

Integration and Configuration of the Webtrekk Tracking Pixel

Version 4.4.7



1	Preface	7
2	Adjustment of the Tracking Pixel	8
2.1	Pixel Configuration	8
2.1.1	Global Configuration in webtrekk_v4.min.js	8
2.1.2	Minimal Page Specific Configuration	9
2.2	TrackID	10
2.3	Domain	10
2.4	Path to the Function Script	11
2.5	Function Call for the Tracking Script	11
2.6	NOSCRIPT Parameter	11
2.6.1	Mandatory-Parameters	11
2.6.2	Optional Parameters	12
2.7	Syntax	14
2.8	1st and 3rd Party Cookies	15
2.8.1	Secure Cookie	15
2.8.2	Using Your Own Track Domain	16
2.8.3	EverID	16
2.9	SSL	17
2.10	Pre-Rendering	17
2.11	Tab browsing	18
2.12	Web to App tracking	19
2.13	Request obfuscation	19
2.14	Request length	20
2.15	Request limiting	20
2.15.1	Filtering URL fragments	21
2.16	Request queue	22
3	Cross Device Bridge	22
3.1	Basic implementation - third-party cookie	22

3.2	Advanced implementation	23
3.2.1	Image cache cookie	23
3.3	Expert implementation - additional user characteristics	24
3.4	Deactivating the Cross Device Bridge	24
4	Integration Position	25
4.1	Standard Position	25
4.2	Redirect / Forwarding Pages	25
4.3	Frame Set	25
4.4	Asynchronous Loading	26
4.4.1	With Security Prompt	26
4.4.2	With Timeout	27
5	Multiple Accounts	27
5.1	Multiple Accounts with the Same Configuration	27
5.2	Multiple Accounts with Different Configuration	28
5.3	Data profile	28
6	Mandatory Configuration	30
7	Optionale Konfiguration	31
7.1	Pages	31
7.1.1	Customized Page Naming	31
7.1.2	Default Page Naming	32
7.1.3	Automatic Entry of the Page URL	32
7.1.4	Content Groups (Page Categories)	33
7.1.5	Page Parameters (Custom Parameters)	33
7.1.6	Predefined page information	34
7.2	Event Tracking	34
7.2.1	Automatic Event Tracking with the "Standard" Setting	35
7.2.2	Automatic Event Tracking with the "Link" Setting	36
7.2.3	Event Tracking with JSON Objects	39
7.2.4	Event Tracking with a Function Call	40
7.2.5	Event Tracking with Redirects	40
7.2.6	Event Parameters (Custom Parameters)	41

7.3	E-commerce Tracking	42
7.3.1	Produkt Information	42
7.3.2	Tracking Orders	44
7.3.3	E-commerce Parameters (Custom Parameters)	44
7.3.4	Predefined product information	45
7.3.5	Uploading e-commerce parameters	46
7.4	Campaign Tracking	47
7.4.1	Names for Media Code	47
7.4.2	Passing Campaign IDs	48
7.4.3	Campaign Parameters (Custom Parameters)	48
7.5	Categories	49
7.5.1	Time Categories	49
7.5.2	Campaign Categories	50
7.6	Custom Parameters	50
7.6.1	Session Parameters	51
7.6.2	Predefined session parameters	51
7.6.3	Independent Parameters	52
7.6.4	Tracking Downloads	52
7.7	Heatmap Tracking	53
7.8	Internal Search	54
7.9	Form Tracking	55
7.10	Custom Visitor IDs	59
7.10.1	URM category	60
7.10.2	Predefined URM categories	60
7.11	Media Tracking	61
7.11.1	Embedding in Flash	61
7.11.2	Embedding in Silverlight	63
8	Additional Functionality	63
8.1	Adobe Flash	63
8.2	Ajax	64
8.2.1	Event Tracking	65
8.2.2	Form Tracking	65
8.3	Server-to-Server Communication	66
8.4	Encrypted Configuration	66
8.4.1	On the Client Side	66
8.4.2	On the Server Side	67

8.5	Processes	67
8.6	Cross-Channel Tracking	68
8.6.1	Recognition via Webtrekk Ever ID	68
8.6.2	Recognition via Customer ID	69
8.6.3	Recognition via Mobile Apps	69
9	Data Protection Opt-Out	70
10	Checking Pixel Configuration	71
11	Examples	73
11.1	Page Parameters	74
11.1.1	"Text" Data Type	74
11.1.2	"Number" Data Type	74
11.2	Event Parameters	75
11.2.1	"Text" Data Type	75
11.2.2	"Number" Data Type	75
11.3	Session Parameters	76
11.3.1	"Text" Data Type	76
11.3.2	"Number" Data Type	76
11.4	Campaign Parameters	77
11.4.1	"Text" Data Type	77
11.4.2	"Number" Data Type	77
11.5	Independent Parameters	78
11.5.1	"Text" Data Type	78
11.5.2	"Number" Data Type	78
12	Features and Bugfixes	78
13	General terms of use	80
13.1	Object of these terms of use	81
13.2	Rights and obligations of the customer according to these terms of use	81
13.3	Rights and obligations of Webtrekk according to these terms of use	81
13.4	Data confidentiality	81
13.5	Protecting the rights of affected parties	81

13.6	Compensation	82
13.7	Technical and organisational data security measures	82
13.8	Term and cancellation	82
13.9	Final provisions	82
14	Contact	83

1 Preface

Webtrekk provides a high-end analytical tool to analyze the traffic on your website, identify success factors and optimize it on the basis of actual key indicators.

The basis for web analysis with Webtrekk is the pixel, which entails embedding a Webtrekk code in your pages. This guide describes the technical implementation of the pixel. Some aspects/parameters are mandatory while others are optional. Conceptual support about which functionalities and parameters you should use and how the corresponding values should be set can be found in our online support center (you can get access via your Webtrekk account). Our consultants are also available to help. Feel free to contact us!

Your account includes most of the tracking data transmitted to Webtrekk, which is standardized solely on the basis of the pixel. Accounts must be explicitly configured for some data, however, to enable Webtrekk to correctly record and interpret the information. This data can only be tracked after the configuration is applied. This affects:

- Campaigns
- Custom parameters
- Processes
- Categories (content groups, product, media, time categories)

Changes/corrections can only be made afterwards to categories. Since this is not possible with the other data, the account must be correctly configured from the start. You will also be explicitly informed when this is the case.

Each function is accompanied by a short description and followed by the technical specification required for implementation. Refer to this guide for help in identifying your tracking requirements. This document contains information on:

- Mandatory Configuration
- Optional Configuration
- Technical Details

A brief explanation of how to check pixel requests sent to Webtrekk is located towards the end of this document.

We wish you every success in your web analyses with Webtrekk.
Your Webtrekk Team

2 Adjustment of the Tracking Pixel

2.1 Pixel Configuration

Webtrekk distinguishes between the mandatory configuration that applies equally to all pages and page-specific configurations. The latter consists of the required minimum pixel configuration and additions to optional configurations.

2.1.1 Global Configuration in `webtrekk_v4.min.js`

The `webtrekk_v4.min.js` tracking script contains the global configuration of the tracking script. Parameters applying to all pages should be entered in this segment. All parameters that you set in the global configuration will be used as default values for every page. These can be overwritten by the corresponding parameters in the page-specific configuration settings.

- **trackId:** enter your Webtrekk customer ID here
- **trackDomain:** enter your Webtrekk tracking URL here
- **domain:** domain of the tracked page; used to recognise the referrer
- **cookie:** used 1st or 3rd-party cookies

Some parameters must be configured in the Webtrekk interface prior to the implementation of the pixel.

The configuration object "webtrekkConfig" contains the object "safetag", in which the TagIntegration ID and TagIntegration domain are defined..

- **async:** defines whether the TagIntegration container should be loaded asynchronously
- **timeout:** defines the maximum amount of time that a system should wait for the TagIntegration file to load
- **safetagDomain:** enter the TagIntegration domain here, if the TagIntegration file should be loaded from a Webtrekk server
- **safetagId:** enter your TagIntegration customer ID here, if the TagIntegration file should be loaded from a Webtrekk server
- **customDomain:** enter your domain here, if the TagIntegration file should be loaded from your server
- **customPath:** enter the path to your JavaScript file here, if the TagIntegration file should be loaded from your server
- **option:** additional TagIntegration information


```
/**
 * webtrekkConfig
 *
 * global webtrekk Configuration
 * global webtrekk config
 * @type Object
 */
var webtrekkConfig = {
  trackId: "11111111111111",
  trackDomain: "track.wt-eu02.net",
  domain: "www.website.com",
  cookie: "1",
  contentId: "",
  safetag: {
    async: true,
    timeout: 2000,

    /** Load SafeTag from Webtrekk Server */
    safetagDomain: "responder.wt-safetag.com",
    safetagId: "11111111111111",

    /** Load SafeTag from Customer Server */
    // customDomain: "www.domain.com",
    // customPath: "js/webtrekk_safetag.js",

    option: {}
  }
};
```

2.1.2 Minimal Page Specific Configuration

The following example highlights the minimal page configuration.

Page and action tracking are activated through the minimal integration of the page script. Page and action names are automatically generated based on the URLs. As a result of this, various standard analyses are then already possible.

```
<script type="text/javascript">
  var pageconfig = {
    contentId: "", // generates an automatic content ID based on the URL
    linkTrack: "link", // activates link tracking [link or standard]
    linkTrackAttribute: "id"
  };

  var wt = new webtrekkV3(pageconfig);
  wt.sendinfo();
</script>
```

Please note that Webtrekk CANNOT guarantee that the tracking pixel will function correctly if changes are made to the pixel source code, which are not described here.

The file webtrekk_v4.min.js must have been loaded prior to adjusting the configuration.

Webtrekk will only provide support if pixel adjustments have been made in accordance with these instructions.

2.2 TrackID

TrackIDs are used by Webtrekk to assign server requests to an account. Your special Webtrekk TrackID must be inserted into the tracking pixel. Normally, it should already be entered into the pixel that you received from us. You can also find your TrackID in the System Configuration area of the Webtrekk tool (Configuration > System Configuration > Data Collection).

Webtrekk TrackID must be inserted into the "<noscript>" segment of the tracking pixel.

```
<script type="text/javascript">
  var pageconfig = {
    contentId: "", // generates an automatic content ID from the URL
    linkTrack: "link", // activates link tracking [link or standard]
    linkTrackAttribute: "id"
  };

  var wt = new webtrekkV3(pageconfig);
  wt.sendinfo();
</script>
<noscript>
  
</noscript>
```

2.3 Domain

In the variable "wt.domain", domains can be defined, which should not be identified as a referrer in the Webtrekk tool. In most cases, the domain of the website being tracked is included in this variable. When using multiple domains, separate them with a semicolon. This configuration can be created globally in the .js and/or in a page-specific manner in the pixel.

```
<script type="text/javascript">
  var pageConfig = {
    // ...
    domain: "www.website.com;shop.website.com"
  };

  // or

  var wt = new webtrekkV3(pageConfig);
  wt.domain = "www.website.com;shop.website.com";
  wt.sendinfo();
</script>
```

It is also possible to use regular expressions when filtering multiple domains. If using a regular expression, its string must start with "REGEXP:". If you used predefined characters - e.g. a point - in their regular expression, this must be double-escaped (\\).

```
wt.domain = "REGEXP:^(www\\.website\\.com)|(shop\\.website\\.com)$";
```

Always enter domains without "http://".

2.4 Path to the Function Script

The function script "webtrekk_v4.min.js" includes all functionalities for collecting the tracking information. Each document to be tracked must load this script. The function script should be saved on the web server where the website intended for tracking is located. The path to the function script should be relative to avoid using a separate SSL integration. The function script should be centrally stored on the web server. This reduces updating outlay enormously if, for example, you want to update to a new pixel version.

```
<script type="text/javascript" src="webtrekk_v4.min.js"></script>
```

NOTE: Webtrekk recommends integrating the script in the website's header.

2.5 Function Call for the Tracking Script

The function call initiates the sending of the tracking pixel. You can always change the position of the function call if you want to start the function call in a specific place on the tracked page. However, you must ensure that the function script is activated before the function call and that this has generated a pixel instance.

2.6 NOSCRIPT Parameter

To track users with deactivated JavaScript, add a NOSCRIPT segment to the configuration. The NOSCRIPT segment consists of mandatory and optional parameters.

Example:

```

```

2.6.1 Mandatory-Parameters

Explanation of individual mandatory parameters

NOSCRIPT-Parameter	Description
http://track.wt-eu02.net/	Webtrekk TrackDomain
1111111111111111/	Webtrekk TrackID
p=447,	Webtrekk Pixel Version
de.home,	Name of the page [optional]
1,	JavaScript activated [optional]
1280x1024,	Screen resolution [optional]
32,	Color depth in bits [optional]
1,	Cookies activated [optional]
1208964116364,	Timestamp of the client (in milliseconds) [optional]
0,	No referrer (otherwise, the coded URL) [optional]
1234x784,	Internal resolution of the browser window [optional]
1	Java installed [optional]

Non-determinable mandatory parameters must be filled with the value "0". The Webtrekk TrackDomain, Webtrekk TrackID and Webtrekk pixel version parameters are always required and should therefore never have the value "0".

Minimal configuration of the NoScript pixel:

```

```

2.6.2 Optional Parameters

The following parameters must be UTF-8- and URL-encoded. The German "ü", for example, is coded as "%C3%BC". This will ensure the proper transmission of all special characters to Webtrekk. This recommendation applies to all websites, regardless of website coding.

The order of the columns can be freely selected. Non-filled parameters can be left empty or removed.

Optional no-script parameter	Javascript parameter	Description
fn	- is read automatically -	Form name
ft	- is read automatically -	Form fields
eid	- is read automatically -	Webtrekk EverId
tb	- is read automatically -	Marking the tabbed browsing request

Optional no-script parameter	Javascript parameter	Description
la	- is read automatically -	Country code for client language setting (such as 'de')
wt_vt	- is read automatically -	Name of the AB test
cdb	- is read automatically -	Cross-device bridge parameter [cdb2, cdb3, ...]
ct	wt.linkId	Click or link name
cp	wt.customParameter	Page parameter [cp2, cp3, ...]
cp770	- is read automatically -	Marking the tabbed browsing request
cp771	wt.numberSearchResults	Number of search results
cp772	wt.errorMessagees	Error messages
cp773	wt.paywall	Paywall calls
cp774	wt.articleTitle	Article header
cp775	wt.contentTags	Content tags
cp776	wt.pageTitle	Page title
cp777	wt.pageType	Page type
cp778	wt.pageLength	Page length
cp779	wt.daysSincePublication	Days since publication
cs	wt.customSessionParameter	Session parameter [cs2, cs3, ...]
cs800	wt.loginStatus	Login status
cs801	- is read automatically -	Pixel version
cs802	- is read automatically -	Tracking platform
ce	wt.customTimeParameter	Custom parameter [ce2, ce3, ...]
cb	wt.customEcommerceParameter	E-commerce parameter [cb2, cb3, ...]
cb563	wt.couponValue	Coupon value
cb760	wt.productSoldOut	Product sold out
cb761	wt.paymentMethod	Payment method
cb762	wt.shippingService	Shipping service provider
cb763	wt.shippingSpeed	Shipping speed
cb764	wt.shippingCosts	Shipping costs
cb765	wt.grossMargin	Margin/mark-up
cb766	wt.orderStatus	Order status
cb767	wt.productVariant	Product variant

Optional no-script parameter	Javascript parameter	Description
cd	wt.customerId	Customer ID
uc	wt.urmCategory	Customer parameter [uc2, uc3, ...]
uc701	wt.emailRID	E-mail receiver ID
uc702	wt.emailOptin	E-mail opt-in [1 = yes, 2 = no]
uc706	wt.gender	Gender [1 = male, 2 = female]
uc707	wt.birthday	Date of birth [YYMMDD]
ov	wt.orderValue	Order value
oi	wt.orderId	Order ID
ba	wt.product	Products in shopping basket
co	wt.productCost	Product cost
cr	wt.currency	Product currency
qn	wt.productQuantity	Number of products
ca	wt.productCategory	Product category [ca2, ca3, ...]
st	wt.productStatus	Status of the shopping basket [add, conf, view]
cg	wt.contentGroup	Page category [cg2, cg3, ...]
is	wt.internalSearch	Search term of the internal search function
mc	wt.campaignId	Campaign ID consisting of media code parameter and value ("wt_mc=newsletter")
mca	wt.campaignAction	Campaign action [c = click, v = view]
ck	wt.customClickParameter	Action parameter [ck2, ck3, ...]
cc	wt.customCampaignParameter	Campaign parameter [cc2, cc3, ...]

2.7 Syntax

Webtrekk supports the UTF-8 standard. A maximum of 255 characters can be entered per parameter. A request may not be larger than 7KB. Input is case-sensitive.

Note the following in the NoScript segment of the pixel:

In general, all parameters in the "<noscript>" segment of the pixel must be set according to UTF-8 and URL code standards. For example, the German "ü" as "%C3%BC". This will ensure the proper transmission of all special characters to Webtrekk and applies to all websites regardless of coding (UTF-8, ISO-8859-1, etc.). These coding requirements only apply to the noscript segment of the pixel. Parameters in the script segment are entered exclusively in plain text.

Certain characters are automatically filtered out during tracking:

- 0x00-0x1f
- ' "
- <
- >
- \
- 0xA0 => space

Several spaces become one. Trailing and ending spaces are removed.

2.8 1st and 3rd Party Cookies

You have the option of using tracking with first or third-party cookies. By default, first-party cookies are used. The difference is that with first-party cookies your website sets the cookie and with third-party cookies, Webtrekk sets the cookie.

Acceptance of first-party cookies in general is higher. The disadvantage of first-party cookies is that they can only count one domain, i.e. if your website is distributed on multiple domains, it is not possible to count a contiguous visit (e.g. www.website.com and www.website-special.com). Switches between various sub-domains (e.g. www1.website.com and www2.website.com) or HTTP/HTTPS is, however, supported.

Setting "1" activates the tracking pixel for first-party cookies, while setting "3" activates third-party cookies.

Note: The cookie settings of the user can not be considered, its the configuration of the tracking pixel that counts. For "1st-Party" the tracking pixel generates an EverID, saves into or reads it from the customer domain and adds it to every track request. For "3rd Party", the tracking pixel can not read the EverID and sends the request without it, our server then reads from or saves a cookie into the trackdomain.

```
// cookie handling  
wt.cookie = "1";    // (1st or 3rd party cookie)
```

Use of first-party cookies requires JavaScript. If the visitor does not accept JavaScript, the noscript segment of the tracking pixel is activated, which supports third-party cookies only. Regardless of the noscript segment, your website should only use one cookie setting.

2.8.1 Secure Cookie

With this option, you have the possibility to add the "secure" flag to all client side Webtrekk cookies. A secure cookie can only be transmitted over an encrypted connection (HTTPS). They cannot be transmitted over unencrypted connections (HTTP). A cookie is made secure by adding the *Secure* flag to the cookie.

This feature is deactivated by default and can be activated with the "cookieSecure" variable.

Note: Use this feature only if your complete website is only accessible via SSL. If you are not sure if this is the case for your website, don't use this flag.

```
// Example of a fully page-specific configuration:
```

```
var pageConfig = {  
  cookieSecure: true  
};  
  
var wt = new webtrekkV3(pageConfig);  
wt.contentId = "de.home";  
wt.sendinfo();
```

```
// Example of a fully global configuration:
```

```
var webtrekkConfig = {  
  trackId: "1111111111111111",  
  trackDomain: "track.webtrekk.net",  
  domain: "www.website.com",  
  cookieSecure: true  
};
```

2.8.2 Using Your Own Track Domain

Cookie setting "1" creates the required cookies in JavaScript and sets the cookies as 1st party in the customer domain. However, you can also have cookies generated as before by Webtrekk and use your own track domain. This sets the cookies in the customer domain for noscript users too, and they are recognised by the browser as 1st party cookies. Contact Webtrekk Support if you wish to use this option.

2.8.3 EverID

The EverID is a unique visitor ID, that is written into an ever cookie created by Webtrekk. This cookie ID is unique for each Webtrekk account and (when using 1st party cookies) for each domain.

Syntax of the ID

EverID is composed by several information:

- **Cookies created by the server (3rd party cookies):** 4 + Timestamp(10) + Random(8)
- **Cookies created by the browser (1st party cookies):** 2 + Timestamp(10) + Random(8)

Validate EverID?

If you activate this feature, the pixel checks if the generated EverID is valid (correct timestamp). If this is not correct, the pixel generate a new ID.

- **New syntax:** 2 + 3 + Random(4) + Random(5) + Random(8)


```
// Example of a fully page-specific configuration:
```

```
var pageConfig = {  
  validateEverId: true  
};
```

```
var wt = new webtrekkV3(pageConfig);  
wt.contentId = "de.home";  
wt.sendinfo();
```

```
// Example of a fully global configuration:
```

```
var webtrekkConfig = {  
  trackId: "1111111111111111",  
  trackDomain: "track.webtrekk.net",  
  domain: "www.website.com",  
  validateEverId: true  
};
```

2.9 SSL

If the tracked website is accessible through an https connection (e.g. an order form), it is necessary to make changes to the URL in the "<noscript>" segment of the tracking pixel. So http://. becomes https://. If this adjustment is not made, the web browser used by the website visitor may generate security warnings.

Line in the "<noscript>" segment:

```

```

No changes need to be made to the "<script>" segment for SSL.

If the path to the function script is absolute, an adjustment for embedding the function script is necessary. We recommend a relative embedding.

```
<script type="text/javascript" src="https://www.website.com/webtrekk_v4.min.js"></script>
```

2.10 Pre-Rendering

With pre-rendering, websites are loaded before the actual user visit. Webtrekk recognises this and suppresses the request by default. This ensures that only those pages are tracked that have actually been seen.

The same logic is used when a user opens your page via a new tab, but does not actually visit the page (i.e. doesn't open the tab).

If you would nevertheless like to track pre-rendered pages, simply use the variable "ignorePrerendering".

```
wt.ignorePrerendering = true;
```

2.11 Tab browsing

With tab browsing you can analyze how users navigate through your page. Only pages actually viewed by the user are measured. A page request is also sent for each subsequent tab view, even if the tab has already been viewed.

In order to differentiate these requests from others, the information of the additional tab browsing request is written to a defined page parameter of the number type. To unlock the required parameter in your Webtrekk account and for the configuration, please talk to your contact at Webtrekk.

This feature is deactivated by default and can be activated with the "tabBrowsing" variable.

```
var pageConfig = {  
    tabBrowsing: true  
};  
  
var wt = new webtrekkV3(pageConfig);  
wt.contentId = "de.home";  
wt.sendinfo();  
  
// or  
  
var webtrekkConfig = {  
    trackId: "11111111111111",  
    trackDomain: "track.webtrekk.net",  
    domain: "www.website.com",  
    tabBrowsing: true  
};
```

1. Example:

A user visits page A and opens pages B, C and D in a new tab. No page request is sent for pages B, C and D because the new pages are not visible yet. Page A is captured because it is being actively viewed by the user. The user subsequently switches to page B, then page D and back to page A. Now the tracking pixel sends the page requests for page B, then for page D and again for page A because it is visible for the second time. In the Webtrekk analysis tool, you can analyze that the user saw the pages in the sequence "A > B > D > A". If you deactivate this feature, you get the constellation "A > B > C > D".

2. Example:

Switching between several tabs with different domains can be analyzed as well. A user switches four times between page A (domain A) and page B (domain B) but page B does not belong to you. In analytics you can analyze that the user saw the pages in the sequence "A > A > A > A". If you deactivate this feature, you get the constellation "A".

Attention: When a user closes a tab that is not active and form tracking has been activated on this page, this form request is not sent.

2.12 Web to App tracking

Web to App tracking is relevant if you want to track web views in your app using the Webtrekk Pixel.

This function enables synchronous sending of requests to Webtrekk when your users open Web content directly in the app. Only in this way can the analysis data be evaluated correctly. An example of this is the display of visits to previous or subsequent pages in chronological order. This also helps to ensure that the correct user path is displayed in the path analysis and that the customer journey is evaluated correctly.

This function is disabled by default.

Please note that activating this feature can lead to a slightly increased security risk when exchanging data. This is because the pixel creates a data exchange to your Webtrekk SDK (which you use to track the app) and does not send the data directly to your Webtrekk account.

```
var pageConfig = {
    sendViaSDK: true
};

var wt = new webtrekkV3(pageConfig);
wt.contentId = "de.home";
wt.sendinfo();

// or

var webtrekkConfig = {
    trackId: "1111111111111111",
    trackDomain: "track.webtrekk.net",
    domain: "www.website.com",
    sendViaSDK: true
};
```

2.13 Request obfuscation

With this option, all track requests are obfuscated, to make it harder to identify webtrekk track requests.

- Random string appended after tracking endpoint ("wt")
- Random order of all tracking parameters (including "p")

This feature is deactivated by default and can be activated with the "requestObfuscation" variable.

```
var pageConfig = {
  requestObfuscation: true
};

var wt = new webtrekkV3(pageConfig);
wt.contentId = "de.home";
wt.sendinfo();

// or

var webtrekkConfig = {
  trackId: "1111111111111111",
  trackDomain: "track.webtrekk.net",
  domain: "www.website.com",
  requestObfuscation: true
};
```

2.14 Request length

A Webtrekk request may not exceed the maximum length of 7000 characters; all subsequent parameters will be cut off and not, therefore, tracked. In order to ensure all important parameters are tracked, you can define the order in which the parameters are transferred. To do this use the variable "paramFirst". Enter the respective NoScript parameter as the value and separate the individual entries with a semi-colon.

```
wt.paramFirst = "ov;oi;st;co;qn;ba;cd;cs15;cs16;cs17;cs18;cs19;cs20;";
```

In the above example, the values for Order Value, Order ID, Product Status, Number of Products, Products, Customer ID and the Session Parameters 15 to 20 will be written to the request first of all, followed by the remaining parameters.

2.15 Request limiting

The maximum number of permissible requests is limited in order to reduce the risk of sending large numbers of requests in error. This could be caused by an incorrect pixel implementation or by bot traffic. As soon as the standard limit of 1000 requests (pages and actions) every 30 minutes is exceeded, no further requests are sent by the pixel for the remaining time.

This feature is activated by default and cannot be deactivated. Only the number of requests and the time can be adjusted with the parameters "requestLimitAmount" and "requestLimitTime".

- **requestLimitAmount:** Maximum number of requests permitted to be sent in the specified time period.
- **requestLimitTime:** Time interval in seconds for sending the maximum X number of requests.

```
var pageConfig = {
    requestLimitAmount: 1000,
    requestLimitTime: 30*60
};

var wt = new webtrekkV3(pageConfig);
wt.contentId = "de.home";
wt.sendinfo();

// or

var webtrekkConfig = {
    trackId: "11111111111111",
    trackDomain: "track.webtrekk.net",
    domain: "www.website.com",
    requestLimitAmount: 1000,
    requestLimitTime: 30*60
};
```

2.15.1 Filtering URL fragments

Note that page URL are not accepted if values differ for individual visitors, such as a session ID. If these values are in the middle of the URL, or, as usual with Java servers, a ";" is appended to the file name, you have the option of filtering out these additions from the page URL with a regular expression.

A typical URL for a Java server might appear as follows:

http://www.webseite.com/teaser_a.htm;jsessionid=1CBE7F79EF7D681569A3BF30DD0C5D72.jp64?
page_id=1

To exclude this session ID in the tool from the page URL, filter it from the URL. To do this, enter a regular expression that places the session and the parameter in the URL. You can also enter a string to replace the located term. The following configuration filters out the session ID up to the first parameter and replaces it with a "?" and adds the remaining parameters to the end of the file name:

```
var pageConfig = {
    pageURLPattern: /;jsessionid=[a-zA-Z0-9\.[#|?|&]?/g,
    pageURLReplace: '?'
};

var wt = new webtrekkV3(pageConfig);
wt.contentId = "de.home";
wt.sendinfo();

// or

var webtrekkConfig = {
    trackId: "11111111111111",
    trackDomain: "track.webtrekk.net",
    domain: "www.website.com",
    pageURLPattern: /;jsessionid=[a-zA-Z0-9\.[#|?|&]?/g,
    pageURLReplace: '?'
};
```

You will now receive the page URL "http://www.webseite.com/teaser_a.htm?page_id=1".

2.16 Request queue

This feature is deactivated by default and can be activated with the "requestQueueActivated" variable.

- **requestQueueActivated:** Activate the request queue functionality.
- **requestQueueTTL:** Please enter the maximum time (in milliseconds) that a request should remain in the queue.
- **requestQueueResendInterval:** Please enter the interval (in milliseconds) after which a failed request should be sent again.
- **requestQueueSize:** Please specify the maximum number of requests that can be in the queue.

```
var pageConfig = {
  requestQueueActivated: true,
  requestQueueTTL: 5 * 60 * 1000,
  requestQueueResendInterval: 5 * 1000,
  requestQueueSize: 100
};

var wt = new webtrekkV3(pageConfig);
wt.contentId = "de.startseite";
wt.sendinfo();

// or

var webtrekkConfig = {
  trackId: "1111111111111111",
  trackDomain: "track.webtrekk.net",
  domain: "www.website.com",
  requestQueueActivated: true,
  requestQueueTTL: 5 * 60 * 1000,
  requestQueueResendInterval: 5 * 1000,
  requestQueueSize: 100
};
```

3 Cross Device Bridge

The Cross Device Bridge (CDB) technology verified for data privacy makes it possible to identify a user anonymously across multiple devices, apps and websites. A separate third-party cookie is created for this purpose and additional data for recognition can be sent to Webtrekk. Each of these functions can be activated/deactivated individually. We differentiate between three implementation levels for this purpose: Basic, Advanced and Expert. The higher the implementation level, the higher the data quality.

3.1 Basic implementation - third-party cookie

The Basic implementation consists of activating the Cross Device Bridge and using a third-party cookie under the "*.wcfbc.net" domain. All that is necessary here is to set the "execCDB" parameter to "true". To the extent possible, the e-mail address should also be transferred in encrypted form via the Cross Device Bridge plugin (see documentation for the Cross Device Bridge plugin).

The Basic implementation of the Cross Device Bridge is activated by default and can be deactivated with the "execCDB" parameter.

Possible requests:

```
Tracking with 1st-party cookie: //fbc.wcfbc.net/v1/fbc?eid=<<EID>>&acc=<<TRACKID>>&t=<<TIMESTAMP>>

Tracking with 3rd-party cookie: //<<TRACKDOMAIN>>/<<TRACKID>>/cc?
a=r&c=wteid_<<TRACKID>>&t=http%3A%2F%2Ffbc.wcfbc.net%2Fv1%2Ffbc%3Facc%3D<<TRACKID>>%26t%3D<<TIMESTAMP>>
```

Name of the cookie: wt_cdbeid
Cookie contents: /^[a-z0-9]{32}\$/
Cookie domain: *.wcfbc.net (without subdomain)
Cookie path: /
Cookie duration: 180

Example of the cookie contents: faef9fe66621c34327911c1291ed791c

3.2 Advanced implementation

To improve the accuracy, the browser's image cache and fingerprinting can be used in addition to the third-party cookie. In particular, the identification of users with mobile devices (Mobile Safari) and users that do not accept third-party cookies is very difficult without activating the image cache cookie and fingerprint.

3.2.1 Image cache cookie

In addition to the classic third-party cookie, the browser's image cache is used here to identify the user. The "Etag" header is used to make this possible; it contains the same cookie ID as the third-party cookie. From this ID, the CDBEID server under "fbc.wcfbc.net" dynamically generates an image that is imported on the client side and can be read with the help of an HTML element. Using the image cache is deactivated by default. You can activate the use of the image cache with the "useCDBCach" parameter.

Possible image cache requests:

```
Tracking with 1st-party cookies: //<<TRACKDOMAIN>>/<<TRACKID>>/cdb?
p=<<PIXELVERSION>>,0&eid=<<EID>>&cdbeid=<<CDBEID>>

Tracking with 3rd-party cookies: //<<TRACKDOMAIN>>/<<TRACKID>>/cdb?
p=<<PIXELVERSION>>,0&cdbeid=<<CDBEID>>
```

Further information about the **CDBEID** is found in the Webtrekk data privacy statement.

```
var pageConfig = {
  execCDB: true,
  useCDBCache: true,
};

var wt = new webtrekkV3(pageConfig);
wt.contentId = "de.home";
wt.sendinfo();

// or

var webtrekkConfig = {
  trackId: "11111111111111",
  trackDomain: "track.webtrekk.net",
  domain: "www.website.com",
  execCDB: true,
  useCDBCache: true,
};
```

3.3 Expert implementation - additional user characteristics

The best prerequisites for cross-device user identification are established via user characteristics such as the e-mail address, social media IDs and a few others. These characteristics can be transferred per user in encrypted ("hashed") form so there is no longer any way to access the original information.

The Cross Device Bridge plugin has to be used to transfer these and additional values. Values can be transferred to it in unencrypted form before they are normalised and then encrypted ("sha256") for sending.

For information about the Cross Device Bridge plugin and transferring additional user characteristics, please see the documentation for the Cross Device Bridge plugin.

3.4 Deactivating the Cross Device Bridge

All requests necessary for the Cross Device Bridge can be deactivated via the global or page configuration.

```
var pageConfig = {
  execCDB: false,
  useCDBCache: false,
};

var wt = new webtrekkV3(pageConfig);
wt.contentId = "de.home";
wt.sendinfo();

// or

var webtrekkConfig = {
  trackId: "11111111111111",
  trackDomain: "track.webtrekk.net",
  domain: "www.website.com",
  execCDB: false,
  useCDBCache: false,
};
```


4 Integration Position

Sending tracking information depends on the position of the tracking pixel on your page. If the user leaves a page before the pixel can be sent, Webtrekk receives no information about the user action. Webtrekk recommends a standard position for the tracking pixel.

4.1 Standard Position

Note: Webtrekk recommends integrating the tracking script in the header (<script type="text/javascript" src="webtrekk_v4.min.js"></script>).

```
<html>
  <head>
    <title>Start page</title>
    <script type="text/javascript" src="js/webtrekk_v4.min.js"></script>
  </head>
  <body>
    The content of your website is placed here.
  </body>
</html>
```

A pixel should be utilized in a similar way to page impressions (viewing of a page). We therefore recommend only using one pixel per page. Please also ensure that only one pixel at a time is invoked in Flash films.

4.2 Redirect / Forwarding Pages

When using redirects or forwarding pages it is important that the tracking pixel is fully loaded prior to carrying out a redirect.

Generally, no pixel should be used in redirects or page forwarding. Webtrekk cannot guarantee the tracking accuracy in relation to redirects or page forwarding.

4.3 Frame Set

Note the following special features in relation to use of the pixel in framesets.

The pixel should be embedded in all frames, but calling the 'sendinfo' method can only occur in the content frame, because otherwise several sessions would be generated causing unnecessary traffic. Tracking action paths would also not be possible otherwise.

To record media codes and referrers, this must be forwarded to the content frame. If this is not possible, a specially adapted function must be embedded to transmit this parameter. Contact Webtrekk Support for help.

When calling up overlay and heatmap analyses, the URL of the frame to be tracked must be entered directly since forwarding is not supported. In addition, the necessary URL parameter for overlay (wt_overlay) and heatmap (wt_heatmap) must be forwarded to the content frame.

4.4 Asynchronous Loading

With asynchronous loading, the file webtrekk_v4.min.js is loaded in the browser in the background.

Webtrekk does not recommend loading the file webtrekk_v4.min.js asynchronously, as this may under certain circumstances mean that not all of the relevant elements will be highlighted or information sent.

If the file webtrekk_v4.min.js is loaded asynchronously, care must be taken to ensure the Webtrekk object is only generated once webtrekk_v4.min.js has finished loading.

The call should therefore be invoked relatively early (in the header or at the start of the body tag). The page-specific configuration should then be placed before the closing body tag.

Various options are available for reloading.

4.4.1 With Security Prompt

This involves placing the page-specific configuration within a security prompt, in order to ensure the Webtrekk object is generated without any JavaScript errors. If it takes too long to reload the file webtrekk_v4.min.js, no data will be passed to Webtrekk.

```
if(typeof webtrekkV3 === "function") {  
  var webtrekk = {  
    contentId: "",  
    linkTrack: "link",  
    linkTrackAttribute: "id",  
    heatmap: "1",  
    form: "1",  
    formAttribute: "id"  
  };  
  
  window.wt = new webtrekkV3(webtrekk);  
  window.wt.sendinfo();  
}
```

4.4.2 With Timeout

With this option, the generation of the Webtrekk object is delayed for 3000 milliseconds, for example. The file `webtrekk_v4.min.js` should have been fully loaded by then. If the user changes or leaves the page within these 3 seconds, important data may be lost.

```
function createWebtrekkPixel() {
  if(typeof webtrekkV3 === "function"){
    var webtrekk = {
      contentId: "",
      linkTrack: "link",
      linkTrackAttribute: "id",
      heatmap: "1",
      form: "1",
      formAttribute: "id"
    };

    window.wt = new webtrekkV3(webtrekk);
    window.wt.sendinfo();
  }
}

window.setTimeout("createWebtrekkPixel()", 3000);
```

5 Multiple Accounts

If you want to use more than one Webtrekk account, it is possible to make tracking information from one account available in all accounts. The configurations for individual accounts can be identical or different. Since these requests are processed separately, all delivered accounts produce billing-relevant tracking traffic.

5.1 Multiple Accounts with the Same Configuration

If the same information should be entered in several accounts, simply enter their corresponding TrackIDs separated by a comma.

```
<script> segment:

wt.trackId = "1111111111111111,2222222222222222";
```

```
<noscript> segment:


```

5.2 Multiple Accounts with Different Configuration

To enter different information for a website in several accounts, a pixel instance must be generated for each. The instances will then work entirely independently of each other on the page and can be configured in any way. Similarly, separate tracking requests must be generated with the corresponding parameters in the noscript segment.

```
<script> segment:

var wt1 = new webtrekkV3(); // Generates the first instance
wt1.trackId="1111111111111111"; // Sets the TrackID for the first instance
wt1.sendinfo(); // Sends the tracking request for the first instance

var wt2 = new webtrekkV3(); // Generates the second instance
wt2.trackId="2222222222222222"; // Sets the TrackID for the second instance
wt2.sendinfo(); // Sends the tracking request for the second instance
```

```
<noscript> segment:



```

5.3 Data profile

This feature can be used to copy collected data (track requests) directly to other accounts. Filtering for the data that are to be transferred can be realised on the basis of every parameter in the track request. This means several different accounts can be populated with one pixel. The configuration is entered under "Configuration > System Configuration > Data Collection".

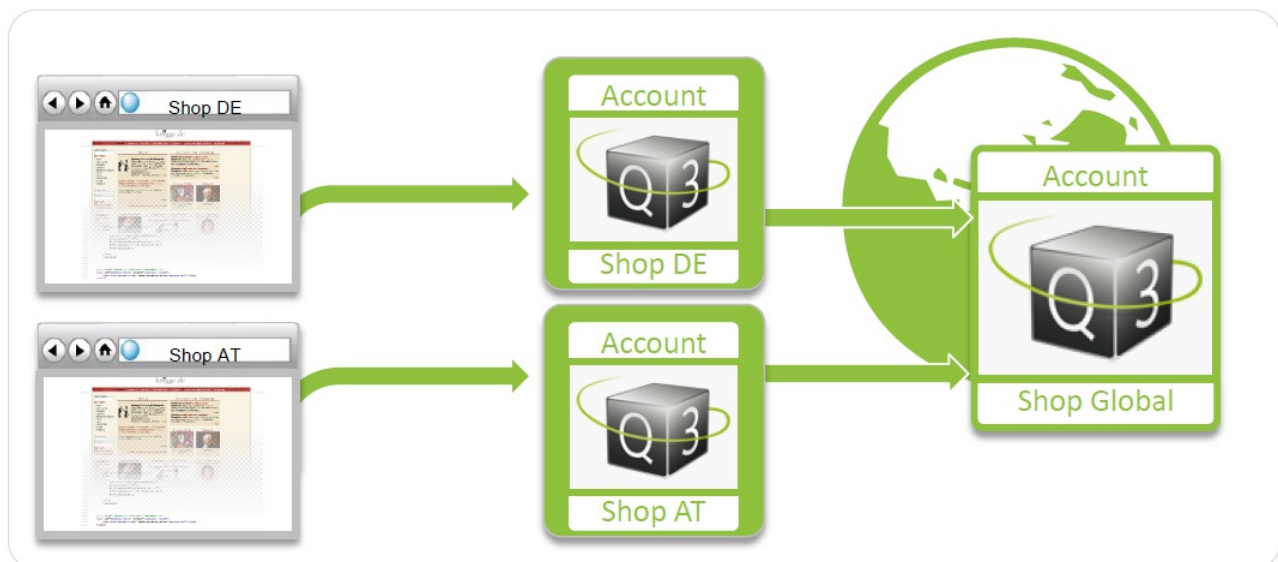
Example:

Data capture in an overall account ("Shop Global") where high-level evaluations are possible. Separate country accounts are used for shop-specific analyses ("Shop DE", "Shop AT" ...).

Copying from the global account to sub-accounts:



Copying from the sub-accounts to the global account:



Note that the setup of the source and target accounts should be identical (for instance custom parameters, categories...).

Configuration examples:

If the parameter cg1 ("content group 1") is set with the value "german", the track request is copied to the "Shop DE" account.

Dataprofile	Account	Parameter Filter	z. B. myshop1.com/[0-9]*/subshop/
	Shop DE	cg1	deutsch

If the domain is "mobile.shop", the track request is copied to the "Mobile Shop" account.

Dataprofile	Account	Parameter	Filter	z. B. myshop\com\[0-9]*/subshop/
	Mobile Shop	pu	mobile.shop\.	<input type="button" value="X"/>

Filters can be set with regular expressions.

6 Mandatory Configuration

Initial tracking data can already be generated with a minimal configuration. This allows initial information to be quickly evaluated in the Webtrekk tool during the test phase.

The integration process includes the following points:

Embed the webtrekk_v4.min.js file in your website. Ensure your account information (TrackID, TrackDomain) are available there.

Integrate the following code into your page source code. Ensure the correct path to the JS file is used.

This mandatory configuration ensures page tracking and action tracking are activated. Page and action names are automatically generated based on the URLs. As a result of this, various standard analyses are then possible.

The tracking of heatmap clicks is also activated. If your website is displayed in a centralised manner, remember that you need to set a reference point in order to generate valid data.

Based on requirements, this configuration can be extended with any of the parameters described in this document.

Please note that the majority of the web analysis functionalities in Webtrekk can only be used after configuring these optional settings.

Appropriate tools are available if you would like to see, directly via the browser, which requests are being transferred.

```
<!-- Webtrekk 4.4.7, (c) www.webtrekk.com -->
<script type="text/javascript" src="webtrekk_v4.min.js"></script>
<script type="text/javascript">
    var pageconfig = {
        contentId: "", // generates an automatic content ID based on the URL
        linkTrack: "link", // activates link tracking [link or standard]
        heatmap: "1", // activates heatmap tracking
        form: "1" // activates form tracking: Only works if the forms
                  // are actually marked
    };

    var wt = new webtrekkV3(pageconfig);
    wt.sendinfo();
</script>
```

7 Optionale Konfiguration

All of the parameters described in this section are optional and are used according to their tracking requirements. These settings do not affect how the "standard pixel" functions. Several of the following functions and parameters can be combined.

The following functions are dependant on the specific package, and not therefore available for every package.

7.1 Pages

As a rule, websites consist of individual pages through which the visitor navigates and carries out a variety of actions.

Since unique identification of each page is of central importance in web analysis, meaningful names should be chosen as the basis for convenient work with Webtrekk. The page names should be unique and provide a good idea as to what kind of page it is. For example, the homepage of a website that is called "index" or "homepage" is clear and understandable. Besides customized naming via the pixel, Webtrekk can also apply names automatically.

Using content groups, several pages can be joined to make aggregated evaluations possible. Different ways of calling up a page can also be described in detail by passing a page parameter.

7.1.1 Customized Page Naming

If page names are customized, the page name is transmitted in the pixel. This enables meaningful and short names.

Names should be unique to ensure pages are correctly identified. Uniqueness can also be ensured by the use of additional parameters set on the page in question. These may be:

- Product pages (differentiation by product parameters)
- Search results (differentiation by product parameters)

```
<script> segment:  
wt.contentId = "homepage";
```

```
<noscript> segment:  

```

Tips on how to name pages correctly can be found in our training document "Data Entry Basics", which is available from our support center.

7.1.2 Default Page Naming

If technical reasons prevent you from generating customized page names, the pixel can create a name on the basis of the page URL. Parameters in the URL are not taken into account in automatic generation.

For example, the URL "http://www.website.com/product_abc.htm?sid=7af49" will be automatically changed to form the page name "www_website_com.produkt_abc_htm". To automatically create a page name, the parameter "contentId" must be left empty in the global **or** page-specific configuration. A "0" must be entered in the noscript segment in place of the page name.

```
<script> segment:

var webtrekkConfig = {
  trackId: "1111111111111111",
  trackDomain: "track.wt-eu02.net",
  domain: "www.website.com",
  contentId: ""
};

// or

var webtrekk = {
  contentId: ""
};

var wt = new webtrekkV3(webtrekk);
wt.sendinfo();
```

```
<noscript> segment:


```

7.1.3 Automatic Entry of the Page URL

The page URL is entered in Webtrekk in the form of a mapping. In other words, the URL is only read out and saved the first time a Content ID is invoked. This, in turn, means that if you change the URL of a page subsequently, or only differentiate pages based on parameters (parameters are ignored), but the Content ID remains the same, the URL of the first page that was opened will always be displayed in the analysis window.

The following page is opened first of all:

```
www.domain.com?pid=23      // Content-ID = www_domain_com
                           // Page URL = www.domain.com?pid=23
```

Then the following page will be opened:

```
www.domain.com?p=100      // Content-ID = www_domain_com
                           // Page URL = www.domain.com?pid=23
```

7.1.4 Content Groups (Page Categories)

Content Groups are used to group pages and so form website areas.

Content Groups can be set as text or a number. In this process, the "text" data type can be used to map the website hierarchy. Content Groups of the data type "number" can be used to evaluate every page call with the value stored for it.

Content Groups must be configured in the Webtrekk tool so that they can be tracked. This is done under "Configuration > Categories > Content Groups". Here, the ID and data type (text/number) are defined for each content group.

Content Groups can be transferred in the pixel or an import via Excel or SOAP interface.

A list of tracked Content Groups of the data type "text" can be called up in the Webtrekk tool under "Navigation > Content Groups". Content Groups of the data type "number" are available as a metric in the analyses.

```
wt.contentGroup = {
  1: "damen",           // Main Category
  2: "oberbekleidung"  // Sub Category
};
```

Content group parameters should not exceed a length of 255 characters. All additional characters will be deleted.

Content groups are assigned one time to a page. If a page is tracked together with a content group, all subsequent page calls will also be assigned to this content group.

When passing the content group in the pixel, only the initial page call (content ID) will be taken into consideration. Webtrekk therefore recommends implementing content IDs and content groups while embedding the pixel.

7.1.5 Page Parameters (Custom Parameters)

You can use custom parameters (depending on your service level) to enrich analytical data with your own website-specific information and/or metrics. Observe the syntax guidelines when defining parameters.

Page parameters must be set up in the configuration (Configuration > Custom Parameters > Page Parameters) before they can be entered. The ID and data type (text/number) are defined for each parameter during setup, among other things.

Page parameters refer to single pages and are assigned to them directly. In contrast to content groups, the reference between the page and the page parameter does not have to be unique. This allows you to specify a page to call up, e.g. by entering the variant or a numerical value.

A list of measured page parameters of the data type "text" can be called up in the Webtrekk tool under "Navigation > Page Parameters". Page parameters of the data type "number" are available as a metric in the analyses.

```
wt.customParameter = {  
  1: "green",  
  5: "200.51"  
};
```

For further options, see the examples.

7.1.6 Predefined page information

You can enrich the analytics data with your website-specific information and/or metrics using "predefined page parameters and categories".

Predefined page information needs to be activated and set up in the configuration (Configuration > Custom Parameters > Page Parameters/Configuration > Categorisation > Content Groups) before it is captured.

```
wt.numberSearchResults = "15"; // Number of search results  
wt.errorMessages = ""; // Error messages  
wt.paywall = "1"; // Paywall calls [ 1 , 0 ]  
wt.articleTitle = ""; // Article heading  
wt.contentTags = ""; // Tags of an article  
wt.pageTitle = ""; // Title of the page  
wt.pageType = "Article"; // Type of page  
wt.pageLength = "large"; // Length of the page  
wt.daysSincePublication = "3"; // Days since publication
```

7.2 Event Tracking

Event tracking lets you track actions or clicks on internal or external links. Other visitor actions, such as clicking a button, can also be tracked as an action.

Measured actions are listed in the tool under "Navigation > Events".

Action tracking requires that JavaScript is active in the visitor's browser. If not, the action on your website will be executed as normal, i.e. your website's functionality will not be affected by action tracking.

The "linkTrack" parameter activates action tracking and must be passed during initialisation. This parameter must be entered if actions are to be automatically recorded. Possible settings for action tracking are "Standard" and "Link".

In some cases (e.g. with JavaScript and Flash links), the automatic recording of clicked links is not possible. These actions must be tracked using a separate function call.

An event is automatically assigned to the page, that was tracked last before the click. Thus, the name of the linking page does not have to be submitted in the name of the event.

Note regarding links on SSL pages:

Please note that requests for action tracking are not sent until the user leaves a page (exception: "Action tracking with function call" ("link" setting) and "Action tracking with redirect").

If the user is on an insecure page, and the next page is an SSL page, an SSL message may be displayed. To avoid this, pages in the insecure area with activated action tracking, which may be followed by an SSL page, must be configured with the setting `wt.forceHTTPS="1"`.

7.2.1 Automatic Event Tracking with the "Standard" Setting

If the "Standard" setting is selected, only those actions on links that are explicitly "marked" in the page source code will be counted. The name tag is generally used for this purpose to enter the desired name. A string or JSON object, which contains the desired information, can be used as the value of the marking tag in use. Many CMS solutions support automatic embedding of these markings.

The syntax of the link title is subject to the same conditions as page naming.

```
var webtrekk = {  
  linkTrack: "standard"  
};  
  
var wt = new webtrekkV3(webtrekk);
```

With this marking, the pixel would generate the Event ID "teaser_a".

Example of integration with a string:

```
<a href="teaser_a.htm" name="teaser_a">Click teaser A</a>
```

Example of integration with a JSON object:

```
<a href="teaser_a.htm" name="{ct:'teaser_a'}">Click teaser A</a>
```

Alternative Tags

If you have already used the "name" attribute for other functions, you can use any other attribute for generating the name of an action. It is defined in the "linkTrackAttribute" configuration parameter.

An alternative attribute such as "rel" could be used. Note that alternative tag use is not supported by older browsers such as Internet Explorer 5.0. These browsers will still use the "name" attribute.

```
var webtrekk = {  
    linkTrack: "standard",  
    linkTrackAttribute: "rel"  
};  
  
var wt = new webtrekkV3(webtrekk);
```

With this marking the pixel would generate the Event ID "teaser_a".

Example of integration with a string:

```
<a href="teaser_a.htm" name="internal_id" rel="teaser_a">Click teaser A</a>
```

Example of integration with a JSON object:

```
<a href="teaser_a.htm" name="internal_id" rel="{ct: 'teaser_a'}">Click teaser A</a>
```

7.2.2 Automatic Event Tracking with the "Link" Setting

The "Link" setting uses the action target of a link as the action tracking name. It is not necessary to change the link.

```
var webtrekk = {  
    linkTrack: "link"  
};  
  
var wt = new webtrekkV3(webtrekk);
```

The pixel generates the Event ID "www.webseite.com.teaser_a.htm".

```
<a href="teaser_a.htm">Click teaser A</a>
```

Limitations

Parameters in the link will not be considered. The following link in the "link" setting would be counted under the action name "www.website.com.teaser_a.htm":

```
<a href="teaser_a.htm?parameter=123">Click teaser A</a>
```

With this setting, all actions on the page will be tracked automatically, but a link consisting of just "#" or JavaScript code as the target name will not produce any meaningful results.

Tags to differentiate between a number of links with the same target

Additional link markings are always useful if a number of links on the same page lead to the same target, e.g. a text and a graphic link. The "name" attribute is used by default in this case. It is preset in the action ID in the Webtrekk tool.

If you have already used the "name" attribute for other functions, you can use any other attribute for name generation. It is defined in the "wt.linkTrackAttribute" configuration parameter. An alternative tag such as "rel" could be used.

Note that alternative attribute use is not supported by older browsers like Internet Explorer 5.0. These browsers will still use the "name" attribute.

```
var webtrekk = {  
  linkTrack: "link",  
  linkTrackAttribute: "rel"  
};  
  
var wt = new webtrekkV3(webtrekk);
```

The pixel generates the Event ID "teaser_textlink.www.webseite.com.teaser_a.htm".

Example of integration with a string:

```
<a href="teaser_a.htm" name="internal_id" rel="teaser_textlink">Klick auf Teaser A</a>
```

Example of integration with a JSON object:

```
<a href="teaser_a.htm" name="internal_id" rel="{ct: 'teaser_textlink'}">Klick auf Teaser A</a>
```

To automatically count actions, the underlying link names of which only differ in relation to their parameters, use the "wt.linkTrackParams" setting. In addition to the link target file name, use this setting to enter the parameters that are to be used for generating the action ID.

The following links, for example, can only be distinguished based on the "page_id" parameter:

```
<a href="page.htm?page_id=1">Link to page 1</a>  
<a href="page.htm?page_id=2">Link to page 2</a>
```

You can also enter several parameters for generating the Event ID (separated by semicolons).

```
var webtrekk = {  
  linkTrack: "link",  
  linkTrackAttribute: ""  
};  
  
var wt = new webtrekkV3(webtrekk);  
wt.linkTrackParams = "teaser_id;page_id";
```

Filtering out URL parameters

Note that no parameters are accepted in "wt.linkTrackParams" whose values differ for individual visitors, such as a session ID. If these values are not appended via the parameter but appear in the middle of the URL, or, as usual with Java servers, a ";" is appended to the file name, you have the option of filtering out these additions from the link target with a regular expression.

A typical URL for a Java server might appear as follows:

http://www.webseite.com/teaser_a.htm;jsessionid=1CBE7F79EF7D681569A3BF30DD0C5D72.jp64&page_id=1

To exclude this session ID in the tool from the action ID, filter it out of the URL. To do this, enter a regular expression that places the session and the parameter in the link. You can also enter a string to replace the located term. The following configuration filters out the session ID up to the first parameter and replaces it with a "?" and adds the remaining parameters to the end of the file name:

```
wt.linkTrackPattern = /;jsessionid=[a-zA-Z0-9\.] +[#|?|&]?/g;  
wt.linkTrackReplace = "?";
```

You will now receive the link "http://www.webseite.com/teaser_a.htm?page_id=1", via which the Event ID is generated.

Excluding individual links

You can use the variable "linkTrackIgnorePattern" to define a regular expression, which excludes the individual links from being tracked. This would be useful, for example, for anchor and JavaScript links. They appear in the tool as, for example, an action with the name "javascript(void)" and do not allow any meaningful analyses, as it is unclear which click was responsible.

```
var webtrekk = {  
  linkTrack: "link",  
  linkTrackIgnorePattern: "^(javascript:|#)"  
};  
  
var wt = new webtrekkV3(webtrekk);
```

Delaying Events

In order to track all events that occur on a page it is necessary to delay the user navigating to the next Page. In order to do so the tracking pixel will delay the click event and thusly delay the user from leaving the current Page for a very short time. This feature is deactivated by default. It can be activated via the parameter "delayLinkTrack". The default delay time is set to 200 ms (milliseconds). In case you deem 200 ms too much/too little you can adjust the delay via the parameter "delayLinkTrackTime". Webtrekk recommends a delay of 200-500 milliseconds.

```
wt.delayLinkTrack = true;  
wt.delayLinkTrackTime = 300;
```

In order to evaluate if the link will not leave the current page the following factors are taken into consideration:

- The "href" attribute of the Link is undefined or empty
- The Link is a JavaScript Link (href="javascript:") or an Anchor (href="#")
- The Link is indicated as a download
- The Link has the attribute "target=_blank", which will be opened in a new tag

Alternatively you can use the Parameter "noDelayLinkTrackAttribute" in order to explicitly tag internal Links. If this Parameter is set ALL Links on the current Page are considered external Links.

Thusly you should tag all links that will remain on the current Page with the defined Parameter, e.g. slider, tabs, anchor, javascript actions, etc.

```
wt.delayLinkTrack = true;  
wt.delayLinkTrackTime = 300;  
wt.noDelayLinkTrackAttribute = "data-wt-delay";
```

```
<a href="http://www.domain.tld/" target="_blank" name="logo" data-wt-delay="1">Link</a>
```

In the example above the event will not be delayed when the user clicks on the Link.

Please keep in mind: Only activate this feature if you require a very high data quality for your event tracking since this feature artificially delays the website when a user clicks on a Link that would leave the current Page.

7.2.3 Event Tracking with JSON Objects

With the help of JSON objects it is possible to provide an event with additional information, in addition to its name. These are generally custom event parameters. Custom ecommerce parameters can also be used in exceptional cases. All optional parameters, which can be evaluated together with a single click, are supported.

Prerequisite for entering the parameters in JSON objects is the activation of event tracking.

The example below shows a link with the event name 'link7' and the custom event parameters with ID 7 (value: 'action7') and ID 15.

```
<a name="{ct: 'link7', ck: {7: 'action7', 15: 'action15'}}" href="link.html">Link</a>
```

Depending on whether event tracking is activated, the pixel will generate the Event ID "link7.www.website.com.link.html" or "link7". The click will be assigned the action parameters "7" and "15", irrespective of the selected type.

7.2.4 Event Tracking with a Function Call

In some cases (e.g. with JavaScript and Flash links), automatic recording of events is not available. These links can, however, be tracked using a function call. The "wt.linkTrack" parameter is irrelevant for this type of action tracking.

Event tracking with a function call requires that JavaScript is active in the browser of the visitor. For technical reasons, the events tracked in this way cannot be evaluated in the "Overlay" analysis.

There are two variations that must be distinguished when tracking by function call:

Variant when leaving the current page

If the link is clicked (=action), the current website is closed and a new page is loaded.

```
<a href="contact.htm" onclick="wt.sendinfo({linkId: 'aktionsname'});">Link to contact page</a>
```

Variant without leaving the current page

Clicking the link changes the content on the same website (e.g. buttons, JavaScript actions).

```
<input type="button" onclick="wt.sendinfo({linkId: 'aktionsname'});" value="test" />
```

Flash tracking

This type of action tracking also counts events within a Flash film.

```
getURL("javascript:wt.sendinfo({linkId: 'aktionsname'});");
```

Please note that a differentiation should be made between page calls (page impressions) and "events" in Flash films, as well.

7.2.5 Event Tracking with Redirects

Event tracking with redirects is used to count clicks on HTML links. JavaScript links cannot be tracked. JavaScript is not required for this action tracking.

Please note that event tracking with redirect may impact the performance of the event to be tracked. We therefore recommend carrying out some tests.

1. Connection protocol (http or https)
2. TrackID
3. Target URL
4. Action ID
5. TrackDomain


```
<a href="http(1)://track.wt-eu02.net(5)/11111111111111(2)/re.pl?  
t=http%3A%2F%2Fwww.webseite.com%2Findex.htm(3)&c=link_to_homepage(4)">Link to homepage</a>
```

If tracking is performed in an SSL-protected zone, change the tracking call to "https"

IMPORTANT: the target URL and page name must be URL-encoded!

You can optionally hide the Webtrekk tracking URL in the browser status bar when hovering over the link to be tracked with the mouse:

```
<a href="http://track.wt-eu02.de/11111111111111/re.pl?  
t=http%3A%2F%2Fwww.webseite.com%2Findex.htm&c=link_to_homepage" onMouseOver="status='This is a  
link';return true;" onMouseOut="status=' ';return true;">Link to homepage</a>
```

When moving over this link with the mouse, the text "This is a link" appears in the browser's status bar, not the Webtrekk tracking URL.

7.2.6 Event Parameters (Custom Parameters)

You can use custom parameters (depending on your service level) to enrich analytical data with your own website-specific information and/or metrics. Observe the syntax guidelines when defining parameters.

Action parameters must be set up in the configuration (Configuration > Custom Parameters > Event Parameter) before they can be entered. The ID and data type (text/number) are defined for each parameter during setup, among other things.

Event parameters may refer to a single link and be directly assigned to it or globally to all links on the page.

```
wt.customClickParameter = {  
  linkid_1: {  
    1: "200.51",  
    5: "green"  
  },  
  2: "no values"  
};
```

In this example, clicking on the link with the ID (or Name tag) "link_1" additionally sends Event Parameter 1 with the value "200.51" and Event Parameter 5 with the value "green". For all other links, Event Parameter 2 is additionally sent with the value "no values".

You also have the option of sending custom parameters via event tracking with a function call.

A list of the tracked event parameters of the data type "text" can be called up in the Webtrekk tool under "Navigation > Event Parameters". Event parameters of the data type "number" are available as a metric in the analyses.

```
<a href="contact.htm" onclick="wt.sendinfo({linkId:'akitionsname', customClickParameter:{1:'200.51',5:'green'}});">Link to contact page</a>
```

```
<input type="button" onclick="wt.sendinfo({linkId:'aktionsname', customClickParameter:{1:'200.51',5:'green'}});" value="test" />
```

For further options, see the examples.

7.3 E-commerce Tracking

In Webtrekk you can analyse in detail which products on your website are viewed, placed in the shopping cart and bought. Aggregated evaluations are possible across product categories. Webtrekk automatically derives abandoned shopping carts from the information transmitted. Comprehensive information must be provided on the underlying order for product purchases, e.g. a unique order number.

Like orders, any additional information can be added to products using e-commerce parameters.

7.3.1 Produkt Information

The following parameters serve to track products in detail. The products can be transmitted to Webtrekk when a product is viewed, placed in the shopping cart or when the cart is purchased. A list of tracked products appears in the Webtrekk tool under "E-Commerce > Products".

Product configuration details must be added dynamically to the pixel. Do not use a thousands separator in the price details. Decimal places are separated using a point or comma.

Order / cart data is not processed when "Event Tracking with Function Call" is used.

wt.product

Saves products placed in the shopping cart. If several products appear in the shopping cart, they are each separated by a semicolon. This parameter must be entered if products are to be measured. All other parameters are optional for product tracking. A product name may not contain more than 110 characters.

wt.productQuantity (optional)

Contains the product quantity. If several products are transmitted, they are each separated by a semicolon. The standard value is "1".

wt.productCost (optional)

Contains the product price ("0" prices are allowed). If you transmit a product several times (quantity parameter wt.productQuantity greater than 1), use the total price not the unit price. If several prices are transmitted, they are each separated by a semicolon. The standard value is "0".

wt.currency (optional)

Contains the currency code of a product or order; the value must be passed to the Webtrekk pixel in line with the ISO standard. If multiple products are transmitted on a single page (e.g. on the order confirmation page if more than 1 product was purchased), only 1 currency will be applied to all products. This means that the value only needs to be set once.

Note: The currency is only passed for the purpose of currency conversion. In other words, the currency will be converted to the one (if any) that is stored in the Webtrekk front-end (Configuration > System configuration: Data Collection). Only one currency is ever displayed here.

wt.productStatus (optional)

Contains the shopping cart status.

If a product is viewed (e.g. on a product's detailed view), the status is "view". This status should always be set if the product can be added to the shopping cart.

If the product is added to the shopping cart, the status is "add". If the shopping cart is purchased, the status "conf" is transmitted. If no status is transmitted when tracking a product, the standard value "view" is assumed.

wt.couponValue (optional)

Contains the value of a coupon. Use this parameter if the customer makes an order with a coupon.

wt.productCategory (optional)

Product categories allow the grouping of products. The relationship between product and product category must be unique. In other words, it is not possible to assign the product "Shoes" once to the "Ladies" product category and once again to the "Sale" product category. Such non-unique relationships can be mapped using e-commerce parameters.

Product categories of the data type "number" can be used to evaluate every product call with the value stored for it.

Product categories must be configured in the Webtrekk tool for Webtrekk to measure this information. This is done under "Configuration > Categorisation > Product Categories".

The same rules for product category syntax apply as for page naming. A product category may not contain more than 110 characters.

With the exception of the "wt.productStatus" parameter, all other parameters must contain the same number of objects, e.g. 2 products, 2 product prices, 2 categories, etc.

A list of measured product categories of the data type "text" can be called up in the Webtrekk tool under "E-Commerce > Product Categories". Product categories of the data type "number" are available as a metric in the analyses.

Product categories are only assigned one time for a product. If a product is tracked together with a category, all other products will also be assigned to this category. Therefore, if the product status "view" has to be invoked prior to the purchase of a product, it is sufficient if product categories are only passed on at that point.

```
wt.product = "pullover;jeans"; // products
wt.productCategory = {
  1: "tops;jeans",           // optional: 1 = category
  2: "noname;levis"          // optional: 2 = brand
};
wt.productQuantity = "2;1"; // optional: quantity
wt.productCost = "99.90;69.95"; // optional: costs
wt.productStatus = "conf"; // optional: product status: [add|conf|view]
wt.currency = "EUR"; // optional: product currency
```

7.3.2 Tracking Orders

Webtrekk can also track orders. To do this, the order value is transmitted along with the order number. "0" values are permitted. A list of tracked orders can be called up in the Webtrekk tool under "E-Commerce > Orders separately".

The difference to product tracking is that the information refers to the total order value that is transmitted rather than the individual products. Besides the total of purchased products, the total order value may also contain such values as discounts and shipping and packaging costs.

The order value per order must be added to the pixel dynamically. Do not use any thousands separator in the price details. Decimal places are separated using a point or comma.

The "wt.orderValue" parameter saves the total order value. This parameter must be entered if total order values are to be tracked.

The "wt.orderId" parameter (optional) contains a unique order number (order ID). Use of this setting ensures that no orders are counted twice.

The currency of an order can be set with the parameter "wt.currency".

```
wt.orderId = "M-12345";
wt.orderValue = "52.99";
wt.couponValue = "10.00"; // optional: coupon value
```

7.3.3 E-commerce Parameters (Custom Parameters)

You can use custom parameters (depending on your service level) to enrich analytical data with your own website-specific information and/or metrics. Observe the syntax guidelines when defining parameters.

E-commerce parameters must be set up in the configuration (Configuration > Custom Parameters > E-Commerce Parameter) before they can be entered. In this setup, an ID that is relevant for tracking and a data type (text/number) are defined for each parameter.

Parameter references

E-commerce parameters are used to transmit additional product information (e.g. size, colour). In the case of several products, the number of single parameter values must match the number of products. The values are separated by a semicolon.

E-commerce parameters can also be used to transmit information about an order, e.g. payment or shipping type. In these cases, order tracking must be used. It is enough to transmit this parameter once per order. It applies equally to all products in the shopping cart.

The reference (product or order) is selected when configuring the pixel. If "individual value" is selected, the parameter refers to the order. If "multiple values" have been selected, the partner can refer to the product or the order.

Note: as website targets in Webtrekk must always be entered as e-commerce parameters, it is also possible to transfer e-commerce parameters that are not linked to orders or products.

E-commerce parameters of the type "Text" are shown in the Webtrekk tool under "E-Commerce > E-Commerce-Parameters". E-commerce parameters of the type "Number" are available as a metric in the analyses.

The configuration example shows a successful order of two products. E-commerce parameter 1 (size) and 2 (colour) have a product reference. E-commerce parameter 3 (payment type) has an order reference.

```
// products
wt.product = "pullover;jeans";
wt.productCategory = {
  1: "tops;jeans",
  2: "noname;levis"
};
wt.productQuantity = "2;1"; // optional: quantity
wt.productCost = "99.90;69.95"; // optional: costs
wt.productStatus = "conf"; // optional: product status: (add|conf|view)
wt.currency = "EUR"; // optional: product currency

// order
wt.orderId = "M-12345";
wt.orderValue = "52.99";

// custom e-commerce parameter
wt.customEcommerceParameter = {
  1: "L;32",
  2: "green;blue",
  3: "bill"
};
```

For further utilisation options, see the examples.

7.3.4 Predefined product information

You can enrich the analytics data with your website-specific information and/or metrics using "predefined product information".

Predefined e-commerce parameters and product categories need to be activated and set up in the configuration (Configuration > Custom Parameters > E-Commerce-Parameters) before they are captured.

```
wt.paymentMethod = "paypal"; // Payment method
wt.shippingService = "dhl"; // Shipper
wt.shippingSpeed = "express"; // Shipping speed
wt.shippingCosts = "4.95"; // Shipping costs
wt.grossMargin = "12.95"; // Margin/mark-up
wt.orderStatus = ""; // Order status
wt.productVariant = "green"; // Product variant
wt.couponValue = "10.00"; // Coupon value
wt.productSoldOut = "1"; // Product sold out [sold out = 1, in stock = 0]
```

7.3.5 Uploading e-commerce parameters

Individual orders and products bought on them can be supplemented with additional information after the fact by upload (e-commerce parameters). The functionality is explained using cancellations as an example:

Cancellations can be transferred in Webtrekk for entire orders or individual products within orders. In order to do so, a corresponding e-commerce parameter first has to be configured (Configuration > Custom Parameters > E-Commerce Parameters).

Webtrekk recommends transferring the cancellation data over a SOAP interface. Since the number of uploads per day is limited, the data should be transferred together. Alternatively the import can be realised with an Excel file. For correct formatting, calling an export of the e-commerce parameters for the cancellation values as a template is recommended (Configuration > Custom Parameters > E-Commerce Parameters).

Order number	Product name	Parameter (text) – cancellation reason	Parameter (number) – cancellation value
12003	Hose Malta	Does not fit	49,95
12004	Jacke	Do not like	56,90
12004	Hose Malta	Does not fit	49,95

To capture cancellations for multiple orders of the same products, the number of cancelled products can be transferred as an additional parameter. For example, it is conceivable for a customer to order a product in several sizes and only keep the ones that fit. If a cancellation value is transferred, it has to be for the total number of cancelled products.

Order number	Product name	Parameter (text) – cancellation reason	Parameter (number) – cancellation value	Parameter (number) – number of cancellations
12003	Hose Malta	Does not fit	49,95	1
12004	Jacke	Do not like	56,90	1
12004	Hose Malta	Does not fit	49,95	2

Note: Transferring more than one piece of information for ONE product within an order (for example more than one cancellation reason) is only possible via more than one parameter. Uploading the information again overwrites the previously stored information. Webtrekk does not converge various uploads, so that corrections can be made by means of uploads.

7.4 Campaign Tracking

Campaign tracking is configured in the Webtrekk tool (Configuration > Marketing Configuration). Without this configuration, no campaign information such as campaign clicks will be collected. Visits to certain pages or the entry of defined links can be tracked as campaign clicks. Most importantly, campaign tracking uses specific parameters – so-called media codes - that are added to the target URLs of the ads.

Using a media code improves the accuracy of the data collected with the pixel. Media codes can also be overwritten or supplemented with additional information using campaign parameters.

7.4.1 Names for Media Code

If you use media codes as a data source for your campaign tracking, entering their name can raise the accuracy of the measurement. Otherwise, accuracy may be reduced by up to 10% if certain firewalls are used, for example.

Example: You have created a campaign called "ABC" in the Webtrekk tool and measure it using the media code process (data source settings e.g. "URL parameter: mc" and "Value: campaign.abc"). The URL for this campaign must be as follows: <http://www.website.com/index.htm?mc=campaign.abc>.

The campaign will then be identified by the URL parameter "mc". The value of the URL parameter ("campaign.abc") plays no role in pixel configuration.

To raise the accuracy of the media code process, enter the URL parameter used for campaign identification, e.g. "mc", in the pixel's configuration segment. This configuration needs only to be made for pages referring to the campaign. You can also define several media codes, each separated by a semicolon.

Note that the media code parameter and value must be in lower case. The media code setting requires JavaScript.

```
wt.mediaCode = "mc"; // mediacode
```

If you only want to track each campaign once within a specific session, you can force this by using the variable "mediaCodeCookie". The pixel will then overwrite the campaign with "ignore" from the second page impression. This means that if a campaign is clicked more than once within a specific session, only the first campaign click will be evaluated.

```
wt.mediaCode = "mc"; // mediacode  
wt.mediaCodeCookie = "sid";
```

7.4.2 Passing Campaign IDs

You have the option of setting your own campaign ID in the configuration segment. A campaign ID consists of a media code name and its value, separated by "%3D". Campaign IDs are set in the "wt.campaignId" variable.

Furthermore, the optional parameter "campaignAction" can be used to determine whether the action was carried out based on a "view" or "click".

```
wt.campaignId = "mc%3DNewsletter_2010_08"; // campaign id  
wt.campaignAction = "view";
```

This parameter overwrites the media code values, which are read out via the referrer.

7.4.3 Campaign Parameters (Custom Parameters)

You can use custom parameters to enrich analytical data with your own website-specific information and/or metrics. Observe the syntax guidelines when defining parameters.

Campaign parameters must be set up in the configuration (Configuration > Custom Parameters > Campaign Parameter) before they can be entered. The ID and data type (text/number) are defined for each parameter during setup, among other things.

Campaign parameters always refer to an advertising medium (the smallest subunit of a campaign in Webtrekk).

Campaign parameters can either be entered directly in the page configuration or in the campaign configuration with a target URL along with the media code. If the same parameters are used for both the URL and the page configuration, the latter takes precedence and overwrites the URL parameter.

A typical example of transmitting a campaign parameter by URL is a link position in a newsletter. If the newsletter contains several links, this identifies which one was used.

```
http://www.website.com?wt_mc=Newsletter_2010_08&wt_cc1=link1
```

As an example of page configuration, we have set the value "personalized" here to identify a personalized campaign.


```
wt.customCampaignParameter = {  
  1: "personalized"  
};
```

Campaign parameters of the type "Text" are shown in the Webtrekk tool under "E-Commerce > Campaign Parameters". Campaign parameters of the type "Number" are available as a metric in the analyses.

For further options, see the examples.

7.5 Categories

Categories serve to group various objects of a type. Unlike custom parameters, category assignment must be unique. In this way, you can aggregate pages in content groups. For example, you could define which page is assigned to which main category in a content group called "Main category" (content group with index 1).

```
wt.contentGroup = {  
  1: "woman" // main category  
};
```

Categories can be set as text or a number. The data type "text" is used for categorisation while the data type "number" is used to evaluate the number of times an object is called with the value stored for it.

Categories must be configured in the Webtrekk tool so that they can be tracked. This is done under "Configuration > Categories". The ID and data type (text/number) are defined for each category.

Categories can be transferred in the pixel or an import via Excel or SOAP interface (only time categories cannot be transmitted in the pixel).

Category parameters should not exceed a length of 255 characters. All additional characters will be deleted.

Webtrekk distinguishes five category types:

- Content Groups (page categories)
- Produkt Categories
- Media Categories
- Time Categories
- Campaign Categories

7.5.1 Time Categories

Time categories are used in Webtrekk to import data assigned to a certain time.

You can categorise the time as a particular hour in the Webtrekk tool. As data type "text", for example, you could categorise every day based on whether a newsletter was sent to it or not. The data type "number" could be chosen to store hourly exchange rates. You can perform calculations in the Webtrekk tool with the data type "number".

Webtrekk recommends that time category data is transmitted via SOAP interface. Since the number of uploads per day is limited, data should be sent as a package. Alternatively, the import can be in an Excel file. To ensure that the correct format is used, an export of existing time categories should be called up as a template (Configuration > Categories > Time Categories).

Time (YYY-MM-DD HH)	Category (number) - Exchange Rate
2010-09-01-00	1.27
2010-09-01-01	1.26
2010-09-01-02	1.27

7.5.2 Campaign Categories

Campaign categories allow evaluation of advertising media as a group.

Campaign categories can be set as text or a number. In the process, the "text" data type can be used to map the advertising media hierarchy. Campaign categories of the data type "number" can be used to evaluate every advertising media call with the value stored for it.

Campaign categories are configured exclusively in the Webtrekk tool. They are not configured in the pixel. The configuration is carried out under "Configuration > Marketing Configuration". The ID and data type (text/number) are defined for each campaign category.

A list of measured campaign categories of the data type "text" can be called up in the Webtrekk tool under "Marketing > Campaigns > Campaign Categories". Campaign categories of the data type "number" are available as a metric in the analyses.

7.6 Custom Parameters

You can use custom parameters (depending on your service level) to enrich analytical data with your own website-specific information and/or metrics. Observe the syntax guidelines when defining parameters.

Custom parameters must be set up in the configuration (Configuration > Custom Parameters) before they can be entered. The ID and data type (text/number) are defined for each parameter during setup, among other things.

Webtrekk distinguishes six custom parameter types:

- Page Parameters
- E-commerce Parameters
- Event Parameters
- Campaign Parameters
- Session Parameters
- Independent Parameters

7.6.1 Session Parameters

Session parameters always refer to a session, in other words, a visit. If the value for the parameter is transmitted during a visit several times, only the last value is evaluated.

For example, the status indicating whether a user was logged in during the visit is transmitted. By default, each visit at the start of a session parameter is indicated as "not logged in". A login is transmitted by the same parameter and thus overwrites the first value.

In contrast to a page parameter, it is not possible to evaluate the page that a session parameter was set on. In addition, a page parameter allows evaluation of any value set during a visit.

```
wt.customSessionParameter = {  
  1: "logged_in"  
};
```

For further options, see the examples.

7.6.2 Predefined session parameters

You can enrich the analytics data with your website-specific information and/or metrics using "predefined session parameters".

Predefined session parameters need to be activated and set up in the configuration (Configuration > Custom Parameters > Session Parameters/Configuration > Categorisation > Product Categories) before they are captured.

For example, the status indicating whether a user was logged in during the visit is transmitted. By default, every visit would be marked as "not logged in" in a session parameter at the beginning. A login is transferred to the same parameter and thereby overwrites the initial value.

```
wt.loginStatus = "logged_in";
```

7.6.3 Independent Parameters

Independent parameters can only be linked with other data in terms of time. Consequently, there is no direct link to a page, click or similar. The measurement does not generate a visit in Webtrekk.

This parameter can be used to count the number of times a newsletter is opened, for example, by embedding a specific pixel in it.

```

```

Note: Independent parameters only include the time of entry as the analysis reference.

For further options, see the examples.

7.6.4 Tracking Downloads

If automatic event tracking (setting "link" or "standard" in the "linkTrack" parameter) is active, the files available in the browser for downloading can be automatically tracked.

To do this, the applicable file types must be entered, separated by a semicolon, in the "wt.linkTrackDownloads" parameter.

Note: With these clicks the pixel is sent immediately and not when the page is unloaded.

```
var webtrekk = {  
  linkTrack: "link"  
};  
  
var wt = new webtrekkV3(webtrekk);  
wt.linkTrackDownloads = "pdf;doc";
```

Tracking with function call

Alternatively, downloads of files such as Excel, PowerPoint, PDF etc. can be tracked by activating event tracking with a function call in the "without leaving the page" variant.

```
<a href="http://www.webseite.com/download.pdf" onmousedown="wt.sendinfo({linkId: 'aktionsname'})";  
target="_blank">Link to download.pdf</a>
```

7.7 Heatmap Tracking

Heatmaps show you the places on your website which received a large number of clicks, regardless of whether there is actually a link at these positions. To produce a Website Heatmap analysis (depending on your service level), you must activate the heatmap parameter.

```
var webtrekk = {  
    heatmap: "1"  
};  
  
var wt = new webtrekkV3(webtrekk);
```

If you have activated the heatmap for a pixel instance, you can deactivate or reactivate the heatmap again using "wt.heatmap = "0"" or "wt.heatmap = "1"" respectively. This option is helpful if you provide a layer or similar via the content area, but do not wish to record any heatmap data for this.

A point of reference must be defined if you work with centred websites. The point of reference tells the pixel the position (x-axis and y-axis) at which the content scope begins. To define the point of reference the "wt_refpoint" ID must be assigned to a random HTML tag (e.g. DIV, SPAN, TABLE, IMG, etc.). The point of reference should be placed exactly in the upper left-hand corner of the website content. The element with the "wt_refpoint" ID cannot be invisible (display:none), otherwise the coordinates of this element cannot be read.

```
<table align="center" id="wt_refpoint">  
    <tr>  
        <td>The centred website content is located here.</td>  
    </tr>  
</table>  
  
or  
  
<div style="text-align:center; width:500px;">  
    <span id="wt_refpoint"></span>  
    The centred website content is located here.  
</div>
```

If your website already contains an element with an ID that can be used as a reference point for the heatmap, you do not need to embed an additional element with the "wt_refpoint" ID. In this case you can assign the ID to the wt.heatmapRefpoint variable in your Webtrekk configuration.

```
var webtrekk = {  
    heatmap: "1"  
};  
  
var wt = new webtrekkV3(webtrekk);  
wt.heatmapRefpoint = "content";
```

```
<table align="center" id="content">
  <tr>
    <td>The centred website content is located here.</td>
  </tr>
</table>

or

<div id="content" style="text-align:center; width:500px;">
  The centred website content is located here.
</div>
```

Note regarding the browser Opera: If the heatmap is activated, the mouse pointer appears to flicker when clicking the page. The reason for this is that the mouse pointer icon changes (to an hourglass) when a request is sent.

Opera clicks will no longer be recorded by heatmap if the following configuration setting is added:
heatmap = (window.opera)?"0":"1";

The heatmap is called up in the Webtrekk tool under "Overlay > Websites Heatmap".

For the first function tests with only a few clicks, select the display type "Standard (detail)" in the Webtrekk tool, as a sufficient amount of data must be collected first of all for aggregated portrayal.

7.8 Internal Search

Analyse search terms used by visitors to your website by including it in tracking.

The GET or POST methods are used to realise a website search function. The POST method requires the "internalSearch" parameter. Dynamically enter the search term in the configuration parameter. Alternatively, the internal search function can be set with the GET method in the Webtrekk system configuration (Configuration > System Configuration) under "Internal search parameter".

```
wt.internalSearch = "adidas shoes";
```

A list of tracked internal search phrases is called up in the tool under "Marketing > Search Phrases > Internal Search Phrases".

GET and POST method can both be used for the same account. But for sending an internal search phrase only one method should be used (not both for the same search phrase at the same time).

7.9 Form Tracking

Form tracking can track all manners of form input. A list of measured forms can be called up in the Webtrekk tool under "Navigation > Forms".

To activate global form tracking, set the "form" variable to "1" in the tracking script global configuration. In this case, every page of the first located form – if there is one – will be marked with the name tag "wt_form" for form tracking.

```
var pageconfig = {  
  trackId: "111111111111111",  
  trackDomain: "track.wt-eu02.net",  
  domain: www.website.com,  
  form: "1"  
};
```

Marking within a form

```
<input type="hidden" name="wt_form" value="1">
```

If form tracking should apply to single pages only, activate the function by passing the corresponding form object directly in the page configuration. In this case, the form does require marking with the "wt_form" element.

```
var wt = new webtrekkV3();  
wt.contentId = "contact";  
wt.formTrackInstall(document.forms[0]);  
wt.sendinfo();
```

The example activates form tracking for the first form on the page.

JavaScript onSubmit handler

The tracking script will be unable to automatically identify your form if it calls up a separate function via the onSubmit handler that then sends the form with form.submit() rather than using a submit button. The reason for this is that the onSubmit handler is not executed if form.submit() is called. In this case, simply set the corresponding pixel instance to "true" in your function's "formSubmit" variable.

```
function mySubmit(formObject) {  
  /* Your Code */  
  wt.formSubmit = true;  
  formObject.submit();  
}
```

Naming forms

By default, your form's "name" attribute is used to generate its name. With the parameter "wt.formAttribute" you can define another attribute (e.g. the "id" attribute) as the form identifier.

If there is no "name" attribute in the form header and "wt.formAttribute" is not defined, the page title "wt.contentId" is used as the identifier.

Selecting form fields

If only certain form fields in a form should be counted, define the affected fields in a semicolon-separated list. To do this, a "hidden" field will be defined in the tracked form:

```
<input type="hidden" name="wt_fields" value="name;location">
```

In the example, only the "name" and "location" fields are counted.

Supported fields and values

All form fields are measured, with the exception of the "hidden", "button", "image", "reset" and "submit" field types. With the form field types "select-multiple", "select-one", "checkbox" and "radio", the precise values can be transmitted. With all other form field types, only the field status is sent to Webtrekk, i.e. only "filled_out" is sent for a completed form field and not the content.

If, however, you would like to evaluate the form field content as clear text, add the corresponding form fields to the configuration parameter "wt.formFullContent". The first 30 characters in each field will be transmitted. In the following example, the form fields "post code" and "age" will be sent as plain text as opposed to "filled_out". If the form field is empty, "empty" is transmitted.

```
wt.formFullContent = "postcode;age";
```

For form fields of the type "radio," "select-one," "select-multiple" and "checkbox", the value of the "value variable" will be sent to Webtrekk by default. Optionally, an alternative value attribute could be used. If the alternative attribute is set, the value of all of the above-mentioned field types in the alternative attribute will be sent to Webtrekk. This is helpful if you are only sending an ID as the standard value, and this would not be understandable for your web analysts when using the Webtrekk tool. In the following example, the content of the attribute "rel" is sent and not the actual value.

```
wt.formValueAttribute = "rel";
```

If you would like to track forms that have been pre-filled with a default value, you must inform the pixel of which form fields are concerned. This is done with the parameter "formFieldDefaultValue". This includes an object with a mapping between form field names and form field default values.

```
wt.formFieldDefaultValue = {  
  "firstname": "Firstname",  
  "lastname": "Lastname",  
  "email": "E-Mail"  
};
```


Path-Analysis for form-field Focus Using the path-view you can display the timely order of objects, in this case form-field objects. This enables you to analyze the order in which form-fields were selected. form-fields will be tracked more than once if they were selected multiple times. If you want to enable this feature, please set the according parameter "formPathAnalysis". You can either set the parameter globally, for all forms, or individually for specific sites.

```
var webtrekkConfig = {
  trackId: "1111111111111111",
  trackDomain: "track.wt-eu02.net",
  domain: "www.website.com",
  contentId: "",
  formPathAnalysis: true
};

// or

var webtrekk = {
  contentId: ""
};

var wt = new webtrekkV3(webtrekk);
wt.formPathAnalysis = true;
wt.sendinfo();
```

Due to the possibility of sending form-fields multiple times the quantity of (not) filled form-fields can be higher than the number of (not) send form-fields. In addition, you might face difficulty deciding 100% if a field was filled or not since both could exist.

Sending form data and content Form data is sent to Webtrekk on leaving the form page, regardless of whether the form was actually sent or the page was left without sending it. No action is taken if the browser is closed without sending the form.

Notice: Radio buttons with the same "name" attribute content are viewed and submitted as a form field by form tracking. The status "empty" is sent if no radio button was clicked, or "filled_out" or the form field value of the clicked radio button.

If you want to capture every individual radio button in a group, use an alternative form field attribute.

Please note that requests for click and form tracking are not sent until the user leaves a page. If the user is on an unsecure page, and the next page is an SSL page, an SSL message may be displayed. To avoid this, non-SSL pages with activated action tracking, which may follow an SSL page, must be configured with the setting wt.forceHTTPS="1".

Anonymising form data and content

If required, you can anonymise all form data – for data protection reasons, for example – before sending it to Webtrekk. This means content from the form fields will not be transmitted to Webtrekk. The Webtrekk analysis screen will therefore only show which form fields were completed and which were not.

```
wt.formAnonymous = "1";
```

Please note that the parameter `wt.formFullContent` always delivers the content of the form fields, even if the form has been anonymised!

Application example

```
wt.formAnonymous="1";
wt.formFullContent = "textarea";
wt.formAttribute = "id";
wt.formValueAttribute = "rel";
wt.formTrackInstall(document.forms[0]);
```

```
<form id="formular-name" method="post" action="form.php">
  <input name="textfield" type="text" size="30"></br>
  <input name="password" type="password" size="30"></br>
  <textarea name="textarea" cols="50" rows="10"></textarea></br>
  <select name="select" size="5" multiple>
    <option rel="select1" value="1">Select 1</option>
    <option rel="select2" value="2">Select 2</option>
  </select></br>
  <input type="submit" value="submit">
</form>
```

```
fn=formular-name|1           // [ Form name | Form was sent ]
ft= textfield.text|filled_out|0; // [ Fieldname.Fieldtype | field content | last focus ]
password.password|filled_out|0; // [ Fieldname.Fieldtype | field content | last focus ]
textarea.textarea|test|1;     // [ Fieldname.Fieldtype | field content | last focus ]
select.select-multiple|select2|0 // [ Fieldname.Fieldtype | field content | last focus ]
```

Tracking several forms on one page

As mentioned above, only one form can be tracked per pixel instance and page. To measure two forms on one page, you will need to initialise a separate pixel instance for the second form. This second instance will be exclusively used to measure the second form and will not collect any further data since this will already be collected by the first instance. This prevents double data capture in the tool.

```
var wt = new webtrekkV3();
wt.contentId = "contact";

// ...

wt.formTrackInstall(document.forms[0]);
wt.sendinfo();

var wt2 = new webtrekkV3();           // generate second instance
wt2.contentId = "contact";             // set the same content ID as in the first instance
wt2.formTrackInstall(document.forms[0]); // activate form tracking
```

Manual update of form fields

If the content of a form field changes or a focus on a form field is simulated, call the following method and pass the changed form field.

```
wt.updateFormFieldStatus (document.getElementById('firstname'));
```

Forms without <form>

You can create a custom form and use it for tracking, if you are using form on your page that are not created by using the html tag "<form>". Create a instance of the class "CustomForm" and pass form name and all belonging form fields which should be tracked. Pass the custom form to the tracking pixel to track it.

```
<div id="form-name">
  <input id="textfield" name="textfield" type="text" size="30"></br>
  <input id="password" name="password" type="password" size="30"></br>
  <textarea id="textarea" name="textarea" cols="50" rows="10"></textarea></br>
  <select id="select" name="select" size="5" multiple>
    <option rel="select1" value="1">Select 1</option>
    <option rel="select2" value="2">Select 2</option>
  </select></br>
  <input id="submit" type="submit" value="submit">
</div>
```

```
var wt = new webtrekkV3();
wt.contentId = "contact";

// ...

var customForm = new wt.CustomForm('form-name', [
  document.getElementById('textfield'),
  document.getElementById('password'),
  document.getElementById('textarea'),
  document.getElementById('select'),
  document.getElementById('submit')
]);
wt.formTrackInstall(customForm);
```

If, for example, you send your form with Ajax, meaning you do not leave the page, you should send the form request manually. For this purpose, use the function "sendFormRequest".

```
function myCustomFormSubmit(formObject) {
  /* Your Code */
  wt.formSubmit = true;
  wt.sendFormRequest();
  /* Your Code */
}
```

7.10 Custom Visitor IDs

To improve visitor identification, you can use custom visitor IDs instead of Webtrekk's long-term cookies ("eid"). The reason for this is that some users or programmes automatically delete long-term cookies once a session (visit) ends. Without custom visitor IDs, repeat visitors would not be identified.

To use custom visitor IDs, include a unique identifier from your shop/CMS system with the tracking pixel. If you do not use unique visitor IDs on your website, you can use the visitor's e-mail address as an alternative unique identifier. In this case, you should encrypt the address to comply with data protection requirements and ensure the e-mail address is unreadable (e.g. using the MD5 hash).

Visitor IDs can be passed to the pixel following a successful login or completed order, for example. These custom visitor IDs can be evaluated in the Webtrekk tool under "Visitors > Visitors > Last Custom Visitor IDs".

```
wt.customerId = "372d1a04d003eebc09e17330d5d3117c";
```

7.10.1 URM category

With the optional parameter "urmCategory", you can also categorise the visitors.

These URM categories must be created in the tool first of all.

In the example below, the family status is assigned to the visitor.

```
wt.urmCategory = {  
  2: "single"  
};
```

7.10.2 Predefined URM categories

With "predefined URM categories", you can also categorise the visitor.

Predefined URM categories have to be activated and set up in the configuration (Configuration > Categorisation > URM Categories) before they are captured.

```
wt.email = "info@webtrekk.com";      // _@_.  
wt.emailRID = "Receiver";           // string  
wt.emailOptin = "3";                // [1 yes | 2 no | 3 unknown]  
wt.firstName = "";                  // string  
wt.lastName = "";                   // string  
wt.telefon = "4930755415101";        // string [0-9]  
wt.gender = "3";                    // [1 m | 2 w | 3 unknown]  
wt.country = "Germany";             // string  
wt.city = "Berlin";                 // string  
wt.postalCode = "10115";             // string  
wt.street = "Robert-Koch-Platz";     // string  
wt.streetNumber = "4";              // string  
wt.validation = "1";                // [1 validation | 0 without validation]
```

```
wt.birthday = "19900215";    // [ JJJJMMDD ]  
  
// or  
  
wt.birthdayJ = "1990";      // [ JJJJ ]  
wt.birthdayM = "02";        // [ MM ]  
wt.birthdayD = "15";        // [ DD ]
```

7.11 Media Tracking

With media tracking you can track the use of media streams (depends on your service package). The recorded functions are open, play, pause and stop (depending on the embedded medium), play duration, bandwidth, volume and mute setting. Webtrekk supports flash and Silverlight players as well as Windows Media Player, Apple QuickTime and RealPlayer.

For this analysis, a separate media tracking pixel is required for additional embedding with the normal tracking pixel or as a standalone on external sites. In the second case, the parameters "wt.trackDomain" and "wt.trackId" from the tracking pixel configuration will be required in addition to the following configuration examples.

Due to the low amount of direct embedding using Windows MediaPlayer, Apple QuickTime or RealPlayer, we refer solely to the integration of Flash players in this guide. If you use Windows Media Player, Apple QuickTime or RealPlayer on your website, our support will be happy to help you with the pixel.

7.11.1 Embedding in Flash

When embedding media tracking in Flash films, the media pixel is embedded with it. This is done with the "getURL" function in AS2 and with "navigateToURL" in AS3.

If you require this version of media tracking, we would be happy to provide you with a Flash demo, which will help you to understand integration and function calls.

To set the track domain and track ID use the JavaScript function "wt_init_media", which must be called up first when launching a Flash film (see the "initMediaPixel" function in the ActionScript).

To track a player action use the JavaScript function "wt_sendinfo_media".

```
/**
 * @param {String} media_id media ID
 * @param {String} player_action media action [init|play|pos|pause|stop|eof|seek]
 * @param {String} clip_position clip position in seconds
 * @param {String} clip_length clip length in seconds
 * @param {String} [media_group] media categories or custom parameter
 * @param {String} [bandwidth] bandwidth in bit/second
 * @param {String} [volume] volume (0-255 or 0-100)
 * @param {String} [mute] mute (1=sound off, 0=sound on)
 */
wt_sendinfo_media(media_id, player_action, clip_position, clip_length, media_group, bandwidth,
volume, mute)
```

Sending player actions:

```
ExternalInterface.call("wt_sendinfo_media", "homepage_film_1", "play", "0", "30",
"mg1=homepage_filme", "64000", "100", "0");
```

The `player_action` parameter contains the media player action. The "init," "play," "pause," "stop," "eof" and "pos" commands are supported.

The "init" command must be sent as soon as the Flash film is loaded. "play," "pause" and "stop" refer to the corresponding controls; "eof" (end of file) is sent at the end of the film. The "pos" command, for example, can be sent every 10 seconds as a "keepalive". This command does not appear in the evaluation but serves to identify the last seen position in the film if the video is ended by closing the browser window, preventing a "stop" or "eof" command from being sent.

The `clip_position` contains the position within the film at the time when the action is sent (displayed in seconds).

The `clip_length` parameter contains the total length of the film in seconds.

Optional parameters

In the same way that pages are grouped into categories as content groups, media can be grouped into media categories. Media Group (`media_group`) is used to group several videos and is entered in the following format: "mg1=mygroup1" gives group 1 the value "mygroup1". Separate several groups with semicolons. Media categories must be configured in the Webtrekk tool so that they can be tracked. This is completed under "Configuration > Categories > Media Categories".

Bandwidth is indicated in bits per second.

Volume ranges from 0 (mute) to 255 (maximum volume). The evaluation shows the last value set by the user.

Mute is passed with the value 1 (sound off) or 0 (sound on).

You can also pass custom commands for evaluation with the current play time. For example, they can be used to mark ads appearing in the film or to indicate when the video's HD version is selected. They are called up in the same way as in the example above, e.g. `player_action` can be "start advertising," "stop advertising" or "switch to hd version".

7.11.2 Embedding in Silverlight

A library is available to simplify embedding of media tracking in Silverlight.

If you require this version of media tracking, we would be happy to provide you with the Silverlight library and embedding instructions, which will help you to understand integration and function calls.

8 Additional Functionality

8.1 Adobe Flash

To track website content created with Macromedia Flash (any version), the tracking pixel must be adapted. If it is a pure Flash page, we recommend commenting out the function call for the tracking pixel ("wt.sendinfo()") with "//". This prevents tracking of the embedded HTML page.

```
// ...  
// wt.sendinfo();  
// ...
```

Calling the tracking pixel via a Flash film requires that the pixel is positioned in the same HTML page as the embedded Flash film.

The function call for the tracking pixel is initiated via the Flash film.

```
ActionScript 2:  
  
getURL("javascript:wt.sendinfo('homepage')");
```

```
ActionScript 3:  
  
ExternalInterface.call("wt_sendinfo", "homepage");
```

The "homepage" in this Flash code is the page name.

Please note that a differentiation should be made between page calls (page impressions) and "clicks" in Flash films, as well.

Note: The Flash film and/or action script must be modified if two "getURL" commands should be executed in sequence in a Flash film frame (e.g. for flash ads). Some browsers (e.g. Internet Explorer) do not execute the second "getURL" command.

The following action script generates the "homepageteaser" action when the Flash element is clicked and links to the target page "link_target.htm":

```
on (release) {
    getURL("javascript:wt_sendinfo ('homepageteaser','link')");

    var delay_mc = _root.createEmptyMovieClip("delay", 1);
    delay_mc.onEnterFrame = function() {
        getURL("link_target.htm");
        delete this.onEnterFrame;
        this.removeMovieClip();
    }
}
```

To display the Flash film in the website overlay and the heatmap in this special case, the film's HTML embedding needs to be modified. The tracked Flash film must contain a dummy link and use the "opaque" setting:

```
<a href="#" name="homepageteaser">
  <object ...>
    <param name="wmode" value="opaque" ...>
    <embed wmode="opaque" ...></embed>
  </object>
</a>
```

8.2 Ajax

When tracking website content loaded by Ajax, the tracking pixel must be sent manually whenever an event is triggered. When sending the tracking pixel, you generate a page or action request, to which you can add any pixel parameters. To do this you should use the function call "wt.sendinfo".

The tracking pixel is called as follows:

```
// Send a page call (Content-ID):
wt.sendinfo({contentId: "pagenname"});
```



```
// Send a page call (place products in shopping cart):
```

```
wt.sendinfo({  
  contentId: "pagename",  
  contentGroup: {  
    1: "level 1"  
  },  
  product: "productname",  
  productStatus: "add",  
  customEcommerceParameter: {  
    1: "yellow",  
    2: "XL"  
  }  
});
```

```
// Send an action (teaser click with position):
```

```
wt.sendinfo({  
  linkId: "actionname",  
  customClickParameter: {  
    1: "teaser",  
    2: "top_position"  
  }  
});
```

Note: Any kind of pixel parameters can be passed to the function calls. The parameter names correspond with those of the script parameters. Please review the parameters in the respective chapter of this document for details.

8.2.1 Event Tracking

The links contained in page content reloaded by Ajax (e.g. when showing layers) can also be marked for automatic event tracking. To do this, call the "linkTrackInit" method as soon as new content is reloaded.

```
wt.linkTrackInit();
```

8.2.2 Form Tracking

Forms reloaded by Ajax (e.g. in a layer) that you want to track must be marked for form tracking. To do this, call the "formTrackInstall" method in the tracking script.

```
wt.formTrackInstall();
```

In this example, the first form on the page is tracked using an existing form element called "wt_form".

If the reloaded form is not marked in this way, you can also pass the form object directly to the method.

```
wt.formTrackInstall(document.getElementById('FORMID'));
```

If, for example, you send your form with Ajax, meaning you do not leave the page, you should send the form request manually. For this purpose, use the function "sendFormRequest".

```
function myAjaxSubmit(formObject) {  
    /* Your Code */  
    wt.formSubmit = true;  
    wt.sendFormRequest();  
    /* Your Code */  
}
```

If the content of a form field changes or a focus on a form field is simulated, call the following method and pass the changed form field.

```
wt.updateFormFieldStatus(document.getElementById('firstname'));
```

8.3 Server-to-Server Communication

The "csid" or "ceid" parameter must be added to custom session and/or Ever-IDs when making a server-to-server communication request; if this is not done, the user cannot be identified. This parameter overwrites the session and Ever-IDs automatically generated by Webtrekk and added to the Webtrekk pixel. The "csid" and "ceid" parameters can contain any characters (max. 255) and should be transmitted with URL-encoding.

Since geo-analyses are based on the IP address, the client IP should be transmitted with the "X-WT-IP" parameter in the http header or URL. The http header is recommended.

8.4 Encrypted Configuration

Some parameters and order information can be encrypted. This prevents the values from being shown in plain text when the source code is viewed or during data transmission.

8.4.1 On the Client Side

Some parameters and order information are encrypted on the client side. Encrypted values are displayed as Unicode character numbers in hex format. In this notation, small "a" becomes "61" and small "ü" "c3bc", etc.

The configuration settings are made via the corresponding page pixel.

Encryption should be tested thoroughly, in order to avoid incorrectly decrypted values. When activating encryption, all details need to be encrypted. If this is not the case, inadvertent values may be generated during decryption.

To encrypt custom parameters, enter the "custom" value in the "wt.secureConfig" parameter.

The value "order" should be set in order to encrypt order data. The following configuration sets a custom parameter in the page content to "green".

```
wt.secureConfig = "custom;order"; // secure config

wt.customParameter = {
  5: "677265656e"
};
```

8.4.2 On the Server Side

Webtrekk offers clients the possibility of encrypting parameter values passed in the pixel request. Symmetric encryption methods are used for this purpose. Encryption of the corresponding values must be carried out on the client side. Webtrekk decrypts the encrypted values using a key provided with the tool for each individual account.

The following algorithms are provided in the tool for selection:

- BLOWFISH128
- AES128
- RC4128

The key that is used has a length of 128 bits. The algorithms "Blowfish" and "AES" are operated in CBC mode with PKCS5 padding. This means that in addition to the key stored in the tool, a dynamic initialisation vector is also used during encryption and decryption. This encrypted value must be placed behind this initialisation vector. The length of the initialisation vector is 8 bytes with "Blowfish" and 16 bytes with "AES".

If you have any further questions, please contact your Key Account Manager or send an e-mail to support@webtrekk.com.

8.5 Processes

Processes in Webtrekk Q3 provide the ability to perform funnel analyses for certain click paths. That means you can analyse certain page sequences, which a visitor must run through to fulfil a specific objective on your website (e.g. an order in your online shop or registration on a content page). These processes enable you to obtain a clear overview of critical areas, such as common exit points, thereby allowing you to identify optimisation potential quickly and simply.

No pixel adjustments are needed in order to analyse processes. The configuration is completed via the Webtrekk interface (Configuration > Processes). However, it is important that the process steps are defined in line with individual page calls. In other words, a unique content ID must exist for each process step (e.g. checkout_step_1, checkout_step_2 und not just one checkout).

This may require an optimisation of pixel integration.

8.6 Cross-Channel Tracking

Webtrekk provides the ability to recognise a user that has visited multiple websites or utilised multiple (mobile) devices. Two options are provided by Webtrekk to achieve this, both of which are explained in detail below.

Note: if this feature is not used correctly, it can happen that multiple users are assigned the same Ever ID or Customer ID. This would have a significant impact on the quality of your data. Please therefore ensure you use this feature correctly!

8.6.1 Recognition via Webtrekk Ever ID

In order to recognise a user based on the Ever ID, it must be transmitted from one page to another. To do this you will need to add two additional URL parameters to the link to the other website. First of all the URL parameter "wt_eid", which includes the Ever ID, and also the parameter "wt_t", which contains a timestamp. This is needed in order to ensure the Ever ID is valid for no longer than 15 minutes in the URL. This, in turn, will minimise the probability of such a link being posted and, for example, evaluated multiple times with the same ID (otherwise, all of the visitors would receive the same Ever ID and, therefore, be counted as visitor 1!).

```
http://new.domain.com/start.html?wt_eid=2135817235100536326&wt_t=1358414378580  
or  
http://new.domain.com/start.html#wt_eid=2135817235100536326&wt_t=1358414378580
```

If the user clicks the link, they will be taken to the new page where they will receive the same Ever ID that they had on the previous page.

8.6.2 Recognition via Customer ID

In order to recognise a user based on the Customer ID, it must be transmitted from one page to another. To do this you will need to add two additional URL parameters to the link to the other website. First of all the URL parameter "wt_cd", which includes the Customer ID, and also the parameter "wt_t", which contains a timestamp. This is needed in order to ensure the Customer ID is valid for no longer than 15 minutes in the URL. This, in turn, will minimise the probability of such a link being posted and, for example, evaluated multiple times with the same ID (otherwise, all of the visitors would receive the same Ever ID and, therefore, be counted as visitor 1!).

```
http://new.domain.com/start.html?wt_cd=213581723100536326&wt_t=1358414378580  
or  
http://new.domain.com/start.html#wt_cd=213581723100536326&wt_t=1358414378580
```

If the user clicks the link, they will be taken to the new page where they will receive the same Ever ID that they had on the previous page.

8.6.3 Recognition via Mobile Apps

To recognise users based on the utilisation of applications, the same logic can be used as explained. In addition to this, however, you must provide your own referrer, to ensure the visit to your website is not reported as direct access. This does not need to be considered when switching between two websites.

To simulate your own referrer, simply add the additional URL parameter "wt_ref" to the link URL. The parameter "wt_t", which is valid for a period of 15 minutes, is also used here again.

Please note that the referrer URL must be encoded.

```
http://new.domain.com/start.html?  
wt_eid=2135817235100536326&wt_ref=http%3A%2F%2Fwww.webtrekk.com%2Fen%2Fhome.html&wt_t=1358414378580  
or  
http://new.domain.com/start.html#  
wt_eid=2135817235100536326&wt_ref=http%3A%2F%2Fwww.webtrekk.com%2Fen%2Fhome.html&wt_t=1358414378580
```

Alternative, you can transfer the Ever ID and referrer URL directly to the pixel.

```
// set Ever ID for the current TrackId  
wt.setEverId("2135817235100536326");  
  
// set Ever ID for the TrackId "123451234512345"  
wt.setEverId("2135817235100536326", "123451234512345");
```

```
wt.setReferrer("http://www.webtrekk.com/en/home.html");
```

9 Data Protection Opt-Out

Website visitors can permanently opt out of having their data stored anonymously. To guarantee this right, a cookie called "webtrekkOptOut" must be set. Whenever the Webtrekk track server receives this cookie with a track request, the request is rejected immediately. In this case, our tracking servers return a header notification containing the "X-WT-OPTOUT" parameter with the value "true".

The opt-out cookie must always be set in the webtrekk.net track domain. If first-party cookies are used, an additional opt-out cookie must be set in the actual domain. Webtrekk recommends the use of 1st party cookies.

The following script on your page sets the opt-out cookie in the webtrekk.net domain:

```
https://<<TRACKDOMAIN>>/<<TRACKID>>/optout?redirect=https%3A%2F%2Fwww.webtrekk.com%2Fopt-out-  
success%2F
```

The parameter "optout" can be used to enter a redirect URL, which the user will be redirected to after the cookie has been set.

If you use first-party cookies, you can set the additional opt-out cookie in your domain with this JavaScript call:

```
wt.setCookie('webtrekkOptOut', 1, 60*60*24*30*12*10);
```

In this example, the opt-out cookie has a life of 10 years.

If you do not wish to name the Opt-Out cookie "webtrekkOptOut", in order to comply with data protection requirements, you can use the variable "optoutName" to define an alternative name for the Opt-Out cookie.

```
wt.optoutName = "trackingOptOut";
```

The call used for setting the Opt-Out cookie would then need to be changed as follows:

```
wt.setCookie('trackingOptOut', 1, 60*60*24*30*12*5);
```

In this example the Opt-Out cookie has a lifetime of 5 years.

Note: this rejection regarding the storage of data remains in place as long as the opt-out cookie is not deleted.

10 Checking Pixel Configuration

The implementation phase can be simplified if you check immediately which parameters are going to be sent. This allows you to avoid waiting until your account is updated again.

You can use any http client to view transmitted pixel requests.

Webtrekk recommends the following tools:

- Firefox: the plugin [HttpFox](#) (OS-independent)
- Internet Explorer, Firefox, Opera, Google Chrome, Safari: [Fiddler2](#) (only for Windows)
- Browser- and OS-independent: [Wireshark](#)

Below you will find an overview of all parameters, which can be sent to Webtrekk based on the respective pixel configuration. Those parameters that are not listed below are internal Webtrekk parameters, which may not be adjusted.

Optionale no-script parameter	Javascript parameter	Description
fn	- is read automatically -	Form name
ft	- is read automatically -	Form fields
eid	- is read automatically -	Webtrekk EverId
tb	- is read automatically -	Marking the tabbed browsing request
la	- is read automatically -	Country code for client language setting (such as 'de')
wt_vt	- is read automatically -	Name of the AB test
cdb	- is read automatically -	Cross-device bridge parameter [cdb2, cdb3, ...]
ct	wt.linkId	Click or link name
cp	wt.customParameter	Page parameter [cp2, cp3, ...]
cp770	- is read automatically -	Marking the tabbed browsing request
cp771	wt.numberSearchResults	Number of search results
cp772	wt.errorMessagees	Error messages
cp773	wt.paywall	Paywall calls
cp774	wt.articleTitle	Article header
cp775	wt.contentTags	Content tags

Optionale no-script parameter	Javascript parameter	Description
cp776	wt.pageTitle	Page title
cp777	wt.pageType	Page type
cp778	wt.pageLength	Page length
cp779	wt.daysSincePublication	Days since publication
cs	wt.customSessionParameter	Session parameter [cs2, cs3, ...]
cs800	wt.loginStatus	Login status
cs801	- is read automatically -	Pixel version
cs802	- is read automatically -	Tracking platform
ce	wt.customTimeParameter	Custom parameter [ce2, ce3, ...]
cb	wt.customEcommerceParameter	E-commerce parameter [cb2, cb3, ...]
cb563	wt.couponValue	Coupon value
cb760	wt.productSoldOut	Product sold out
cb761	wt.paymentMethod	Payment method
cb762	wt.shippingService	Shipping service provider
cb763	wt.shippingSpeed	Shipping speed
cb764	wt.shippingCosts	Shipping costs
cb765	wt.grossMargin	Margin/mark-up
cb766	wt.orderStatus	Order status
cb767	wt.productVariant	Product variant
cd	wt.customerId	Customer ID
uc	wt.urmCategory	Customer parameter [uc2, uc3, ...]
uc701	wt.emailRID	E-mail receiver ID
uc702	wt.emailOptin	E-mail opt-in [1 = yes, 2 = no]
uc706	wt.gender	Gender [1 = male, 2 = female]
uc707	wt.birthday	Date of birth [YYYYMMDD]
ov	wt.orderValue	Order value
oi	wt.orderId	Order ID
ba	wt.product	Products in shopping basket
co	wt.productCost	Product cost
cr	wt.currency	Product currency
qn	wt.productQuantity	Number of products

Optionale no-script parameter	Javascript parameter	Description
ca	wt.productCategory	Product category [ca2, ca3, ...]
st	wt.productStatus	Status of the shopping basket [add, conf, view]
cg	wt.contentGroup	Page category [cg2, cg3, ...]
is	wt.internalSearch	Search term of the internal search function
mc	wt.campaignId	Campaign ID consisting of media code parameter and value ("wt_mc=newsletter")
mca	wt.campaignAction	Campaign action [c = click, v = view]
ck	wt.customClickParameter	Action parameter [ck2, ck3, ...]
cc	wt.customCampaignParameter	Campaign parameter [cc2, cc3, ...]

If you use HttpFox, you can filter all requests that are sent based on their tracking domain. This means only information will be displayed that has been sent to Webtrekk.

The screenshot shows the HttpFox application interface. The top panel displays a list of HTTP requests with columns: Gestartet, Dauer, Gesendet, Empfangen, Metho..., Resultat, Typ, and URL. The bottom panel shows the 'Query String' tab for a selected request, displaying a table of parameters and their values.

Parameter	Wert
p	315,manieren_per_mausklick,1,1280x1024,24,1,1322819407314,0,1280x872,1
tz	1
eid	2132065400400287276
one	0
fns	1
la	de
cg1	home
cp5	7654
cp6	645
cp7	4601
cp8	9718
eor	1

11 Examples

Implementation and evaluation of custom parameters is explained using the examples below.

Note: multiple values can be passed to one parameter at the same time (exception: e-commerce parameters and session parameters). These values can then be analysed separately. Multiple values must be separated using a semicolon.

11.1 Page Parameters

11.1.1 "Text" Data Type

These can be used to record the error messages on a page using the page parameter. By using the page parameter you can establish a direct relationship to the page on which the error message was generated.

```
wt.customParameter = {
  1: "Postcode incorrect"
};
```

In the Webtrekk tool, the parameter has been configured with the title "error message". The analysis is called up under "Navigation > Pages > Page Parameters > Error Message".

For example:

Error Message	Number of Error Messages	Number of Error Messages in %
Postcode incorrect	700	70
Name missing	200	20
Accept Terms & Conditions	100	10

11.1.2 "Number" Data Type

These are used to assign a virtual value to each individual page impression. Name the parameter "Page value" in the Webtrekk configuration.

```
wt.customParameter = {
  2: "0.5"
};
```

This parameter of the type "number" is available as a metric in the analyses. An analysis can, for example, be carried out via the page analysis function under "Navigation > Pages > Pages".

For example:

Page	Page Impressions	Page Value in %
Newsletter order	400	200

11.2 Event Parameters

11.2.1 "Text" Data Type

A typical example is when passing the teaser position (top teaser, content teaser etc.) as an Event ID for teasers that frequently change on the page. The precise name of an individual teaser is updated as an event parameter.

```
wt.customClickParameter = {
  LINKNAME: {
    1: "20_percent_discount"
  }
};
```

In the Webtrekk tool, the parameter has been configured with the title "Teaser name". The analysis is then carried out via "Navigation > Event Parameters > Teaser Name".

For example:

Tease Name	Number of Teaser Names	Number of Teaser Names in %
20_procent_discount	352	36,59
sale_start	325	33,78
new_collection	285	29,63

11.2.2 "Number" Data Type

A typical area of use for the event parameter is when a virtual value is provided with a click.

```
wt.customClickParameter = {
  LINKNAME: {
    1: "2"
  }
};
```

This parameter of the data type "number" is available as a metric in the analyses. An analysis can, for example, be called up via "Navigation > Events".

For example:

Link	Quantity Events	Teaser Value
finale_sale	240	480
20_procent_discount	215	430

11.3 Session Parameters

11.3.1 "Text" Data Type

A typical use of a session parameter is to transmit user information. In this example, the login is followed by the user's club membership status.

```
wt.customSessionParameter = {  
  1: "Club Member"  
};
```

In the Webtrekk tool, the parameter has been configured with the title "Visit type". The analysis is called up under "Visitors > Visits > Session Parameters > Visit Type".

For example:

Visit Type	Number of Visit Types	Number of Visit Types in %
Club Member	120	60
Guest	80	40

11.3.2 "Number" Data Type

A typical use for the session parameter is to transmit a visit with a virtual value. The value can be updated during a visit. The Webtrekk tool shows only the last transmitted value.

```
wt.customSessionParameter = {  
  2: "1"  
};
```

This parameter of the data type "number" is available as a metric in the analyses. An analysis of the most recent sessions can be called up under "Visitors > Visits > Visit IDs".

For example:

Session-ID	Time	Visit Value
2123197367900589185	2010-01-01 15:30:30	1
2123188667600985169	2010-01-02 20:45:12	4

11.4 Campaign Parameters

11.4.1 "Text" Data Type

This can be used to evaluate a link position without setting up every link as a separate ad (i.e. with its own media code).

```
wt.customCampaignParameter = {
  1: "Logo"
};
```

Campaign parameters can also be passed in the URL. If the same parameter is defined in the pixel and the URL, the pixel takes precedence. A URL example: "www.webseite.com?mediacode=gmx&wt_cc1=Logo"

In the Webtrekk tool, the parameter has been configured with the title "Link position". An analysis can then be called up under "Marketing > Campaigns > Campaign Parameters > Link Position".

For example:

Link Position	Number of Link Positions	Number of Link Positions in %
Logo	230	67,25
Mainteaser	100	29,24
Footer	12	3,51

11.4.2 "Number" Data Type

This can be used if you pay a different amount per click and wish to evaluate this for every campaign.

```
wt.customCampaignParameter = {
  2: "0.70"
};
```

This parameter of the data type "number" is available as a metric in the analyses. An analysis of the advertising media can, for example, be called up under "Marketing > Campaigns > Advertising Media".

For example:

Campaign	Campaign Clicks	Click Costs
Banner portal.de	800	560

11.5 Independent Parameters

11.5.1 "Text" Data Type

A typical use of independent parameters is to measure the times a newsletter is opened. This parameter is not related to other objects in the Webtrekk tool. Measuring it does not create a session in Webtrekk so that opening a newsletter does not raise the number of visits, for example.

```

```

In the Webtrekk tool, the parameter has been configured with the title "Newsletter view". The analysis can be called up under "Visitors > Independent Parameters > Newsletter View".

For example:

Newsletters Viewed	Number of Newsletters Viewed	Number of Newsletters Viewed in %
newsletter_cw22_2010	14500	39,62
newsletter_cw23_2010	22100	60,38

11.5.2 "Number" Data Type

This can be used to assign a virtual value to a newsletter view.

```

```

Name the parameter "Newsletter value" via the Webtrekk configuration. This parameter of the data type "number" is available as a metric in the analyses. A daily analysis can, for example, be called up under "Visitors > Time > Days".

For example:

Days	Number of Newsletters Viewed	Newsletter Value
01.09.2010	16000	12800
02.09.2010	8600	6880
03.09.2010	7400	5920

12 Features and Bugfixes

Version	Description	Configuration parameter
4.4.7	support pre-rendering in safari 12.2	wt.ignorePrerendering
	support CTRL & LMB, if using "delayLinkTrack"	---
	Implement "Web to App" tracking	wt.sendViaSDK
4.4.6	write request queue instantly	---
	support comma in page name	---
	deactivate CDB for Safari (ITP 2.1)	---
4.4.5	integrate request queueing functionality	wt.requestQueueActivated, wt.requestQueueTTL, wt.requestQueueResendInterval, wt.requestQueueSize
	send only tracking parameter that differ from the default value	---
	remove not supported tracking parameter	---
4.4.4	remove Webtrekk Real Time Bidding	---
4.4.3	random string appended after tracking endpoint ("wt")	wt.requestObfuscation
	random order of all tracking parameters (including "p")	wt.requestObfuscation
	page names contain hash-tag values	---
	ignore eid in url, if timestamp is greater than 15 minutes	---
4.4.2	possibility to filtering out URL fragments from page URL	wt.pageURLPattern, wt.pageURLReplace
	extension of the Webtrekk App-SDKs (Android) to overwrite the ever ID	---
	updated default value for "execRTA" = false	---
4.4.1	security fix for overlay / heatmap	---
4.4.0	deactivate RTA and CDB, if the user has a 1st party optout or sampling cookie	---
	use existing ever ID from Webtrekk App-SDKs (iOS and Android)	---
	generate a new Ever ID, if it is invalid (wrong timestamp)	wt.validateEverId
	insert pixel version in cdb requests	---
4.3.9	setting Ever ID with URL Hashes	---

Version	Description	Configuration parameter
	separate function to override the Ever ID	wt.setEverId
4.3.8	separate function to override the referrer url	wt.setReferrer
	send 'cdbeid' every 15 minutes	---
	increase cdb image service timeout from 500ms to 2000ms	---
4.3.7	No "pu" parameter in form and heatmap requests	---
4.3.6	Manual update of form fields	wt.updateFormFieldStatus
	Custom form object	wt.CustomForm
	Send two same campaigns with different campaign action(click, view)	---

13 General terms of use

With the pseudonymised data transferred via the Webtrekk tracking system (Analytics, Cross Device Bridge), Webtrekk may process information classified as personal data (such as the hashed e-mail address, 3rd-party cookie and so on). With the implementation of the Webtrekk tracking described in this document, the data are processed on behalf of the portal operator (referred to as the customer in the following).

The customer has chosen Webtrekk as a service provider within the scope of the duty of care pursuant to Section 11 of the Federal Data Protection Act (BDSG). A prerequisite for the legitimacy of contract data processing is that the customer has already concluded a licensing and testing agreement with Webtrekk, and deliberately issues an order to Webtrekk for mapping the tracking data (Analytics, Cross Device Bridge) to user profiles with the implementation of the tracking. These terms of use complement the existing order for contract data processing pursuant to Section 11 BDSG.

Insofar as the term "data processing" or "processing" (of data) is used in these terms of use, this generally means the use of personal data collected under a pseudonym. Using personal data encompasses in particular the collection, storage, transmission, blocking, deletion, anonymisation, pseudonymisation, encryption or other use of data.

13.1 Object of these terms of use

1. The vendor provides the following services for the customer:
 - Exchange of pseudonymised data with the customer and storage of the data
 - By implementing the tracking (Analytics, Cross Device Bridge) on the customer's portals, the vendor sets up pseudonymised data mapping for the customer
 - Upgrading of data (data mining)
 - Reporting the results and insights
2. Circle of entities affected by data processing:
 - Clients of the customer
 - Parties interested in the customer's products/services
3. Definition of pseudonymised data: The customer provides the vendor with pseudonymised, user-specific profiles. Such a profile assigns certain profile characteristics to a user via a pseudonym cookie ID.

13.2 Rights and obligations of the customer according to these terms of use

1. The customer is the responsible body (Section 3, Paragraph 7 BDSG) for the processing of data by the vendor under contract. Evaluating the legitimacy of data processing is the sole responsibility of the customer.
2. The customer shall promptly inform Webtrekk if errors or irregularity are noted in the context of the tracking.

13.3 Rights and obligations of Webtrekk according to these terms of use

1. Webtrekk processes personal data exclusively within the scope of the existing license agreement and these terms of use.
2. Webtrekk confirms that it has appointed an operational Data Privacy Officer pursuant to Section 4f BDSG.
3. Webtrekk is obligated to organise the company and operating processes so that the data it processes under contract for the customer are protected as respectively required and secure against third-party access.
4. Webtrekk shall promptly inform the customer if it is of the opinion that the use of the tracking by the customer violates legal regulations. Webtrekk has the right to suspend the respective services until this is confirmed or changed by the customer.
5. Webtrekk has the right to adapt the tracking and data processing to the applicable legal provisions at any time, even without a prior directive issued by the customer.
6. Webtrekk has the right to limit the tracking and data processing at any time unless anything to the contrary is specified in the license agreement or another agreement.

13.4 Data confidentiality

1. In the course of processing data for the customer, Webtrekk is obligated to maintain data confidentiality pursuant to Section 5 BDSG.
2. Webtrekk warrants that the company and its employees are familiar with the respective applicable data privacy regulations and their application.

13.5 Protecting the rights of affected parties

1. The customer bears sole responsibility for protecting the rights of affected parties.
2. Insofar as the participation of Webtrekk - in particular providing information, correction, blocking or deletion - is required for the customer to protect the rights of affected parties, Webtrekk shall take the respective steps required according to the directives of the customer.

13.6 Compensation

The remuneration of the Webtrekk services are regulated by separate license agreements. Additionally, the Cross Device Bridge is available for free until further notice with an existing Webtrekk license agreement. Webtrekk reserves the right to implement a billing model for its use at a later date and only after first consulting the customer.

13.7 Technical and organisational data security measures

Webtrekk commits to the customer that it shall maintain the following technical and organisational measures required to comply with the applicable data privacy regulations: physical access control, data access control, transfer control, input control, order control, availability control. See the license agreement for the respective underlying details.

13.8 Term and cancellation

1. The term of the Webtrekk services is regulated in separate license agreements. The usage of additional services (e.g. Cross Device Bridge) is limited to the term of the respective license agreement.
2. The customer can revoke consent for the further recording of its data at any time with deactivation of the respective tracking (Analytics, Cross Device Bridge).
3. After use of the tracking ends, Webtrekk has the right to block and delete data collected with the tracking unless anything to the contrary was agreed with the customer.
4. Webtrekk has the right to terminate the tracking services (Cross Device Bridge) at any time unless anything to the contrary is specified in the license agreement or another agreement.

13.9 Final provisions

1. The provisions of the respective license agreement take precedence.
2. Subsidiary agreements require the written form.
3. Asserting the right of retention pursuant to Section 273 BGB is excluded in regards to the processed data and the corresponding data carriers.
4. Should parts of these terms of use be ineffective, the effectiveness of the remaining provisions shall remain unaffected.

14 Contact

Please do not hesitate to contact us if you have any questions regarding configuration. Webtrekk offers various support and consulting packages for priority support and comprehensive advice. Please feel free to contact us to obtain your own personalized offer.

Webtrekk GmbH
Robert-Koch-Platz 4
10115 Berlin

fon 030 - 755 415 - 0
fax 030 - 755 415 - 100
support@webtrekk.com

www.webtrekk.com