo&ea=new&el={{Lead.Marketo Unique Code}}&cd1={{Lead.Marketo Unique Code}}
```

CHEAT-SHEET: END-TO-END MARKETO AND GOOGLE ANALYTICS INTEGRATION

Where:

- **UA-999999-9** is your Google Analytics Tracking ID found under GA Admin/Web Property/Tracking Info/Tracking Code. The GA Property (UA#) should obviously be the same used in your Google Tag Manager configuration.
- **{{lead.ga.cid}}** is the required value of the GA client id
- **cd1** is the custom dimension used in GA to hold the Marketo Lead ID {{Lead.Marketeto Unique Code}}. Make sure you are using the right Custom Dimension as defined in step 2 above

① If your Google Analytics configuration do not currently use the User ID feature, you could use the Marketo Lead ID as your user id.

6. Modify the workflow

Under Marketo Marketing Activities/Interesting Moments/Fills out Form (or the workflow where you want to have end-to-end integration):

- Drag the Integration/Call Webhook item in your workflow
- Pick the “GA: received” hook

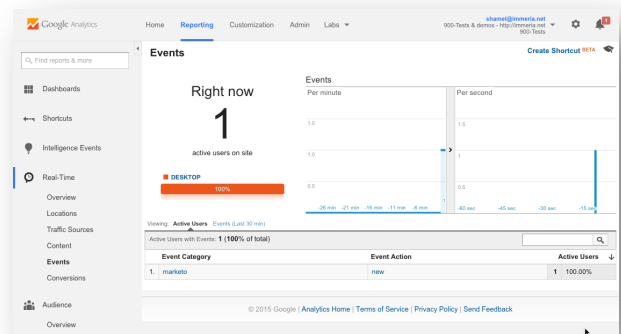
7. Validate if it works

Under Marketo/Marketing Activities, for the workflow you just modified, look under Results and you should see an entry for “Activity Type/Call Webhook” with the Detail of “sent GA: received”. If you click on the ID you will see the exact GA

Measurement Protocol request that was sent.

ID	Date/Time EDT	Activity Type	Det.	Lead Name	Email	Company
72	20-Jul-2015 4:01 PM	Call Webhook	Sen...	test test	shamel@mmmeria...	
71	20-Jul-2015 4:01 PM	Interesting Moment	We...	test test	shamel@mmmeria...	
67	20-Jul-2015 3:56 PM	Call Webhook	Sen...	test test	shamel@mmmeria...	
66	20-Jul-2015 3:56 PM	Interesting Moment	We...	test test	shamel@mmmeria...	
60	20-Jul-2015 3:54 PM	Call Webhook	Sen...	test test	shamel@mmmeria...	
59	20-Jul-2015 3:54 PM	Interesting Moment	We...	test test	shamel@mmmeria...	
53	20-Jul-2015 2:53 PM	Call Webhook	Sen...	test test	shamel@mmmeria...	
52	20-Jul-2015 2:53 PM	Interesting Moment	We...	test test	shamel@mmmeria...	
47	20-Jul-2015 2:51 PM	Call Webhook	"GA...	test test	shamel@mmmeria...	
46	20-Jul-2015 2:51 PM	Interesting Moment	We...	test test	shamel@mmmeria...	
42	20-Jul-2015 2:49 PM	Call Webhook	"GA...	test test	shamel@mmmeria...	
41	20-Jul-2015 2:49 PM	Interesting Moment	We...	test test	shamel@mmmeria...	

Under Google Analytics, you can use the Real Time Events report:



① If you use this approach, you could forgo the onSuccess() handler, so only the successful reception of the lead on the Marketo side is tracked.

Marketo leads via REST API

Leads don't have to be created with Marketo forms. For example, you could use server-side processing to generate a new lead with the help of the Marketo API. In this case, all you have to do is add the hidden token to the list of input fields sent along with your form submit.

1. Modify your web form

Given a “normal” HTML form, you want to add a hidden field and set its value to the GA Client ID.

```
<form id="signup-form" action="/blog/email-signup2.php"
method="post">
<input type="email">
<input type="hidden" name="gacid">
<input type="submit" value="Sign me up!">
</form>
<script>
// Get the value of the GA client id
function getgacid() {
```

```
try {
  var tracker = ga.getAll()[0];
  return tracker.get('clientId');
} catch (e) {
  return 'n/a';
}
// and set the value of our hidden field
$('#[name=gacid]').val(getgacid());
$('#signup-form').submit(function(e) {
  datalayer.push({event: 'mktoLead',mktoFormId: 'signup-form'});
  //... your code
});
</script>
```

2. Modify the server-side handler

Refer to the Marketo REST API for details.

For example, using PHP, you need to create a JSON Request that looks like this:

```
{
  "action": "createOrUpdate",
  "lookupField": "email",
  "input": [
    {
      "email": "<?php $_POST['email']; ?>",
      "gacid": "<?php $_POST['gacid']; ?>"
    }
  ]
}
```

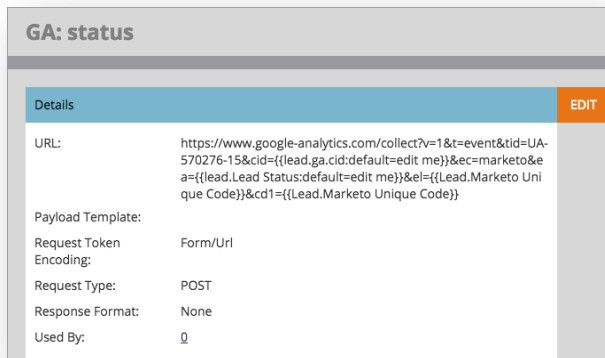
(code provided as example only - not exact/complete)

Extend the concept to Marketo Lead workflows

The concept explained above can be extended to fire Google Analytics events when the status of a lead is modified (ex. lead rejected), send an ecommerce transaction when a lead is closed, or any other important moments in your workflow.

Lead Status is Changed

For example, the following Webhook could be used whenever the Lead Status is changed:



The URL is as follow:

```
https://www.google-
analytics.com/collect?v=1&t=event&tid=UA-999999-
9&cid={{lead.ga.cid:default=edit
me}}&ec=marketo&ea={{lead.Lead
Status:default=rejected}}&el={{Lead.Marketo Unique
Code}}&cd1={{Lead.Marketo Unique Code}}
```

A Google Analytics event will be registered using the Measurement Protocol with event Category: "marketo", event Action: {{lead.Lead Status}}, event Label: the unique ID of this lead. The other fields are the same as previously described.

Closing a lead

If you want to send a transaction to GA when the deal is closed, you will need to refer to the [Measurement Protocol for Enhanced Ecommerce](#), and fill each of the required info with the Lead data.

Additional References

- Marketo Developers: <http://developers.marketo.com/>
- Google Tag Manager for Web Tracking: <https://developers.google.com/tag-manager/devguide>
- Google Analytics Measurement Protocol: <https://developers.google.com/analytics/devguides/collection/protocol/v1/>