

EXT: tollwerk Google Analytics

Extension Key: tw_gooleanalytics

Language: en

Version: 1.0.0

Keywords: google, analytics, tracking, forAdministrators, forDevelopers, forIntermediates, forAdvanced

Copyright © 2012 Dipl.-Ing. Joschi Kuphal, <joschi@tollwerk.de>

This document is published under the Open Content License available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3 – a GNU/GPL CMS/Framework available from www.typo3.org

In case you are reading this manual online at the TYPO3 website, we strongly recommend that you also visit the [TYPO3 Extensions & Manuals page](#) respectively the page about the [tollwerk Google Analytics TYPO3-Extension](#) at our own website. We provide a PDF version of this manual there, which probably renders more nicely than the online version on typo3.org. (Sorry for our website currently being available in German language only. However, the PDF extension manuals are in English of course.)

You can also visit the extension's [TYPO3 Forge project site](#) for latest news, feature requests and issue tracking.

Table of Contents

EXT: tollwerk Google Analytics.....	1		
Introduction.....	3		
What does it do?.....	3		
Screenshots.....	3		
Installation.....	4		
Google Analytics.....	4		
Extension setup.....	6		
Configuration.....	7		
		Constants.....	7
		Setup.....	9
		Usage & advanced techniques.....	15
		Disable tracking for individual pages.....	15
		Using custom variables.....	15
		Event tracking & JavaScript API.....	17
		Known problems.....	19
		To-Do list.....	20
		ChangeLog.....	21

Introduction

What does it do?

The extension provides a frontend plugin for the integration and usage of Google Analytics tracking with your TYPO3 websites. It is built on the extbase / fluid architecture and supports the asynchronous method of embedding the tracking code. Besides the plain tracking of page views the plugin leverages many of the more advanced Google Analytics API methods including:

- Custom variables (feat. database lookups)
- Event tracking
- Sub-domain, multi-domain and cross-domain tracking
- Tracking of external URLs, downloads and e-mail links
- Social interaction tracking
- Registration of direct access keywords and referrers
- Registration of additional search engines
- IP address anonymization (required e.g. by German law)
- and more ...

As the tracking code has no visual appearance on a website, the extension doesn't provide a content element you could visually put onto a page. Instead the tracking code has to be included via TypoScript. Most of the plugin options can be easily configured via the constant editor. The main JavaScript code is rendered by a fluid template, so you can customize it to your needs if necessary.

To learn more about Google Analytics please see <http://google.com/analytics>.

Screenshots

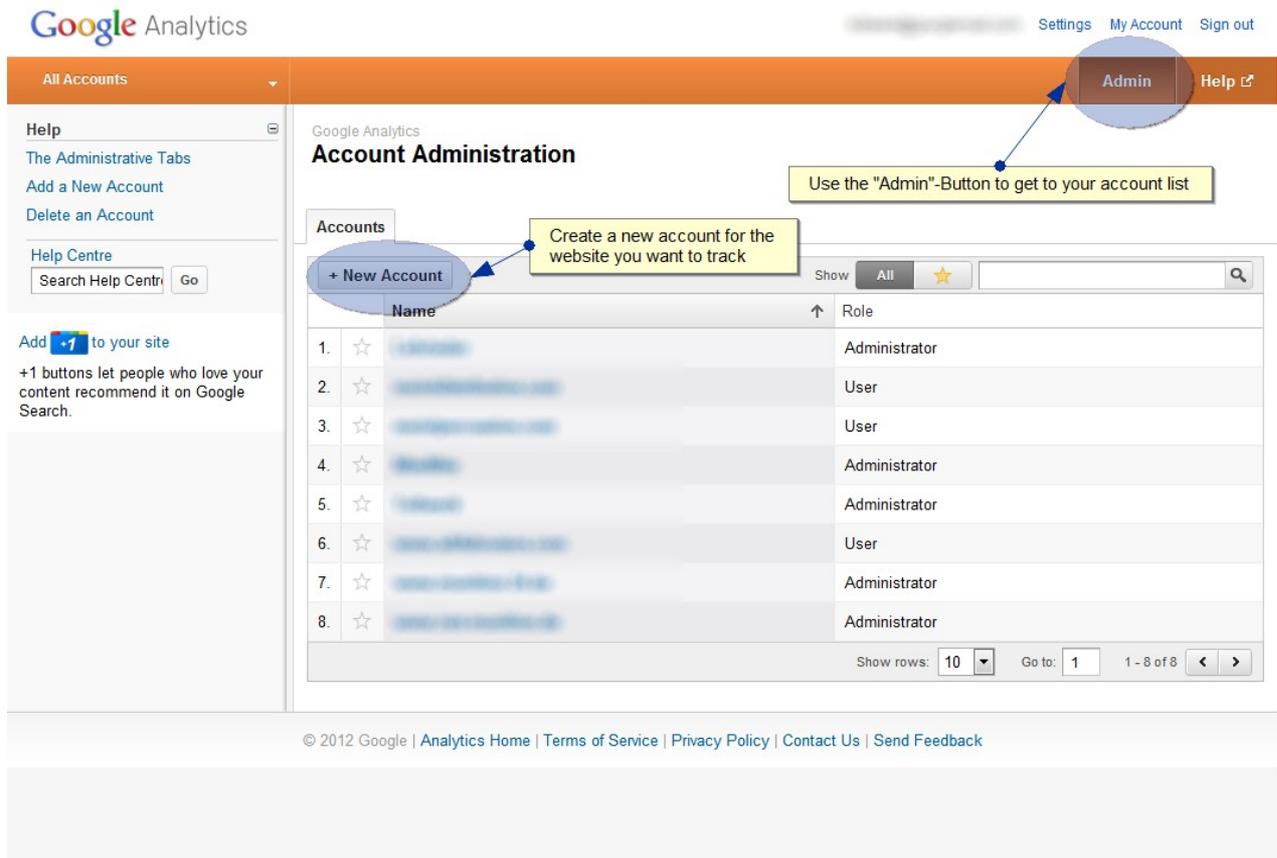
The plugin doesn't have any visible output, so there is nothing to take a screenshot of.

Installation

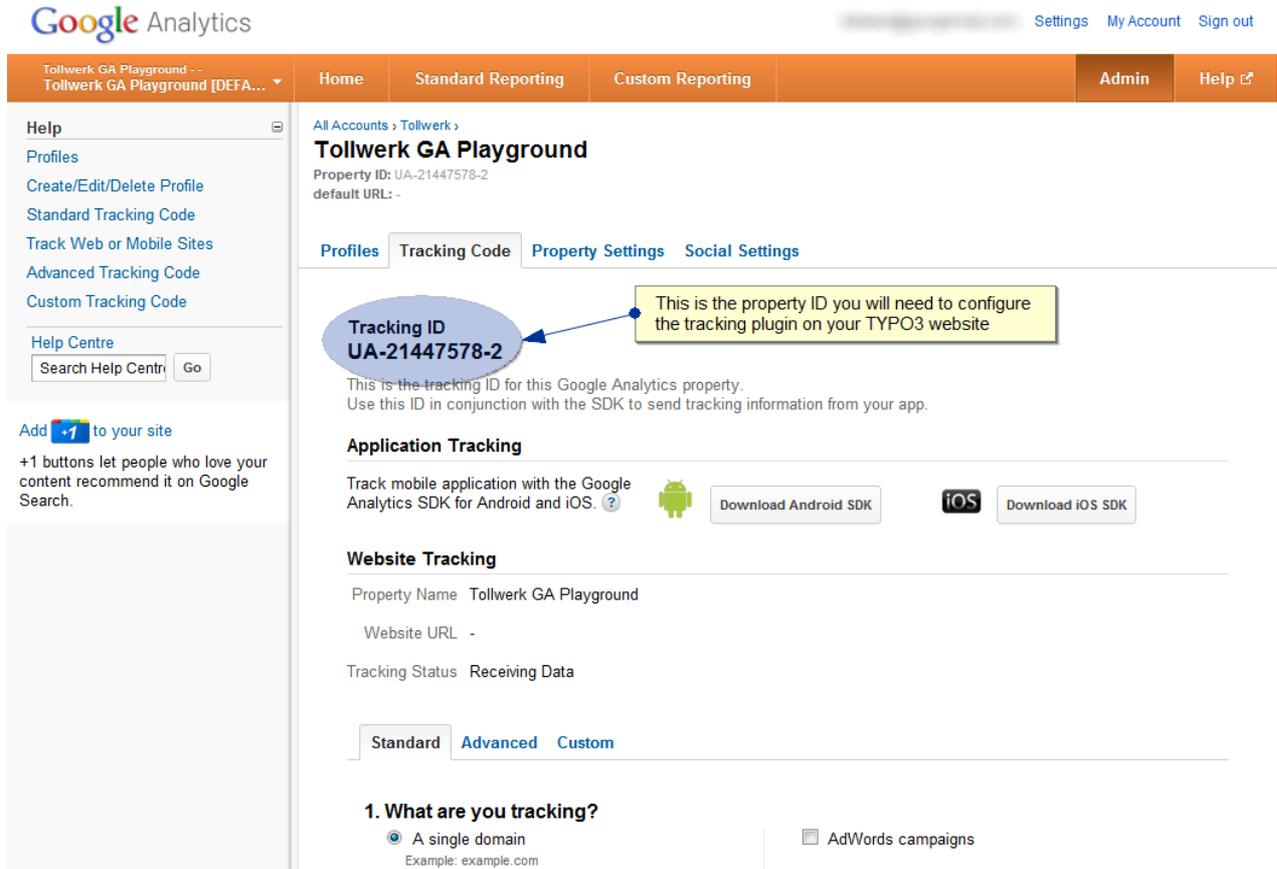
To get your website(s) tracked by Google Analytics you will first of all have to prepare an appropriate Google Analytics account. Afterwards, you will have to install and configure this extension in your website's TYPO3 backend.

Google Analytics

Start by registering with Google Analytics at <http://www.google.com/analytics> if you haven't done so before. Google Analytics let's you track an arbitrary number of so called "properties". In our case the **property** we want to track is a website. Any property has to belong to what Google calls an **account**, so go on and create one. During account creation you will be asked for the URL of your website, which will automatically be created as a first property inside the new account (alternatively you could of course just create a new property for your website within your already existing account).



After setup has finished, you will be able to see the details of your newly created property. Find and remember the so called property ID for your website – this is what you’ll need to configure the TYPO3 extension.



Note: You don't need the JavaScript tracking code that is provided by Google here, just grab the property ID. The TYPO3 plugin will take care of the necessary JavaScript and provide an extended tracking object.

Hint for Webmaster Tools users

In case you plan to link your Google Analytics property to a verified website in Google Webmaster Tools (<https://www.google.com/webmasters/tools>), please be aware of the fact that the TYPO3 plugin uses it's own integration source code. In contrast to the original Google Analytics tracking code the plugin's one will not work for website verification and thus will not be recognized by the Webmaster Tools. You will have to use one of the alternative methods to verify your website (e.g. HTML file upload or meta tag based verification).

Extension setup

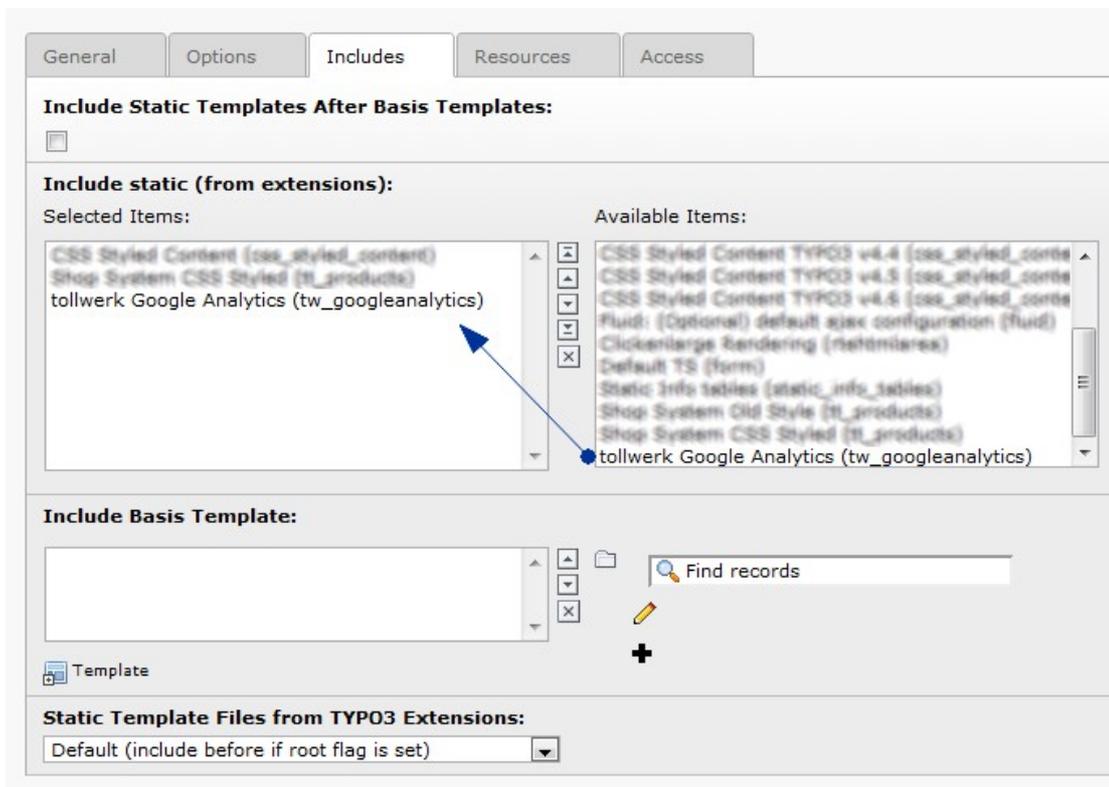
Follow these steps to get the Google Analytics tracking up and running on your website:

1. Extension installation

Download the extension from the TYPO3 extension repository and enable it. The extension key is `tw_googleanalytics`. There is nothing to configure via the extension manager.

2. Including the static TypoScript

Include the plugin's static TypoScript into the TypoScript root template of your site.



Note: The plugin's static TypoScript assumes that your main page object is called “**page**”. If this is not the case, please don't include this file. Instead just copy it's contents (EXT:tw_googleanalytics/Configuration/TypoScript/setup.txt) into a custom TypoScript template that is part of your setup and follow the comment in the second half of the template.

3. Including the Plugin

Finally you have to include the tracking code plugin to your website's output. This is done via TypoScript. Please put something like the following into one of your TypoScript templates:

```
# GOOGLE ANALYTICS TRACKING CODE PLUGIN
page.headerData.9999 =< plugin.tx_twgoogleanalytics
```

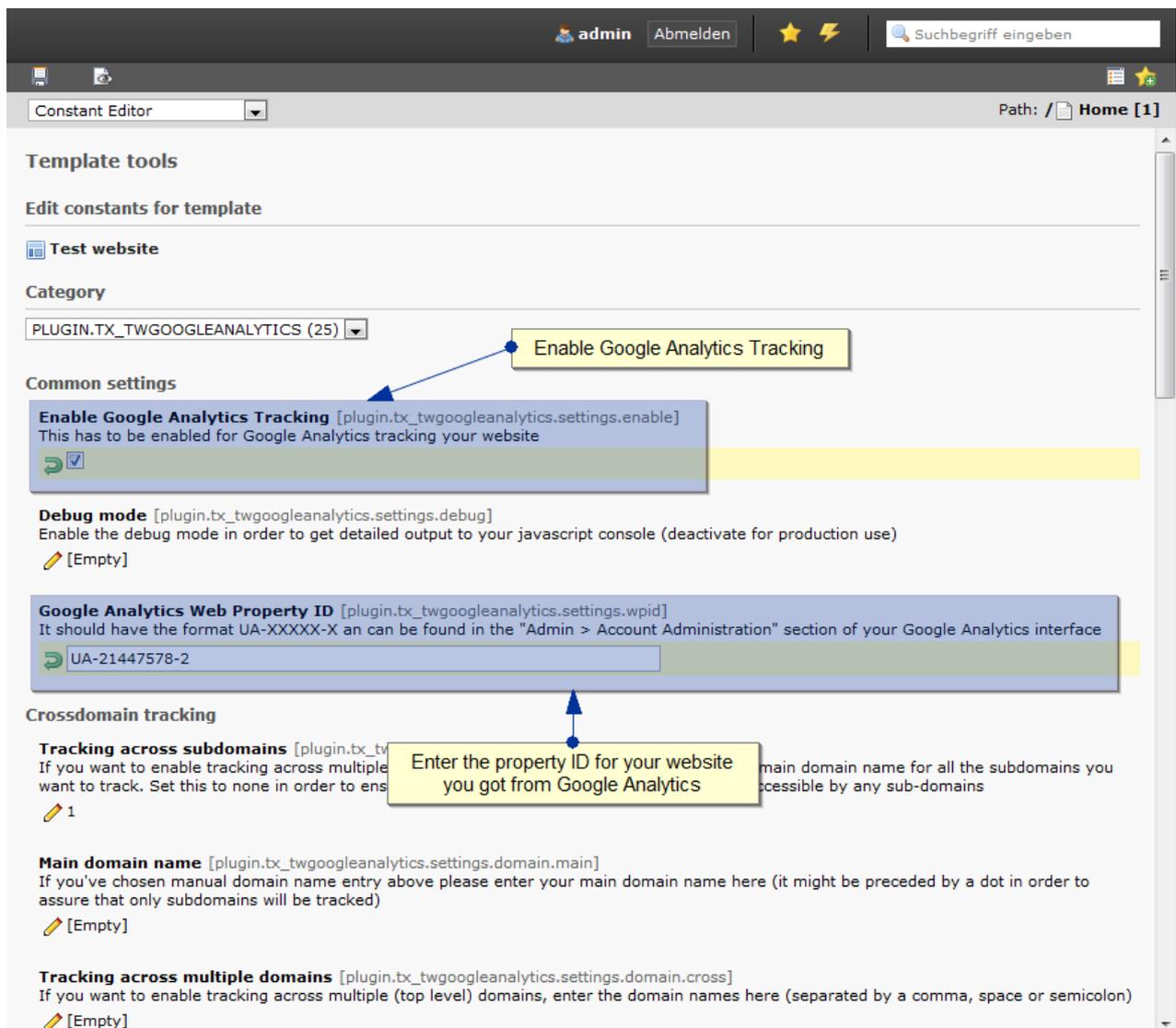
Note: Again, the example above assumes your main page object to be called “page” – change this appropriately if necessary. Also, the tracking code should be rendered as close as possible to the closing `</head>`-element of your HTML source code. In case you already used the headerData-index 9999 (or higher) for something different, you may want to alter the slot into which to insert the plugin.

Configuration

Most of the plugin's features can easily be configured via constants (i.e. the constant editor). Some more advanced features like custom variables currently have to be defined via manually crafted TypoScript though – please see the chapter Usage & advanced techniques for those techniques.

Constants

You can use the constant editor (e.g. on your TypoScript root template) to control the majority of the plugin's features. At the minimum you will have to **enable the tracking in general** and enter the **Google Analytics property ID** for your website in order to make the plugin do something reasonable (the default property ID will result in the plugin sending it's data to one of the author's test accounts, so be sure to put yours one in here).



Among the available options you will find means to

- enable the **debug mode**,
- enable **IP address anonymization**,
- configure **sub-domain / multi-domain / cross-domain tracking**,
- configure **download, e-mail and external URL tracking**,
- refine **direct traffic keywords / referrers and search engine recognition**
- and much more.

Almost any constant is represented as a setup property as well ([see below](#)), and the constants' descriptions in the editor should be rather meaningful, so there is no reason in repeating them here.

Debug mode

If you enable the **debug mode**, the plugin will write some information to the JavaScript console of your browser (using the `console.xxx()` methods). This should work in most modern browsers. You can see which tracking commands your website is effectively sending to Google Analytics. If you set the debug mode to “**Debug only**”, no real data transmission to Google Analytics will occur – the information is just printed to the console. It is recommended to disable the debug mode in production use.

Setup

The following are the plugin's TypoScript settings. Using the default configuration, all but two of them (see *customVariables* and *pageUrl* at the end of the table) are controlled by a corresponding constant you can set via the constant editor . Of course you can override any setting at any time via TypoScript ...

Property:	Data type:	Description:	Default:
view.templateRootPath	dir	Root path for the fluid templates of the plugin. The plugin has just one controller (named "Googleanalytics") with one action (named "track"). Therefore the only template definitely needed for rendering is located at <i>Googleanalytics/Track.html</i> relative to the template root path.	{\$plugin.tx_twgoogleanalytics.view.templateRootPath}
view.partialRootPath	dir	Root path for the fluid partials of the plugin (not used by default)	{\$plugin.tx_twgoogleanalytics.view.partialRootPath}
view.layoutRootPath	dir	Root path for the fluid layouts of the plugin (not used by default)	{\$plugin.tx_twgoogleanalytics.view.layoutRootPath}
settings.debug	int	Debug mode for printing tracking commands to the JavaScript console of your web browser Options: <ul style="list-style-type: none"> • 0: Disabled • 1: Enabled • 2: Enabled in "Debug only" mode (real data transmission to Google Analytics will be suspended; useful for testing) 	{\$plugin.tx_twgoogleanalytics.settings.debug}
settings.wpid	string	Web property ID (in the format UA-XXXXX-X) which defines to which Google Analytics account tracking data will be submitted. Defaults to the property ID "UA-21447578-2", which belongs to a tollwerk test account.	{\$plugin.tx_twgoogleanalytics.settings.wpid}
settings.features.anonymizeIP	boolean	With this feature enabled the IP addresses of your visitors are anonymized by discarding the last octet (required e.g. by German law). It is disabled by default.	{\$plugin.tx_twgoogleanalytics.settings.anonymizeIP}
settings.features.track.title	boolean	Enables transmission of page titles (enabled by default)	{\$plugin.tx_twgoogleanalytics.settings.track.title}
settings.features.track.client	boolean	Enables transmission of web browser details (enabled by default)	{\$plugin.tx_twgoogleanalytics.settings.track.client}
settings.features.track.flash	boolean	Enables detection and transmission of flash player details (enabled by default)	{\$plugin.tx_twgoogleanalytics.settings.track.flash}

Property:	Data type:	Description:	Default:
settings.crossdomain.sub	int	<p>Controls the tracking behaviour across sub-domains. The setting defaults to 1 (“auto”). You don't have to change this as long as you only track one single domain. If you want to track a single domain and / or it's sub-domains, or if you want to track across several top level domains (“multi-domain tracking”), you'll have to change this to 2 (“manual”) and set the main domain name (see the <i>crossdomain.main</i> setting). If you explicitly want to ensure that sub-domains are not tracked, you should set this to 0 (“none”).</p> <p>Options:</p> <ul style="list-style-type: none"> 0 (“none”): Ensures that the cookies in a given domain are not accessible by any sub-domains 1 (“auto”): Suitable for tracking one single domain 2 (“manual”): Required for tracking of sub-domains or across multiple top level domains (use in conjunction with the <i>crossdomain.main</i> setting) 	{\$plugin.tx_twgoogleanalytics.settings.domain.sub}
settings.crossdomain.main	string	<p>Sets the main domain name. You need to do this if you set <i>crossdomain.sub</i> to something else than 1 (“auto”).</p> <p>Example: <code>crossdomain.main = example.com</code></p> <p>If you want to ensure, that only the sub-domains of your main domain are tracked, prepend the domain name with a dot.</p> <p>Example: <code>crossdomain.main = .example.com</code></p> <p>This setting defaults to an empty string.</p>	{\$plugin.tx_twgoogleanalytics.settings.domain.main}
settings.crossdomain.cross	string	<p>Controls cross-domain tracking. Enter a comma-separated list of external domains (i.e. not sub-domains of your main domain!) you want to track across. For further information about multiple domain tracking please visit https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingSite.</p> <p>Example: <code>crossdomain.main = first-domain.com, another.second-domain.com</code></p> <p>This setting defaults to an empty string.</p>	{\$plugin.tx_twgoogleanalytics.settings.domain.cross}
settings.external.track	int	<p>Enables tracking of clicks on links to external domains. You can choose if those clicks are submitted as “pageView” or as “event” records.</p> <p>Options:</p> <ul style="list-style-type: none"> 0: Disabled (default) 1: Enabled, tracking as “pageView” 2: Enabled, tracking as “event” <p>Note: In contrast to cross-domain tracking (see <i>crossdomain.cross</i>) the tracking cookie will not be passed to the external domains. Just the clicks on links to them will be registered.</p>	{\$plugin.tx_twgoogleanalytics.settings.external.track}

Property:	Data type:	Description:	Default:
settings.external.prefix	string	<p>Sets a prefix for external domain tracking. Depending on your <i>external.track</i> setting the prefix will be used</p> <ul style="list-style-type: none"> as URL prefix, in case you are tracking external domains as “pageViews” (Example: “/External:http://external-domain.com/path/to/some/page.html”), or as event category, in case you are tracking external domains as “events” (see Event tracking for further information). <p>This setting defaults to “External”.</p>	{ \$plugin.tx_twgoogleanalytics.settings.external.prefix }
settings.external.restrict	string	<p>Restricts external domain tracking to the specified domains. Enter a comma-separated list of domains.</p> <p>This setting defaults to an empty string.</p>	{ \$plugin.tx_twgoogleanalytics.settings.external.restrict }
settings.email.track	int	<p>Enables tracking of clicks on e-mail links. You can choose if those clicks are submitted as “pageView” or as “event” records.</p> <p>Options:</p> <ul style="list-style-type: none"> 0: Disabled (default) 1: Enabled, tracking as “pageView” 2: Enabled, tracking as “event” 	{ \$plugin.tx_twgoogleanalytics.settings.email.track }
settings.email.prefix	string	<p>Sets a prefix for email link tracking. Depending on your <i>email.track</i> setting the prefix will be used</p> <ul style="list-style-type: none"> as URL prefix, in case you are tracking e-mail links as “pageViews” (Example: “/Email:test@example.com”), or as event category, in case you are tracking e-mail links as “events” (see Event tracking for further information). <p>This setting defaults to “Email”.</p>	{ \$plugin.tx_twgoogleanalytics.settings.email.prefix }
settings.email.restrict	string	<p>Restricts e-mail link tracking to specified e-mail addresses. Enter a comma-separated list of e-mail addresses.</p> <p>This setting defaults to an empty string.</p>	{ \$plugin.tx_twgoogleanalytics.settings.email.restrict }
settings.download.track	int	<p>Enables tracking of clicks on download links. You can choose if those clicks are submitted as “pageView” or as “event” records.</p> <p>Options:</p> <ul style="list-style-type: none"> 0: Disabled (default) 1: Enabled, tracking as “pageView” 2: Enabled, tracking as “event” 	{ \$plugin.tx_twgoogleanalytics.settings.download.track }
settings.download.prefix	string	<p>Sets a prefix for download link tracking. Depending on your <i>download.track</i> setting the prefix will be used</p> <ul style="list-style-type: none"> as URL prefix, in case you are tracking e-mail links as “pageViews” (Example: “/Download:/path/to/file.pdf”), or as event category, in case you are tracking e-mail links as “events” (see Event tracking for further information). <p>This setting defaults to “Download”.</p>	{ \$plugin.tx_twgoogleanalytics.settings.download.prefix }

Property:	Data type:	Description:	Default:
settings.download.template	string	<p>As of version 0.9.5 it is possible to provide a substitution template string for download link tracking. The template string is used for composing the download URL sent to Google Analytics and may contain some markers that will be substituted with the real link values. The markers mainly represent the properties of the native JavaScript "location" object (see http://www.w3schools.com/jsref/obj_location.asp) plus the two additional markers "{filename}" and "{extension}":</p> <ul style="list-style-type: none"> • {hash}: The anchor portion of a URL (including the "#" sign, e.g. "#section-1") • {host}: The hostname and port of a URL (e.g. "example.com:80") • {hostname}: The hostname of a URL (e.g. "example.com") • {href}: The entire URL (e.g. "http://example.com:80/path/to/file.pdf?foo=bar#section-1") • {pathname}: The path name of a URL including the leading slash (e.g. "/path/to/file.pdf") • {filename}: The file name of a URL (e.g. "file.pdf") • {extension}: The file extension of a URL (e.g. "pdf") • {port}: The port number the server uses for a URL (e.g. 80) • {protocol}: The protocol of a URL (e.g. "http:") • {search}: The query portion of a URL including the leading question mark (e.g. "?foo=bar") <p>Any other component of the template string will remain untouched and literally be sent to Google Analytics (after being prefixed with the <i>settings.download.prefix</i>).</p> <p>This setting defaults to "{pathname}".</p>	
settings.download.list	string	<p>Restricts download link tracking to specified directories and / or file extensions. Enter a comma-separated list of directory-extension-combinations, where each element has the format</p> <pre>path = space separated extension list</pre> <p>If this setting is given, only download links matching at least one specified directory-extension-combination will be tracked. A download URL will be tracked,</p> <ul style="list-style-type: none"> • if it's local path (relative to your site root) starts with the given <i>path</i> component, and • if the file extension is contained in the list of given <i>extensions</i>. <p>Examples:</p> <pre>download.list = fileadmin/tutorials=pdf doc txt download.list = fileadmin=pdf, public/downloads=zip exe</pre> <p>As a special value for the <i>path</i> component you can use "/" for recursively matching any directory of your website. As a special value for the <i>extension</i> component you can use "*" for matching any file extension. Having said that it should be obvious that</p> <pre>download.list = /*</pre> <p>effectively is a full wildcard (in which case it should be omitted as it doesn't result in any restriction of download URLs).</p> <p>This setting defaults to "/=doc docx pdf xls ppt zip gz bz2 rar txt vsd vxd js css exe wma mov avi wmv mp3 mp4".</p>	{\$plugin.tx_twgoogleanalytics.settings.download.list}

Property:	Data type:	Description:	Default:
settings.direct.keywords	string	<p>Registers keywords that should be treated as direct access to your website. Enter a comma-separated list of keywords.</p> <p>Traffic from (known) search engines and directories to your website will normally be registered as “organic traffic” by Google Analytics. But if a visitor just searched for your company name before entering your website, it might be more appropriate to register this kind of traffic as direct access (comparable with someone who simply entered your domain name into his browser address bar) instead of as traffic based on a search request. Defining your company name a direct access keyword will accomplish this task.</p> <p>This setting defaults to an empty string.</p>	{ <code>\$plugin.tx_twgoogleanalytics.settings.direct.keywords</code> }
settings.direct.referrers	string	<p>Registers referrers that should be treated as direct access to your website. Enter a comma-separated list of domains.</p> <p>Traffic from external domains will generally be registered as “referral traffic” by Google Analytics. In case you are running several websites under different domains linking to each other, you might want to register incoming inter-domain traffic rather as direct access than as referral traffic.</p> <p>Example: <code>direct.referrers = my-sister-site.com</code></p> <p>This setting defaults to an empty string.</p>	{ <code>\$plugin.tx_twgoogleanalytics.settings.direct.referrers</code> }
settings.searchengines	string	<p>Registers additional search engines and their appropriate query variable names. Enter a comma-separated list of search engine / query variable definitions.</p> <p>Google Analytics internally uses a pretty comprehensive list of search engine signatures in order to identify “organic traffic” (see https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingTraffic#searchEngine). However, it might be necessary to add further search engines to this list. You can do so by listing search engine definitions in the following format: <code>search engine name = query variable name</code></p> <p>The specified <i>search engine name</i> has to be part of the search engine's domain name for Google Analytics being able to identify your custom search engine. Furthermore, Google Analytics will expect a query variable named <i>query variable name</i> to be present in the referring URL.</p> <p>Example: <code>searchengines = MySpecialSearchEngine=query, AnotherSE=search</code></p> <p>In order to add a search engine definition to the beginning of the recognition list, append an exclamation mark to the query term: <code>searchengines = MostImportantSearchEngine=q!</code></p> <p>This setting defaults to an empty string.</p>	{ <code>\$plugin.tx_twgoogleanalytics.settings.searchengines</code> }

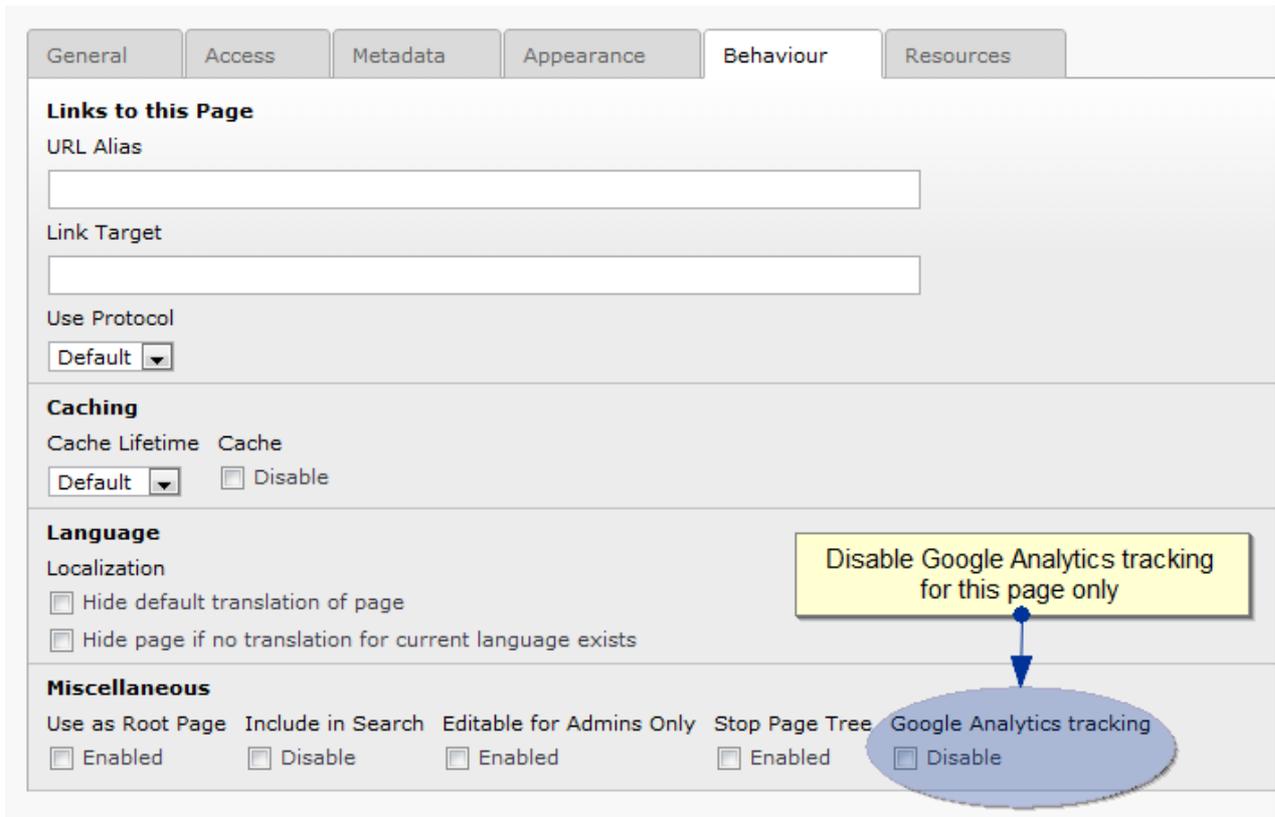
Property:	Data type:	Description:	Default:
settings.customVariables	array	Registers custom variables for transmissions to Google Analytics. Custom variables are name-value pair tags that you can send along with “pageViews” and “events” in order to refine Google Analytics tracking. For further information on custom variables please see https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingCustomVariables . This setting defaults to an empty array and can only be controlled via custom TypoScript. For instructions on how to set custom variables please read the dedicated chapter below.	
settings.customVariables.[1-5]	->customVar	See the definition of custom variables.	
settings.pageUrl	cObject	Sets the URL that should be tracked with a “pageView” when a frontend page is visited. This setting defaults to the current request URL, but in certain cases you might want to override this. Imagine e.g. a central page for rendering the single display of news items. Instead of tracking an individual URL for each and every news item (leading to pretty unusable results in Google Analytics) you could also track the news independent central page URL as “pageView” along with a custom variable indicating the title of the visited news item. This way you'll receive a clearly represented total of news item reads together with readable statistics for all your news items in Google Analytics. Effectively you just have to strip off the GET parameters of the current request URL ... Example: <pre>plugin.tx_twgoogleanalytics.settings.pageUrl { data > typolink { parameter.data = TSFE:id returnLast = url } }</pre>	<pre>pageUrl = TEXT pageUrl. data = getenv:REQUEST_URI</pre>

[tsref:plugin.tx_twgoogleanalytics]

Usage & advanced techniques

Disable tracking for individual pages

In case you don't want certain pages to be tracked by Google Analytics, you can use the new flag "Disable Google Analytics tracking" in the page record's properties dialog:



Using custom variables

Custom variables are name-value pair tags that you can send along with "pageViews" and "events" in order to refine Google Analytics tracking. For further information on custom variables please see <https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingCustomVariables>.

There may be **up to 5 custom variables** being sent to Google Analytics at the same time (i.e. with each page view or event). It is your part to define both the **variable name** and the **variable value** that gets submitted – Google Analytics doesn't impose any preferences here.

Custom variables have to be defined as TypoScript array. **Valid keys** for this array are the numbers **1 to 5** (including), representing the respective variable index. Each custom variable definition is an array itself, consisting of the following elements:

customVar

Property:	Data type:	Description:	Default:
name	string	Name of the custom variable. You can choose an arbitrary term here. This is what Google Analytics will refer to as "Custom variable key".	

Property:	Data type:	Description:	Default:
<p>value</p>	<p>string / array</p>	<p>The value of the custom variable. This is what Google Analytics will refer to as “Custom variable value”. Currently two value types are supported:</p> <p>String value If the variable value is given as a string, this string will be used as literal value.</p> <p>Example: <pre> plugin.tx_twgoogleanalytics.settings.customVariables.1 { name = Section value = Global level = page } </pre> </p> <p>Array value The value of a custom variable may be pulled from the current GET / POST variables. In this case <i>value</i> itself has to carry the literal value “GP” and must define a sub-array specifying the GET / POST variable you want to pull the value from. The sub-array must contain the index key 1, otherwise the value will not be evaluated.</p> <p>Example: <pre> plugin.tx_twgoogleanalytics.settings.customVariables.1 { name = News value = GP value { 1 = tx_ttnews[uid] } level = page } </pre> </p> <p>Furthermore, it is possible to run a database lookup based on one or more GET / POST variables. The keys of the specified GET / POST variables serve as replacement markers then (prepended with a dollar sign “\$”). Use the key <i>lookup</i> to define the SQL statement to be used for the database lookup. The first column of the first row of the query result will be used as custom variable value.</p> <p>Example: <pre> plugin.tx_twgoogleanalytics.settings.customVariables.1 { name = News value = GP value { 1 = tx_ttnews[uid] 2 = L 0 lookup = SELECT title FROM tt_news WHERE (uid=\$1 OR l10n_parent=\$1) AND sys_language_uid=\$2 } level = page } </pre> </p> <p>Please have a close look at the definition of the second GET / POST variable in this last example. The pipe character “ ” followed by the digit “0” indicates that “0” should to be taken as fallback value if no GET / POST variable named “L” is available. You can use this technique to specify fallback values for the variables you specify.</p> <p>Important: If any of the specified GET / POST does neither exist nor have a default value specified, then the whole custom variable is unresolvable and will not be sent to Google Analytics during this very page view.</p>	

Property:	Data type:	Description:	Default:
level	string	<p>The level of user engagement with your site, which effectively means the scope of the custom variable. Please see the official Google Analytics reference for more details.</p> <p>Options:</p> <ul style="list-style-type: none"> • visitor: Visitor-level custom variable • session: Session-level custom variable • page: Page-level custom variable 	

[tsref:plugin.tx_twgoogleanalytics.settings.customVariables.(customVariableIndex).customVar]

If you are using custom variables for your website you will most likely want to have different variable values on different pages. You could spread several TypoScript templates defining the appropriate variable values all over your site, or you could follow the author's recommendation and define all your values within one TypoScript template using conditions.

Example:

```
# GENERAL VARIABLE DEFINITION FOR ALL PAGES
plugin.tx_twgoogleanalytics.settings.customVariables {
    1 {
        name = Section
        value = Homepage
        level = page
    }
}

# NEWS AREA
[PIDinRootline = 5]
plugin.tx_twgoogleanalytics.settings.customVariables {
    1.value = News area
    2 {
        name = News
        value = GP
        value {
            1 = tx_ttnews[tt_news]
            lookup = SELECT title FROM tt_news WHERE $1 AND uid = $1
        }
        level = page
    }
}
[global]
```

Explanation:

In the latter example the custom variable 1 named “Section” is used to set an abstract name for the website area the visitor currently visits. It defaults to the literal value “Homepage”. Be aware that the variable definition is recursively effective for any sub-page of the page the above TypoScript is applied to (unless it is explicitly overridden), so there's a kind of custom variable inheritance all over your page tree. For the page with the ID 5 (and all of it's sub-pages as defined by the condition *PIDinRootline*) the value for the custom variable 1 will be set to the literal value “News area”, indicating that the visitor just visits the news section (respectively any of it's sub-pages, e.g. the news archive). Furthermore, a second custom variable 2 named “News” is introduced, holding the title of the currently displayed news item. This variable will not be sent to Google Analytics if there's no news displayed.

Event tracking & JavaScript API

JavaScript API

The plugin introduces a JavaScript object wrapping up and complementing the original Google Analytics tracking code. This wrapper object is globally available as JavaScript variable *tw_gat*, which automatically gets initialized on each page load and primarily receives all those parameters resulting from your constant settings.

Additionally there are two particular methods that have to be triggered intentionally if desired:

1. **trackEvent**: Use this method for event tracking (see next chapter).
2. **trackSocial**: Use this method for tracking the social engagement of your visitors (for further information please visit <https://developers.google.com/analytics/devguides/collection/gajs/gaTrackingSocial>)

In general the JavaScript library (EXT:tw_googleanalytics/Resources/Public/Js/tw_googleanalytics.js) should be sufficiently commented for understanding the available methods and their arguments, so please have a look there if you're interested. It's up to you to also use the other methods of the wrapper object explicitly, but you should know what you are doing then.

Event tracking

In contrast to page view tracking (i.e. recording which pages of your website are visited) event tracking is a method that you can use to record user interaction with single website elements. You could e.g. track particular UI elements like the buttons of a video player in order to learn more about your visitors' watching behaviour. For more information about event tracking please visit <https://developers.google.com/analytics/devguides/collection/gajs/eventTrackerGuide>.

By definition, event tracking is based on explicitly emitting JavaScript calls on certain user interactions. The plugin's wrapper object offers the method **trackEvent** for this purpose. The original JavaScript method looks like this (excerpt):

```
/**
 * Tracking an event
 *
 * For a detailed explanation of the available arguments please
 * @see https://developers.google.com/analytics/devguides/collection/gajs/eventTrackerGuide
 *
 * @param {String} category      Event category
 * @param {String} action        Event action
 * @param {String} label         Optional: Event label
 * @param {Number} value         Optional: Event value
 * @param {Boolean} nonInteraction Optional: Don't consider as interaction for bounce
rate calculation
 * @return {Object}              Self reference (liquid interface)
 */
tw_gat.trackEvent = function(category, action, label, value, nonInteraction) {
    ...
    return this;
}
```

As stated above you can find plenty of information about the meaning of the event tracking arguments in the [original Google Analytics guide](#). A video player button could e.g. be armed like this (which is not an example for unobtrusive JavaScript, truly, but just for the sake of brevity):

```
<input type="button" value="Play" onclick="videoplayer.play(); tw_gat.trackEvent('Video', 'Click', 'Play');"/>
```

Note: The custom variables you may have defined will be sent along with your event tracking commands.

Known problems

Currently there are no known problems or bugs. However, some of the features have been implemented without thorough testing, as they have not been needed in production use so far. Also, it might be possible that some errors introduced themselves while preparing this extension for public release (in an effort of over-correcting everything ...). If you experience any problem please don't hesitate to [contact the author](#). Thank you!

To-Do list

The following features could be part of future versions:

- Content campaign tracking
- E-Commerce tracking
- Setting custom variables via the page properties form (like metadata)

ChangeLog

Version:	Changes:
1.0.0	Switched extension status to stable; Added further German localization labels; Created TYPO3 Forge project for the extension; Minor documentation changes; Fixed a bug where the click handlers got installed although the ga.js could not be loaded in time (thanks to Kay Großweischede / Medien Partner for the bug report & help)
0.9.5	Fixed a bug with download / external link tracking (JavaScript errors when download / external link tracking was disabled) and a redundant registration of click handlers; Introduced a substitution template string for download link tracking; Fixed some minor documentation errors
0.9.4	Fixed a bug with download link tracking in some special situations (JavaScript error when clicking outside a link)
0.9.3	Added German and French language labels, fixed a bug with download link tracking (relative URLs) and updated the manual
0.9.2	Fixed a bug with download link tracking when there were images wrapped inside the links
0.9.1	Backported a variant of the “format.raw” fluid view helper (introduced with TYPO3 4.6) so that the extension also works with TYPO3 4.5 LTS now
0.9.0	Initial release