FACT-Finder

Integrating the Tracking Interface

FACT-Finder®
Europe's leading conversion engine

FACT-Finder Development

E-mail: support@fact-finder.com
Phone: +49 (0)7231/12597-701

Version: 6.7.0

Latest Update: 19/01/2012

Table of Contents

FACT-Finder Tracking

Tracking information lets the FACT-Finder Search tool learn automatically from user behaviour.

The data provided through this interface can be used for a variety of purposes. These include the tracking of events such as users clicking on a details page and placing a product into the shopping basket and purchasing it. The information can form the basis for automated search optimisation. It automatically improves the search result on the basis of established consumer behaviour.

The information on product purchases can also be used as a basis for the Recommendation Engine module. More information about that module can be found in the corresponding documentation.

The events you pass on to FACT-Finder depend on your needs and objectives. However, automated search optimisation requires the details page event to be sent at the very least. The following events are supported:

- Clicking on the product details page
- Placing a product into the shopping basket
- Purchasing a product
- Clicking on a product suggested by the Recommendation Engine
- Feedback for the search results

Sending data to FACT-Finder

By default, the information is sent in a URL containing specific parameters. Please see the explanation at the end of this documentation concerning the parameters required for each event.

In order not to distract or irritate the user, we employ a JavaScript function that uses AJAX to send the data to FACT-Finder in the background. However, any other option for sending a URL in the background can be used to send the data. We also offer a web service interface to send the click information.

Integration

HTTP interface (JavaScript function)

Below is an example of how to implement the function that sends the clicks on the details pages to FACT-Finder. This function uses the jQuery JavaScript library (see `http://www.jquery.com/`).

```
function clickProduct(query, id, pos, origPos, page, simi, sid, title,
pageSize, origPageSize, channel, event){
if(event == null){ event = 'click';}

$.ajax({
  type: 'POST',
  url: '../SCIC.ff',
  data: {
          query: query,
          id: id,
          pos: pos,
          origPos: origPos,
          page: page,
          simi: simi,
          sid: sid,
          title: title,
          event: event,
          pageSize: pageSize,
          origPageSize: origPageSize,
          channel: channel
  },
  contentType: 'application/x-www-form-urlencoded; charset=UTF-8',
  cache: false,
  async: false
});
}
```

In the script above you will need to change the URL to point to your FACT-Finder Search instance. You will also need to use a proxy script that lies under the shop domain that redirects the call to FACT-Finder if the Search environment and the web shop are located on separate servers. You can find instructions on how to do this in the Suggest documentation.

The script above is designed specifically for tracking clicks on the product-details pages. You will need to customise the parameters that are passed accordingly for other request types. The parameters are defined in the script as values of `data`. The structure is `PARAMETER_NAME: PARAMETER_VALUE`.

Web service function

You can retrieve the interface documentation in the form of WSDL code by calling a URL as follows:

```
../webservice/ws67/SCIC?wsdl
```

The `logInformation` method passes the data on to FACT-Finder. You will need to specify a map that contains the tracking information in addition to the channel name and the authentication object parameters.

Within the map, the parameter name must be specified as the key, and the key's value specified as a value. See the list below for the names of the parameters.

Parameters

The appropriate function must be called each time a corresponding event occurs (e.g. clicking on the details page, purchasing a product).

Optional parameters are marked by an *.

General parameters

The following parameters must or can be passed on with each call.

id

ID of the item for which information is to be sent. If you are using several ID fields (product ID, version ID, EAN, ISBN, etc.) please talk to Omikron about the value to be passed on at this point.

masterId*

If your shop contains item versions and you need to transmit the item version number with the `id` parameter, you will need to transmit the master item number via this parameter to ensure that products and possible events can be assigned to each other.

channel

The name of the FACT-Finder channel in which the search was conducted.

sid

Use this to pass the user's session identifier. This is only needed in order to determine the number of people interested in a product and can be anonymised.

event

This parameter specifies the event type. Please use the following values for the various types:

- `click` – Click on the product or link to the details page
- `cart` – Product placed in the shopping basket
- `checkout` – Product purchased
- `recommendationClick` – Click on a product suggested by the Recommendation Engine
- `feedback` – Comment on the search result left by the shop visitor

title*

The item name. This value is optional; you can also transmit an empty string or omit the parameter.

userId*

Use this to pass a user identifier. As with the session ID, this can be anonymised. The user identifier differs from the `sid` in that it remains the same for users who visit more than once. Use of this parameter is mandatory if the Recommendation Engine module is being used.

cookieId*

You use this parameter to pass a token that identifies the user over a longer period of time, even when not logged in to the shop. Typically this is stored in a browser cookie.

Click on the details page

You will need to send the following parameters in addition to the general parameters if you wish to pass product-click information:

query

The search term for which the user has searched.

pos

The position of the product in the search results. The position is the absolute position. This means the first product on page 2 is number 11 if a search result page contains 10 products. The first product on page 1 is number 1.

origPos

This transmits the original position of the item in the search result. FACT-Finder returns this value in a field (default: "`__ORIG_POSITION__`") in the search result.

Exception: If the number of products returned is 1, the `__ORIG_POSITION__` field is omitted. This is also the case if the products have come from the Campaign Manager module (Pushed Products that have been inserted into the search result). In such circumstances clicks on these products should not be passed to the API. The omitted field should be used to stop this happening.

page

The number of the search result page on which the selected product was displayed. The first page has the number 1.

pageSize*

The number of products per search result page at the time the click was executed.

origPageSize

The default number of products per search result page, i.e. before the user made any changes.

simi*

The FACT-Finder similarity value for the respective product. The similarity value is returned with the search result.

Shopping basket and purchase information

The additional data for shopping basket and purchase information is identical. If multiple different products are to be passed, multiple requests are required.

count

Quantity of product purchased. The figure is an integer.

price*

Product price; in each instance, the unit price should be provided, even if a customer has purchased several units of the same product and/or placed them into the shopping cart. If you offer support for multiple currencies, the tracking information passed for a given channel must always be in the same currency. Otherwise, statistical reports will be incorrect.

Recommendation Engine click

If you are using the FACT-Finder Recommendation Engine you can also transmit this click data to allow you to ascertain which recommendation is of greater interest for customers.

mainId

ID of the article for which the clicked article was recommended.

Search result feedback

If you want to allow users to comment on the search results, this can be provided here. Such a function allows you to specifically optimise search results and obtain customers' opinions. Unlike the other tracking options, this does not relate to a specific product so you will need to transmit a pseudo value in the ID parameter.

query

The search term for which a search was conducted and on which feedback has been provided.

positive

Send the value `true` at this point if the customer provided positive feedback. Send `false` if negative feedback was provided on the result.

message*

If the customer left a message as justification of their opinion, you can send it using this parameter.

User data

Information about shop visitors is key in improving the existing Recommendation Engine and Automatic Search Optimisation modules, and helping to develop the future Behavioural Targeting and Analysis Center modules. The Tracking interface is able to receive any sort of information. Naturally, all information should first be anonymised. The following information should form the basis for the data that is provided:

gender

The gender of the web shop visitor. Possible values are `m` (male) and `f` (female).

age

The age of the shop visitor.

Disguised data

Some of the data that is passed, while not contravening data protection legislation, could give users cause for concern if it is seen. For this reason, FACT-Finder allows you to disguise information before it is passed. This is done by encoding the URL portion that is to be sent using the Base64 algorithm and sending this as the value of a specific parameter. The names of parameters that contain encoded data can be configured[1]. The default configuration treats the parameters "`p`", "`q`" and "`j`" as encoded parameters.

Example: You wish to send the following data to the FACT-Finder Tracking module in coded format:

```
page=5&query=cigars&age=27&gender=f
```

The Base64 encoded value for this text would be:

```
cGFnZT01JnF1ZXJ5PWNpZ2FycyZhZ2U9MjcmZ2VuZGVyPWY=
```

When combined with the parameter name and the other, unencoded parameters, the request would appear as follows (the equals sign in the Base64 encoded URL portion has been URL-encoded before creating the http request):

```
./SCIC.ff?q=cGFnZT01JnF1ZXJ5PWNpZ2FycyZhZ2U9MjcmZ2VuZGVyPWY%3D&event=click
```

The outgoing text should be URL encoded before conversion to Base64 encoding. This process should use the same encoding that has been configured for FACT-Finder (normally UTF-8). The example above contains no special characters, so it is not clear that the URL has already been encoded. If the search query "skirts & dresses" is to be encoded, for example, the output text presented for Base64 encoding should be:

```
page=5&query=skirts%20%26%20dresses&age=27
```

Notes

The interface will return the value `true` if the parameters were transmitted correctly. This allows you to check that you are sending all of the required parameters.

[1] `tracking.namesOfEncodedParameters` in `fff.properties`

Depending on how the FACT-Finder Search is integrated (HTML, XML or web service), some of the values will need to be calculated. Most, however, can be derived from the search result and its search parameters (*SearchParams* object).

You can also call the URL in the source code when accessing the details page. However, you will need to ensure that the required parameters are also sent in order for this to work.

If you give users a direct option to add products to the shopping basket from the search results page, you need to ensure that you pass both *click* data and *cart* data when this event occurs. This is the only way to determine that the item has been placed into the shopping basket directly from a search term.

Any questions?

If you have any questions on how to proceed, please call: +49 (0)7231/12597-701 or e-mail support@fact-finder.de. A qualified assistant will be ready to help you.