Extension Key:
Language:
Version: 4.0.0
Keywords:

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 Github repository for latest news, feature requests and issue tracking,

Table of Contents

Introduction

What does it do?

The extension provides a frontend plugin for the integration and usage of Universal Analytics 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 Universal Analytics API methods including:

- Custom variables / dimensions / metrics (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
- Remarketing & advertising reporting features
- Registration of direct access keywords and referrers
- Registration of additional search engines
- IP address anonymization (required e.g. by German law) and tracking disabling (opt-out)
- and more ...

As the tracking code has no visual appearance on a website, the extension dœsn'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

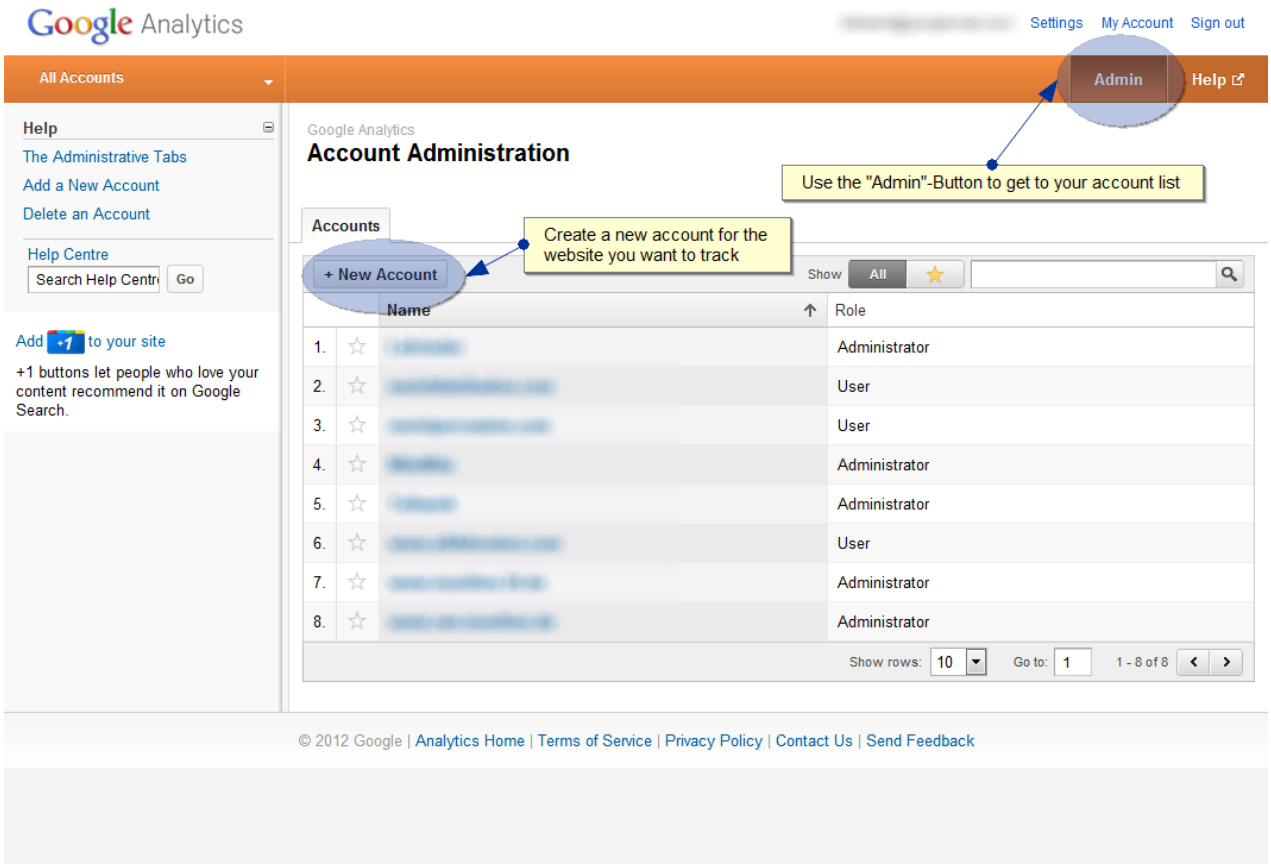
The plugin dœsn't have any visible output, so there is nothing to take a screenshot of.

Installation

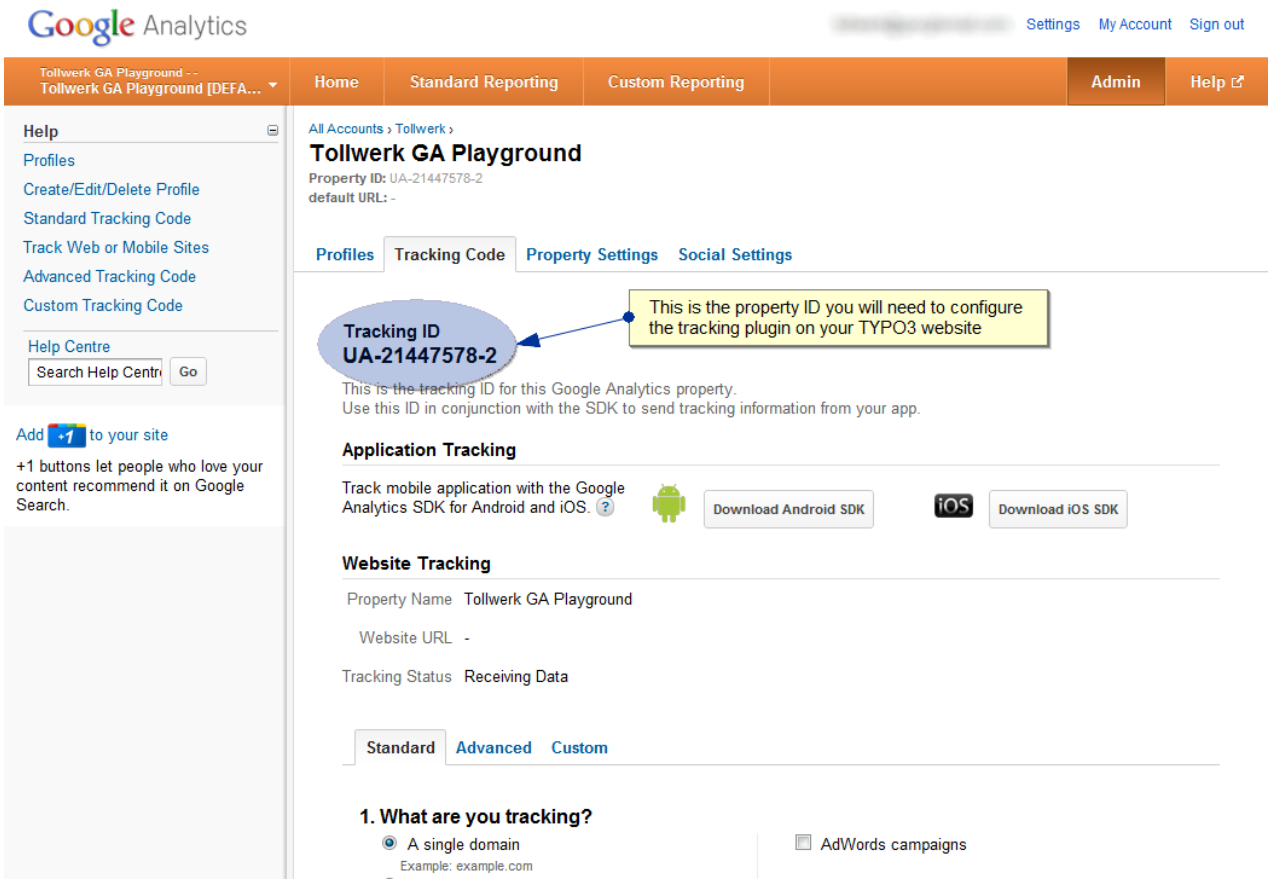
To get your website(s) tracked by Google Analytics you will first of all have to prepare an appropriate Google Analytics account. Afterwards, you will have to install and configure this extension in your website's TYPO3 backend.

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.

Universal Analytics

In March 2013 Google has introduced **Universal Analytics (UA)**, which uses different tracking methods (analytics.js). With version 4.0 of this extension, support for the old Google Analytics has been removed, making Universal Analytics the only option. The screenshots above show the traditional Google Analytics and may differ when for Universal Analytics.

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

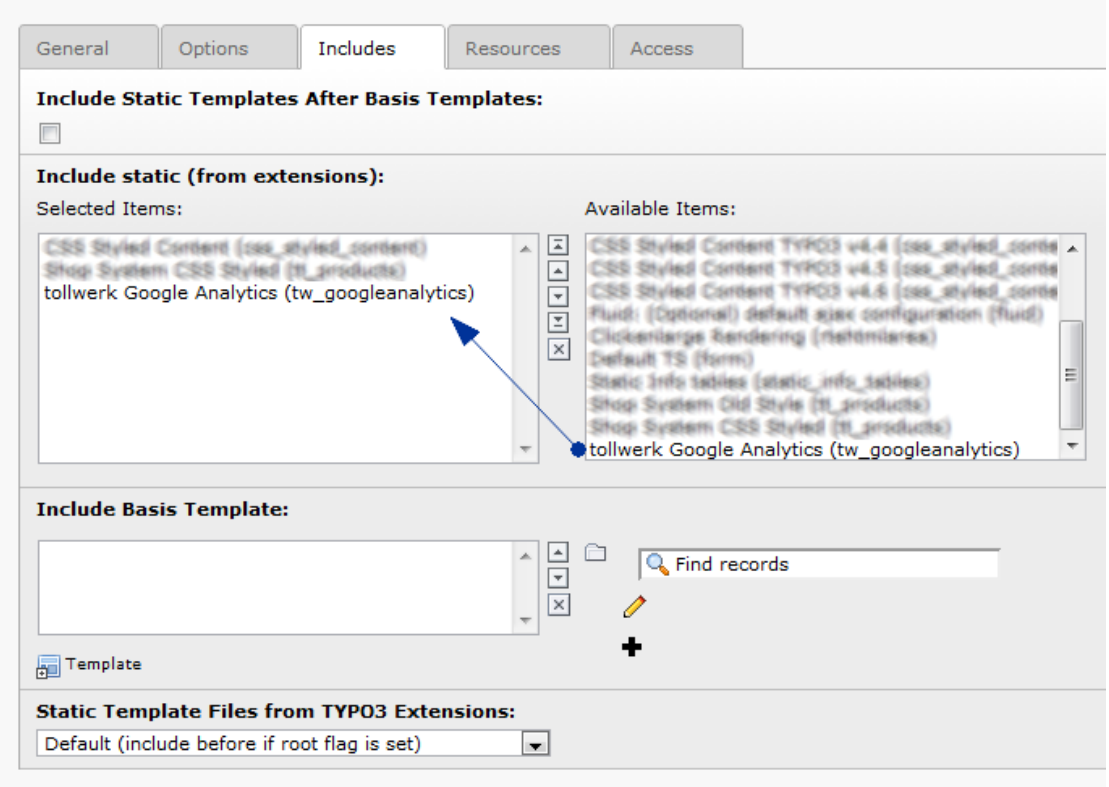
Follow these steps to get the Google Analytics tracking up and running on your website:

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
```

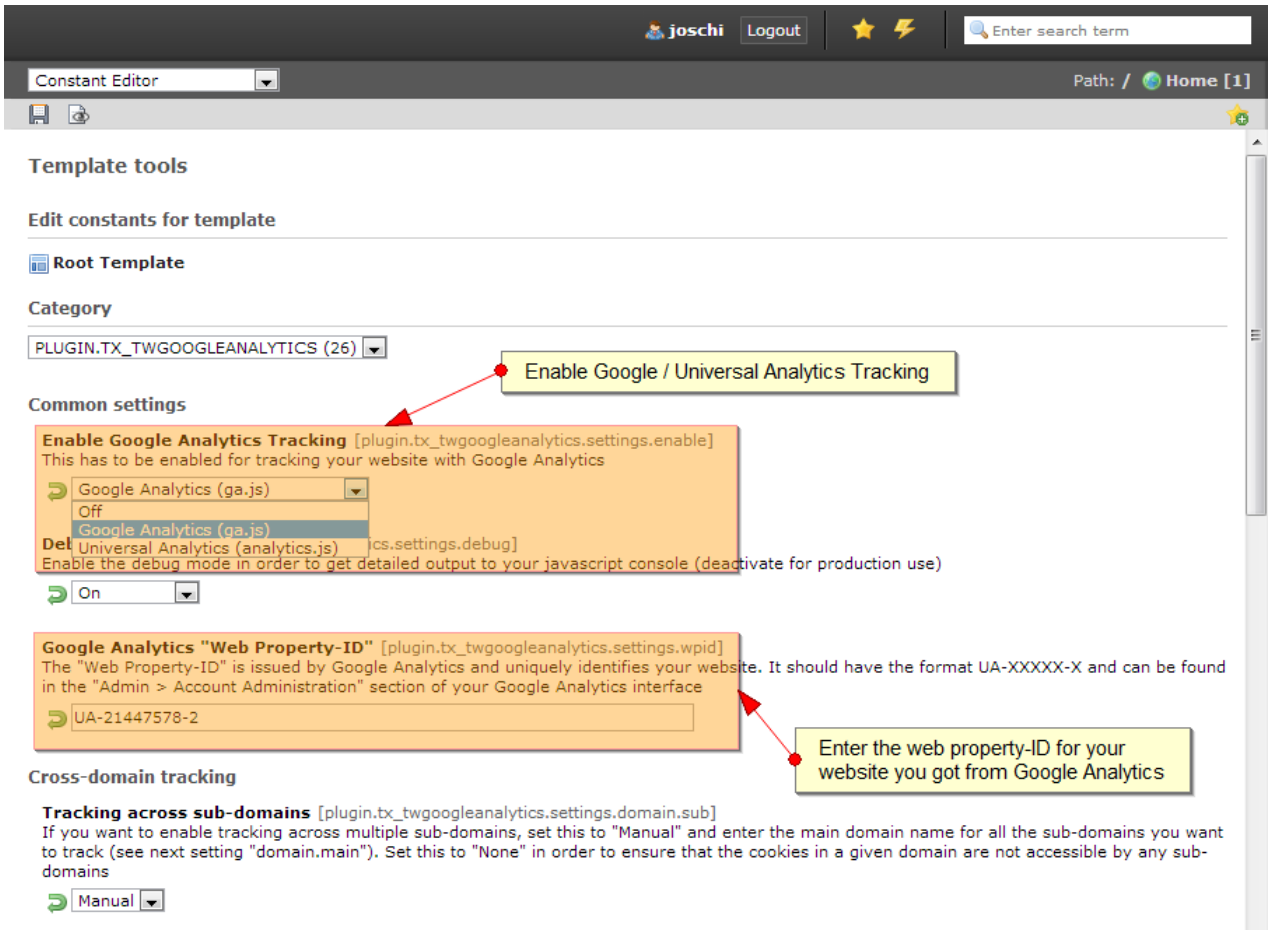
Note: Again, the example above assumes your main page object to be called "page" – change this appropriately if necessary. Also, the tracking code should be rendered as close as possible to the closing </head>-element of your HTML source code. In case you already used the headerData-index 9999 (or higher) 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**,
- enable **remarketing and advertising reporting features**,
- 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 *Googleanalytics/Track.html* 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)	{$plugin.tx_twgoogleanalytics.view.layoutRootPath}
settings.debug	int	**Debug mode** for printing tracking commands to the JavaScript console of your web browser **Options**: • **0**: Disabled • **1**: Enabled • **2**: Enabled in "Debug only" mode (real data transmission to Google Analytics will be suspended; useful for testing)	{$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.	{$plugin.tx_twgoogleanalytics.settings.wpid}
settings.features.advertising	boolean	Enables remarketing and advertising reporting features (see here for further information).	{$plugin.tx_twgoogleanalytics.settings.advertising}
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}
settings.crossdomain.sub	int	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 *crossdomain.main* setting). If you explicitly want to ensure that sub-domains are not tracked, you should set this to **0** ("none"). **Options**: • **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 *crossdomain.main* setting)	{$plugin.tx_twgoogleanalytics.settings.domain.sub}

Property:	Data type:	Description:	Default:

settings.crossdomain.main	string	Sets the **main domain name**. You need to do this if you set *crossdomain.sub* to something else than 1 ("auto").
		
		Example:
		`crossdomain.main = example.com`
		
		If you want to ensure, that only the sub-domains of your main domain are tracked, prepend the domain name with a dot.
		
		Example:
		`crossdomain.main = .example.com`
		
		This setting defaults to an empty string.
			{$plugin.tx_twgoogleanalytics.settings.domain.main}

settings.crossdomain.cross	string	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. **Example**: `crossdomain.main = first-domain.com, another.second-domain.com` This setting defaults to an empty string.	{$plugin.tx_twgoogleanalytics.settings.domain.cross}
settings.external.track	int	Enables tracking of clicks on **links to external domains**. You can choose if those clicks are submitted as "pageView" or as "event" records. **Options**: • **0**: Disabled (default) • **1**: Enabled, tracking as "pageView" • **2**: Enabled, tracking as "event" **Note**: In contrast to cross-domain tracking (see *crossdomain.cross*) the tracking cookie will not be passed to the external domains. Just the clicks on links to them will be registered.	{$plugin.tx_twgoogleanalytics.settings.external.track}
settings.external.prefix	string	Sets a **prefix for external domain tracking**. Depending on your *external.track* setting the prefix will be used • 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). This setting defaults to "External".	{$plugin.tx_twgoogleanalytics.settings.external.prefix}
settings.external.restrict	string	**Restricts external domain tracking to the specified domains**. Enter a comma-separated list of domains. This setting defaults to an empty string.	{$plugin.tx_twgoogleanalytics.settings.external.restrict}
settings.email.track	int	Enables tracking of clicks on **e-mail links**. You can choose if those clicks are submitted as "pageView" or as "event" records. **Options**: • **0**: Disabled (default) • **1**: Enabled, tracking as "pageView" • **2**: Enabled, tracking as "event"	{$plugin.tx_twgoogleanalytics.settings.email.track}
settings.email.prefix	string	Sets a **prefix for email link tracking**. Depending on your *email.track* setting the prefix will be used • as **URL prefix**, in case you are tracking e-mail links as "pageViews" (Example: "/Email:test@example.com"), or • as **event category**, in case you are tracking e-mail links as "events" (see Event tracking for further information). This setting defaults to "Email".	{$plugin.tx_twgoogleanalytics.settings.email.prefix}

Property:	Data type:	Description:	Default:
settings.email.restrict	string	**Restricts e-mail link tracking to specified e-mail addresses**. Enter a comma-separated list of e-mail addresses. This setting defaults to an empty string.	{$plugin.tx_twgoogleanalytics.settings.email.restrict}
settings.download.track	int	Enables tracking of clicks on **download links**. You can choose if those clicks are submitted as "pageView" or as "event" records. **Options**: • **0**: Disabled (default) • **1**: Enabled, tracking as "pageView" • **2**: Enabled, tracking as "event"	{$plugin.tx_twgoogleanalytics.settings.download.track}
settings.download.prefix	string	Sets a **prefix for download link tracking**. Depending on your *download.track* setting the prefix will be used • 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). This setting defaults to "Download".	{$plugin.tx_twgoogleanalytics.settings.download.prefix}
settings.download.template	string	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}": • **{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") Any other component of the template string will remain untouched and literally be sent to Google Analytics (after being prefixed with the *settings.download.prefix*). This setting defaults to "{pathname}".	

Property:	Data type:	Description:	Default:

settings.download.list	string	**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
		`path = space separated extension list`
		
		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,
		• if it's local path (relative to your site root) starts with the given *path* component, and
		• if the file extension is contained in the list of given *extensions*.
		
		Examples:
		`download.list = fileadmin/tutorials=pdf doc txt`
		`download.list = fileadmin=pdf, public/downloads=zip exe`
		
		As a special value for the *path* component you can use "/" for recursively matching any directory of your website. As a special value for the *extension* component you can use "*" for matching any file extension. Having said that it should be obvious that
		`download.list = /=*`
		effectively is a full wildcard (in which case it should be omitted as it dœsn't result in any restriction of download URLs).
		
		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".
			{$plugin.tx_twgoogleanalytics.settings.download.list}

settings.direct.keywords	string	Registers **keywords that should be treated as direct access** to your website. Enter a comma-separated list of keywords. 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. This setting defaults to an empty string.	{$plugin.tx_twgoogleanalytics.settings.direct.keywords}
settings.direct.referrers	string	Registers **referrers that should be treated as direct access** to your website. Enter a comma-separated list of domains. Traffic from external domains will generally be registered as "referral traffic" by Google Analytics. In case you are running several websites under different domains linking to each other, you might want to register incoming inter-domain traffic rather as direct access than as referral traffic. **Example**: `direct.referrers = my-sister-site.com` This setting defaults to an empty string.	{$plugin.tx_twgoogleanalytics.settings.direct.referrers}

Property:	Data type:	Description:	Default:

settings.searchenginges	string	Registers **additional search engines** and their appropriate query variable names. Enter a comma-separated list of search engine / query variable definitions.
		
		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:
		`search engine name = query variable name`
		
		The specified *search engine name* 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 *query variable name* to be present in the referring URL.
		
		Example:
		`searchengines = MySpecialSearchEngine=query, AnotherSE=search`
		
		In order to add a search engine definition to the beginning of the recognition list, append an exclamation mark to the query term:
		`searchengines = MostImportantSearchEngine=q!`
		
		This setting defaults to an empty string.
			{$plugin.tx_twgoogleanalytics.settings.searchengines}

settings.customDimensions	array	Registers **custom dimensions** for transmission to Universal Analytics.
		
		Custom dimensions can be used to send additional, arbitrary information along with “pageViews” and “events”. For further information on custom dimensions please see https://developers.google.com/analytics/devguides/collection/analyticsjs/custom-dims-mets
		
		This setting defaults to an empty array and can only be controlled via custom TypoScript. It holds a list of key-value-pairs, where the keys are used as dimension names (always starting with the keyword “dimension”, followed by an integer suffix). For the values, the same lookup functionality as for custom variables is supported (see chapter on custom dimensions and metrics).
		
		Example:
		`plugin.tx_twgoogleanalytics.settings.customDimensions {`
		`  dimension1 = Value for first dimension`
		`  dimension2 = Value for second dimension`
		`  dimension3 = GP`
		`  dimension3 {`
		`  1 = tx_ttnews[uid]`
		`  }`
		`}`
		
		Please be aware that you have to **register your custom dimensions at your Google Analytics account** prior to sending dimension data. Once a metric is defined for a page, it will automatically be sent along with all page views and events.	

Property:	Data type:	Description:	Default:
settings.customMetrics	array	Registers **custom metrics** for transmissions to Universal Analytics. Custom metrics are name-value pair tags that you can send along with "pageViews" and "events" in order to refine Universal Analytics tracking. Make sure to meet the following requirements: • Metric names have to start with the keyword "metric", followed by an integer suffix (e.g. "metric1"). • Metric values may only be integer or floating point number. 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.linkid.enable	boolean	Enables support for Enhanced Link Attribution (please see https://developers.google.com/analytics/devguides/collection/analyticsjs/enhanced-link-attribution). Disabled by default.	{$plugin.tx_twgoogleanalytics.settings.linkid.enable}
settings.linkid.cookie	string	Name of the cookie used to identify links (used for enhanced link attribution). Defaults to "*_gali*".	{$plugin.tx_twgoogleanalytics.settings.linkid.cookie}
settings.linkid.duration	int	Lifetime of the cookie used for enhanced link attribution (in seconds). Defaults to *30*.	{$plugin.tx_twgoogleanalytics.settings.linkid.duration}
settings.linkid.levels	int	Maximum number of parent element levels the enhanced link attribution script will look for unique IDs (id attribute). Defaults to *3*.	{$plugin.tx_twgoogleanalytics.settings.linkid.levels}

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:
		`plugin.tx_twgoogleanalytics.settings.pageUrl`
		`{`
		`  data >`
		`  typolink {`
		`  parameter.data = TSFE:id`
		`  returnLast = url`
		`  }`
		`}`
			`pageUrl = TEXT pageUrl.data = getenv:REQUEST_URI`

Property:	Data type:	Description:	Default:

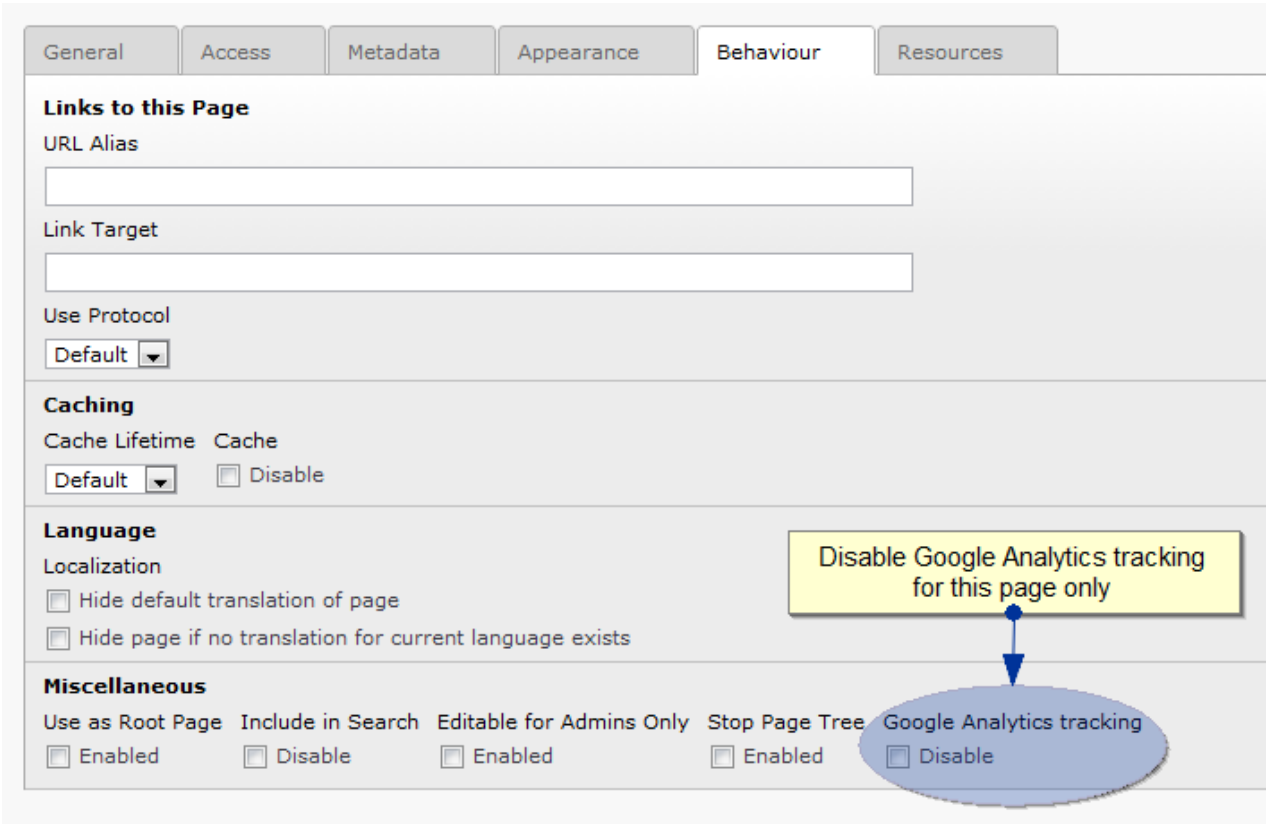
settings.user	string|cObject	This setting lets you make use of Google's **user ID tracking** feature: https://developers.google.com/analytics/devguides/collection/analyticsjs/cookies-user-id#user_id
		
		The value **must be set via TypoScript** and can either be a simple value or any type of content object. It's totally up to you which value you use as a user ID — typically you would get this through some kind of authentication mechanism (e.g. the currently logged in FE user).
		
		Example:
		`plugin.tx_twgoogleanalytics.settings {`
		`user = TEXT`
		`user.data = TSFE:fe_user|user|username`
		`}`	

[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:



Disable tracking for individual users (opt-out cookie)

Under certain conditions it is necessary that users can intentionally disable tracking (e.g. in conformance with the right of objection required by German law). Therefore, starting with version 2.5.1 the extension supports the opt-out cookie which is respected by the Google tracking code.

You can disable tracking by calling the JavaScript method **tw_gat.optOut()**, providing a single boolean parameter indicating if you want to opt out (or in again):

```
// Disable tracking
tw_gat.optOut(true);

// (Re-)Enable tracking
tw_gat.optOut(false);
```

Google recommends using this in conjunction with a link like this:

```
<a href="javascript:tw_gat.optOut(true)">Click here to opt-out of Google Analytics</a>
```

Further information can be found at https://developers.google.com/analytics/devguides/collection/gajs/#disable.

Using custom dimensions and metrics

Custom dimensions and metrics 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

dimensions and metrics please see
https://developers.google.com/analytics/devguides/collection/analyticsjs/custom-dims-mets.

There may be **up to 10 custom dimensions and / or metrics** 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 dœsn't impose any preferences here.

Custom dimensions and metrics have to be defined as TypoScript array. **Valid keys** for this array are the strings “dimensions” or “metric”, followed by an integer between **0 and 9** (including), representing the respective dimension / metric index. Each custom dimension / metric definition is an array itself, consisting of the following elements:

customVar

Property:	Data type:	Description:	Default:
name	string	**Name of the custom dimension / metric**. You can choose an arbitrary term here.	

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th><th>Default:</th></tr>
<tr><td>value</td><td>string / array</td><td>The value of the custom variable. This is what Google Analytics will refer to as “Custom variable value”. Currently two value types are supported:

String value
If the variable value is given as a string, this string will be used as literal value.

Example:
<pre>plugin.tx_twgoogleanalytics.settings.customVariables.1 {
	name = Section
	value = Global
	level = page
}</pre>
Array value
The value of a custom variable may be pulled from the current GET / POST variables. In this case value 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.

Example:
<pre>plugin.tx_twgoogleanalytics.settings.customVariables.1 {
	name = News
	value = GP
	value {
		1 = tx_ttnews[uid]
	}
	level = page
}</pre>
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 lookup 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.

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>
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.

Important: If any of the specified GET / POST dœs neither exist nor have a default value specified, then the whole custom variable is unresolvable an will not be sent to Google Analytics during this very page view.

Note: Any GET / POST value will be escaped using PHP's mysql_real_escape_string() before being substituted into the given lookup SQL statement, but it will not be padded with quotes or such by default. If you need a value to be quoted you will have to take care of this yourself by preparing the SQL statement appropriately.</td><td></td></tr>
</table>

Property:	Data type:	Description:	Default:
level	string	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. **Options**: • **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 dimensions or metrics 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 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 three 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

3. **optOut**: Use this method if you want to disable tracking (or re-enable it). Please see the chapter above for further details.

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.

Note: As of version 2.5.0 of this extension, the new Universal Analytics (UA) is experimentally supported as well. Please be aware that the “nonInteraction” argument for the “trackEvent” method is not supported if you choose UA.

Note: As of version 4.0.0 of this extension, only Universal Analytics (UA) is supported.

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 / dimensions / metrics via the page properties form (like metadata)

ChangeLog

Version:	Changes:
4.0.0	Dropped support for old Google Analytics; Added support for TYPO3 8.7+
3.0.4	Added minification to the tracking code
3.0.3	Added support for enhanced link attribution; Added support for user ID tracking
3.0.2	Added TYPO3 8.x support
3.0.1	Added remarketing & advertising reporting features option
3.0.0	Added TYPO3 CMS 7 support
2.6.0	Added TYPO3 CMS 6.2 support; Added Bower support
2.5.2	Maintenance release: Fixed a bug with page URL calculation (thanks to René Pflamm)
2.5.1	Added support for disabling tracking (opt-out)
2.5.0	Added experimental support for Universal Analytics (UA); bugfix with tracking client information
2.0.1	Fixed a problem with multiple file extensions (e.g. ".tar.bz2") when tracking downloads;
2.0.0	TYPO3 CMS 6 Release: No new features, but switch to PHP namespaces and the new class / sysext structure (no usage of the compatibility layer!) ATTENTION: Support for TYPO3 4.x has been dropped, use the 1.x versions instead! Removal of the backported "format.raw" viewhelper introduced in version 0.9.1 as fluid 6.0.0 is a requirement as of now
1.0.2	Bugfix in TypoScript setup
1.0.1	Bugfix in TypoScript setup; Switch to XLIFF language files
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