eMail-Marketing-System

Release 7

Dokumentation

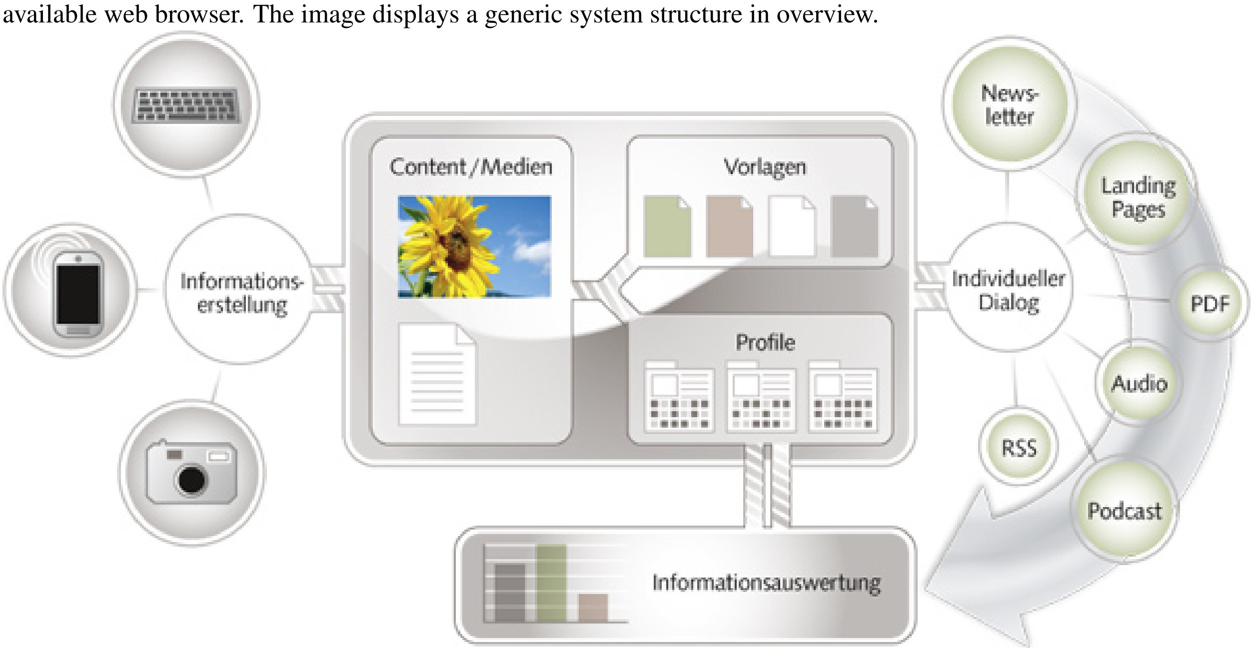
November 30, 2020

Contents

CHAPTER 1

System Overview

This overview shows in a succinct way the principal structures and functions of your email marketing system. As an ASP platform it is based on the most modern web technology and provides every function for the planning, design, implementation and control of results for marketing campaigns per eMailing compactly under one roof. With target-group-specific differentiation of contents and diverse display options: HTML/TEXT for screen display, PDF for print output, text-to-speech for content readout and with RSS-Feed for the inclusion of current data. Password-protected access to the email marketing system is possible at every workstation having Internet access and a commercially available web browser. The image displays a generic system structure in overview.



1.1 System time

The system time used internally is based on **Greenwich Mean Time** (GMT) as well as **Universal Time, Coordinated** (UTC) - of the international system time. All times have to be converted accordingly, i.e. for German-speaking regions (DACH): wintertime +1 hour, summertime +2 hours.

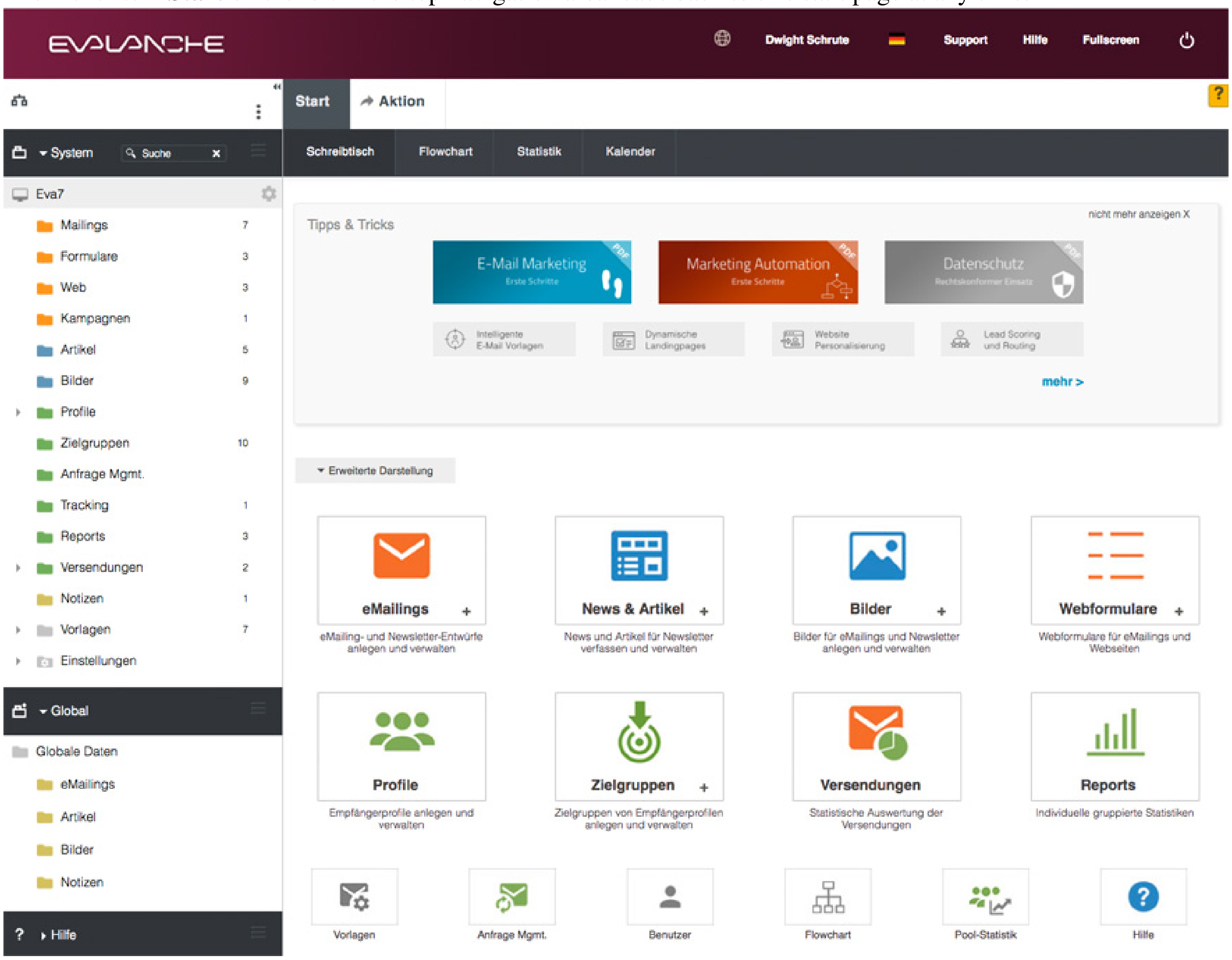
Example for Germany, Austria and Switzerland

Start time for an eMailing is to be 10:00 a.m. during the summer. In this case, transmission time is to be configured at 08:00 system time. For the same start time in the winter, transmission time is to be configured at 09:00 system time.

1.1.1 Start page on the client level

Following successful login, the system‘s start page is displayed on the client level. It serves primarily to provide access to every object and function for the planning, execution and quality control of eMailings.

The menu item **Start** on the left in the top navigation area leads back to this start page at any time.



You can get support on questions concerning system utilization under the menu items **Help** and **?**. Inquiry-Managers by mail are possible via the item **Support** on the top right. Context help on individual functions can be retrieved directly at relevant places via small *?*.

The left-hand section of the window shows the folder structure with every function for content-related and structural management of data and objects in the system. The order of these functions in the list can be easily and individually organized per drag-and-drop in accord with your conceptions and preferences.

1.1.2 Emailing made easy

The system provides a clear folder structure for saving and editing all objects - address profiles, images, texts, articles, templates etc It supports the associated work processes with a high degree of automation. This leads to intuitive work with simple work processes and high-quality results in a relatively short time, as the following example shows. Programming skills are not normally required when using the newsletter templates provided. An integrated Content Management CMS allows intuitive work by means of editing functions.

You can find every function and component for your eMailing in the system in various folders:

- Recipient profiles of client addresses
- Selection characteristics for target groups
- Selection of a suitable template
- Populating contents with text, images and buttons
- Variables for personalization and customization

The system provides an integrated form editor for the preparation of addresses. Simple and complex web form scenarios can thus be created in minutes and integrated in websites and articles, as well as on landing pages.

1.2 The first eMailing for beginners

1. Step: On the start page under **eMailing** select the prepared**example eMailing (Crossmedia)**. This is based on a template whose contents you can flexibly design as required entirely by means of CMS with texts and images. Without any programming skills at all.
2. Step: In the tab on the top right, select the **Transmission** tab
3. Step: In the target group field select *All Addresses* and leave the transmission time on *now*. Then click on **Transmission preview** to check the layout, contents and customer addresses.
4. Step: select **TRANSMISSION** tab - Check transmission-specific settings such as e.g. target group, sender, time, etc.
5. Step: Starting transmission - Now click on the button **Start transmission**. A message will now inform you that the eMailing will be sent to all valid profiles of the target group. Confirm this message.
6. Step: Check statistical data - in the **Transmissions** folder you will receive the first statistical evaluations of your eMailing approx. 5 minutes after the start of transmission.

1.3 Creating a new eMailing

- To create a new eMailing, click on the + sign on the **eMailing** button on the start page.

- A page opens subsequently, in which you first enter a name for the eMailing and text for the subject line.
- Then select an appropriate template in the field below, here ” Examplenewsletter 3 (Crossmedia) ”.
- Click on *save* to save this basic data.
- Change to the editing mode by clicking on the *Contents* tab.
- Here, you can now replace the header graphic by clicking the mouse on the graphic and inserting another graphic from the alternatives displayed below.
- In the middle part, delete the sample article by clicking on -.
- Click on + to insert your own article from a drop-down list.
- By clicking on an article in the eMailing you can change the contents of its text and images in a new window if necessary - finally click on *Save and apply*.
- The window will close automatically and you return again to the editing mode.
- Click on *Save editor* to apply all changes.
- You will now see your finished newsletter in a preview.

1.4 Creating a new article

- To create a new article, click on the + sign on the **News & Article** button on the start page.
- Select *Standard article* as article type
- This takes you to the editing mode and you can then define the individual fields with your information.
- **Text** In the template, enter your individual short text and long text in the designated fields. Define and mark your individual links to landing pages and / or forms.
- **Image**: From images already imported select the appropriate one for the individual articles and insert this. You can also import images from other sources as well. The system adapts the scaling and resolution of the images to the available space automatically.

You can then assign the article in the eMailing under the **Contents** tab.

1.5 Quality of the display

For this purpose there is the **Preview** function so that the quality of the display and contents can be checked for any errors prior to transmission.

1.6 Transmission quality check

The system provides a number of functions for quality assurance prior to transmission by clicking on the **CHECK** tab:

- **Display** provides a preview for checking contents and layout in different email clients.
- **Spam Check** uses a very rough but helpful check that - provided that no obvious deficiencies were detectable - returns the result "No Spam = GREEN". This allows you to control the quality of the delivery rate.
- **Display** (Quick) provides a preview for checking contents and layout by hiding code components.
- **Spam (Quick)** uses independent software containing the most common rules and characteristics for detecting spam and performs a calculation of the likelihood of spam with this.
- **Eye Tracking** analyses the mailing with regard to the places that a recipient will most likely view
- **Text analysis** provides an analysis of the advertising features of the text based on various criteria such as font, length of text, relationship between text and image
- **Links** displays the list of all standard and individual links integrated in the mailing in order to check for any possible errors. Such errors are are highlighted in red - as a visual prompt for correction.
- **Individualization** displays the representation of individualized and customized variables for the various addresses from the pool
- **Test transmission** checks e.g. by means of a target group "Test recipient", whether the display quality, customization/personalization and deliverability in the mail client are ok.

If the eMailing must be released by an authorized person, the tests must pass successfully.

1.7 Transmission control

The transmission of the eMailing can be started under **Transmission**. **now** is preset as start time - i.e. the eMailing is sent immediately after the start of transmission. With the picker you select a target group containing the profiles from the pool where the eMailing is sent. By clicking on **Transmission preview** the entire mailing is displayed once again in an overview . Here, you can make sure one last time that everything is ok before the actual transmission is started. Click on **Start transmission** and confirm the confirmation prompt.

1.8 Transmission statistics

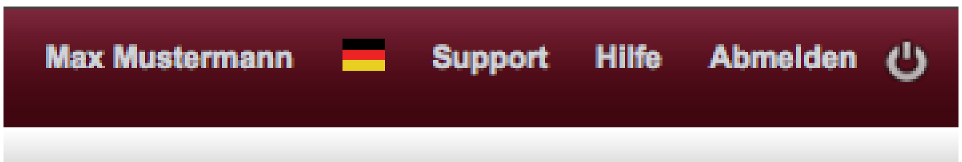
Immediately after the transmission, you can observe how recipients respond to the campaign with the aid of the feedback analysis. With the impression tracking you can see how many recipients open the eMailing. Link tracking allows you to find out which of the information of landing pages and websites provided was called by the recipients. The system also statistically records the number of deliveries and presents this graphically in the report. You can continue using all information gathered by the campaign later for statistical evaluations in order to address recipients even more personally.

1.8.1 Folder and Register Structure

1.9 Logo bar

Here you can change your access data, set support requests or call up the Help Center

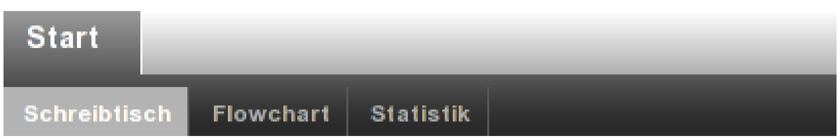
- **Your username** - Max Sampleman in this case - is used for the administration of your profile.
- **stylized flag** - changes the system language - there is a choice between German, English, French and Italian
- **Support** - leads to a request form for forwarding queries and error messages to the support team.
- **Help** - provides information on all topics relating to system utilization.
- **Logoff** - Logs you off from the system and logs you out.



1.10 Registers (tabs)

The start page and other pages in the system provide access to functions and data via registers (tabs). These are arrayed above the working area. With this, a clear and well-arranged display of the navigation is assured in every segment. The connection to the navigation in the overall system is maintained at all times.

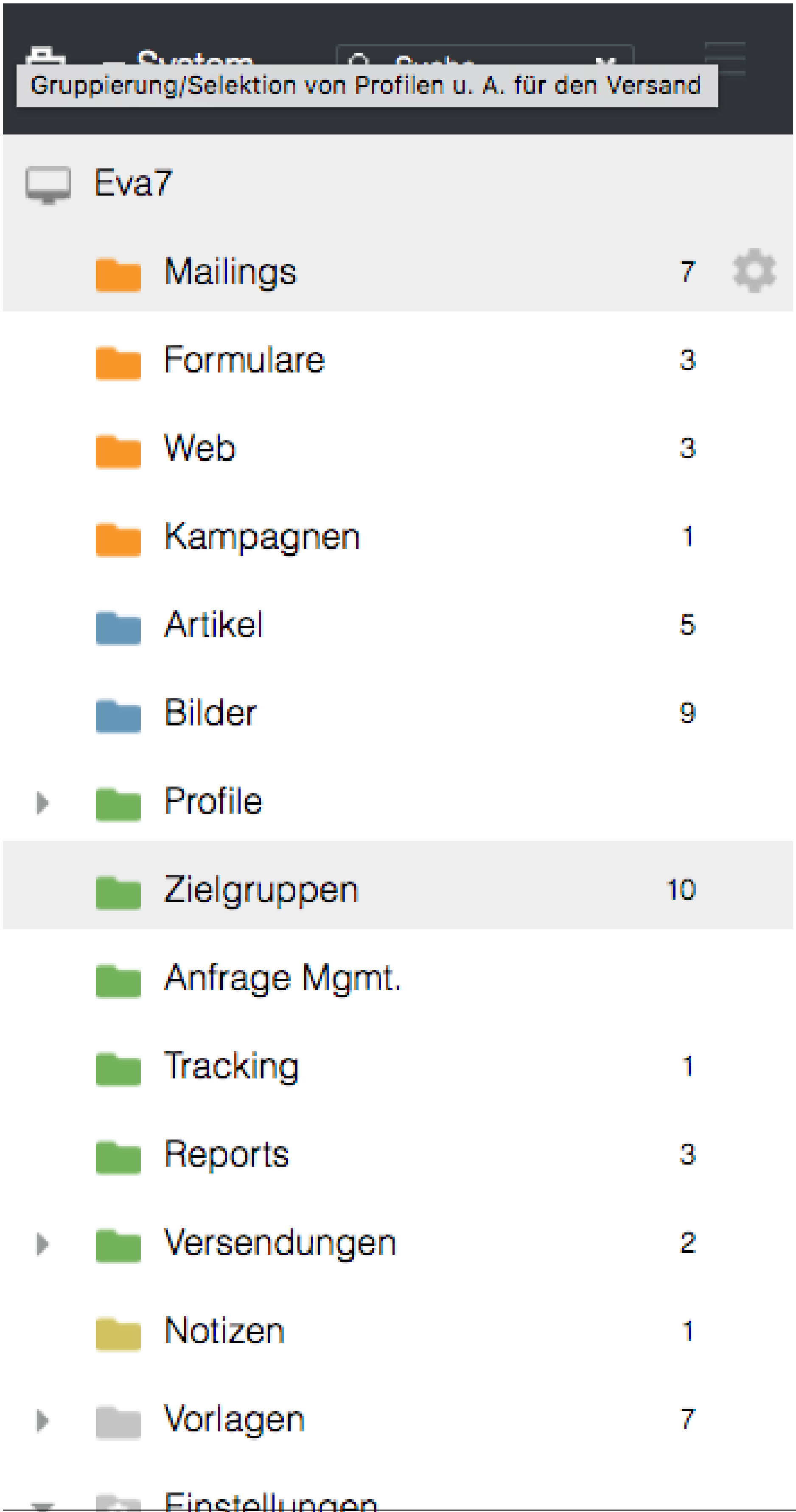
The following registers are available to you on the start page:



- **Desktop** - after logging in, displays the essential part of the functions for daily work once again in enlarged form from the complete list on the left edge of the screen.
- **Flowchart** - displays in graphic form the connection of the most important system objects and the information flow defined when new requests for further editing arrive via address forms. Here, the objects from the current configuration of the client and their relationships are displayed which handle the communication or internal processes. The objects can be clicked directly. The display allows dragging and clicking with the mouse wheel. The format (SVG) used is currently not supported directly in Internet Explorer 9 versions. In this case, use the download version and view it with software such as the free Inscape or Adobe Illustrator. Alternatively, a plug-in can also be used. All other current browsers allow viewing directly in the browser.
- **Statistic** - displays a summary of the result evaluation of the databases in a clear tabular form as well as bar charts or pie charts.

1.11 Folder structure

The system has numerous objects for managing, structuring and customizing information. These objects are stored hierarchically in folders. The following folder structures are used for orientation and structuring of the media and objects needed for an eMailing. The structure can be extended individually and customized flexibly to personal requirements. **See creating and editing folders**


Gruppierung/Selektion von Profilen u. A. für den Versand
Eva7
Mailings 7
Formulare 3
Web 3
Kampagnen 1
Artikel 5
Bilder 9
Profile
Zielgruppen 10
Anfrage Mgmt.
Tracking 1
Reports 3
Versendungen 2
Notizen 1
Vorlagen 7
Einstellungen

1.12 Main folder System

This folder contains the following subfolders:

- **eMailings:** Management of eMailings - This is the central work area where you create, send and change the contents of newsletters and eMailings by means of appropriate templates.
- **Articles:** Management of articles and news contents - Here you create new articles and manage them for further use and integration in newsletters and eMailings. Standard articles always have the same structure with a short and long version. This ensures optimized representation via the Cross Media formats newsletter HTML, newsletter TEXT, newsletter as website, news site, PDF and RSS as well as podcast and reading aloud by speech synthesis (Text-to-Speech TTS). Once articles have been created, you can use them in different eMailings repeatedly.
- **Images:** Management of images - In this area you can import and manage new images for eMailings and articles and classify with additional information such as "license-free". The formats supported are .jpg, .png and .gif. The resolution should be suitable for high-quality Cross Media representation on the screen and in PDF print format. The system generates suitable scaling of size and resolution fully automatically for displaying online and in printed media.
- **Web forms:** Management of contact and order forms - Here, you create and design web forms appropriate for your eMailing for address generation in the course of campaigns. Degrees of freedom in the design as well as in the number and arrangement of input fields ensure CI-compatible integration into existing web environments.
- **Profiles:** Management of address profiles - Here, you maintain and manage your address databases of customers and sales prospects. Entries can be created and maintained via data import, entry via web forms or manual entry. You can adapt and extend the data structures of a pool in the database for another profile as required.
- **Target groups:** Management of target groups - In this area, you can define, configure and edit the contents of target groups. Target groups are the central communication control for sending eMailings with customized contents to different recipient groups. The entire address database of target groups can be defined for all pools. For this purpose, you can use any number of comparative operators of characteristics e.g. target group A: *salutation = Mr* and *hobby = golf* and *zipcode = 20000 to 69999*.
- **Transmissions:** Transmission history of eMailings sent - This area is "fed" automatically with the data of your eMailings by the system. The data here corresponds to a status determined in intervals. For the real-time evaluation of the eMailing click on the corresponding eMailing in the list.
- **Reports:** Management of statistics or reports - In this area, you can create and edit new reports. A report consists of a grouping of several partial statistics - e.g.: How was the transmission history and development of the address database during the last eMailings over a defined period.
- **Query Mmgt.:** Management of queries that arrive via web forms and lead to entries in pools. This allows you to define the subsequent information flow and work processes systematically for different persons in order to quickly process incoming orders, contact queries, callback service requests and brochure requests.
- **Web:** The management and editing of websites, landing pages, news sites, RSS/Atom-Feeds, RSS/Atom-Feed-Reader and target group links is centrally organized here.
- **Tracking:** Here, you manage and maintain your checkpoints integrated in the eMailing in order to observe the utilization of the tracking objects with the aid of the Post Click Tracking Tool.
- **Templates:** Management of eMailing and article templates - Here, you can use a number of templates for different purposes by default, e.g. for newsletter or Cross Media eMailings.
- **Notices** Here, you can create notices and assign to objects - e.g. the notice that a template must not be edited.
- **Settings:** This menu provides all functions for the account and user administration. Access to this area is only possible with the corresponding access rights. Here, user accounts are set up and corresponding roles assigned - with clearly defined access rights. Several roles can be assigned to each user. The appearance of the user

environment with the familiar, individual look and feel can be adapted here to the particular CI if required. This area is also used for the administration of client information.

1.13 Main folder Help

This folder contains jump-off points for important and often used parts of the documentation such as video tutorial, object overview and current information for updates.

1.14 Main folder Statistics

Here you can track a statistic for the design of your current address count in a concise way: e.g., entries as a whole, entries in the last 7 days and unsubscriptions.

1.15 Main folder Notices

If you create a new notice for an object, this will be saved here by default.

1.16 Main folder Favorites

Your favorites as well as a list of the objects last used are displayed here. By clicking on the name, the system will jump immediately to the selected object. The favorites here can be rearranged via drag & drop

1.17 Main folder Global

All objects saved at group level in the **global data** folder are displayed for you here. These can be templates, images as well as articles that the corresponding clients can integrate in their articles, eMailings etc

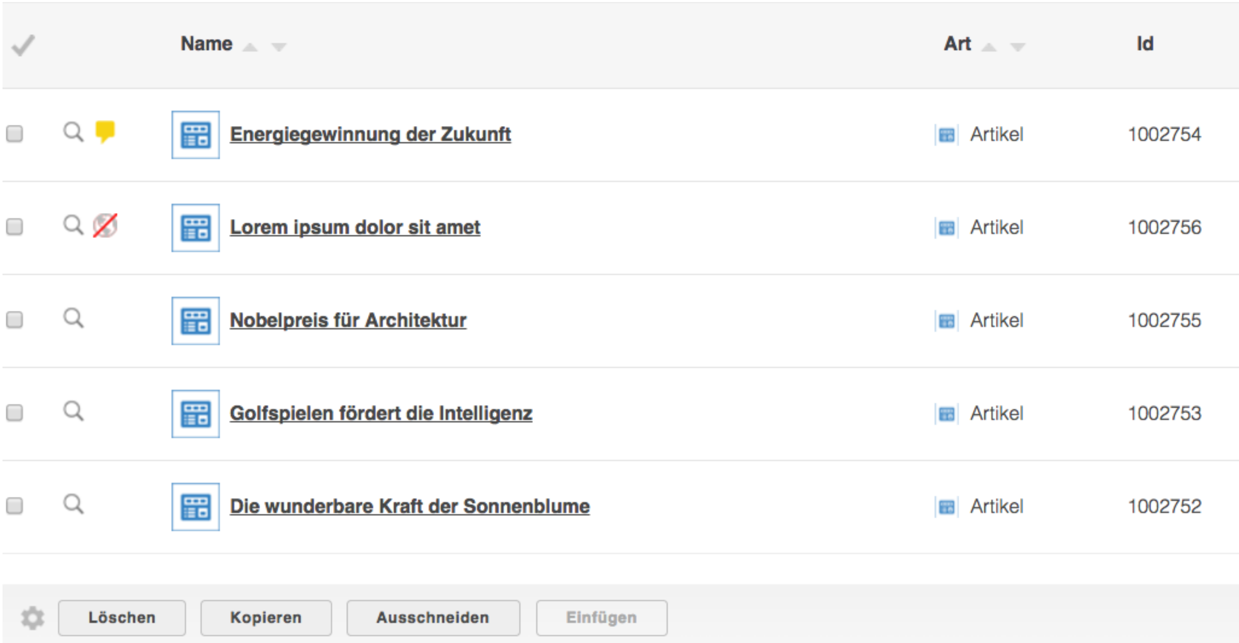
1.18 Main folder Reports

Here you can configure selected reports in a concise way and enable for a graphic display.

1.18.1 Working with folders and objects

1.19 List displays

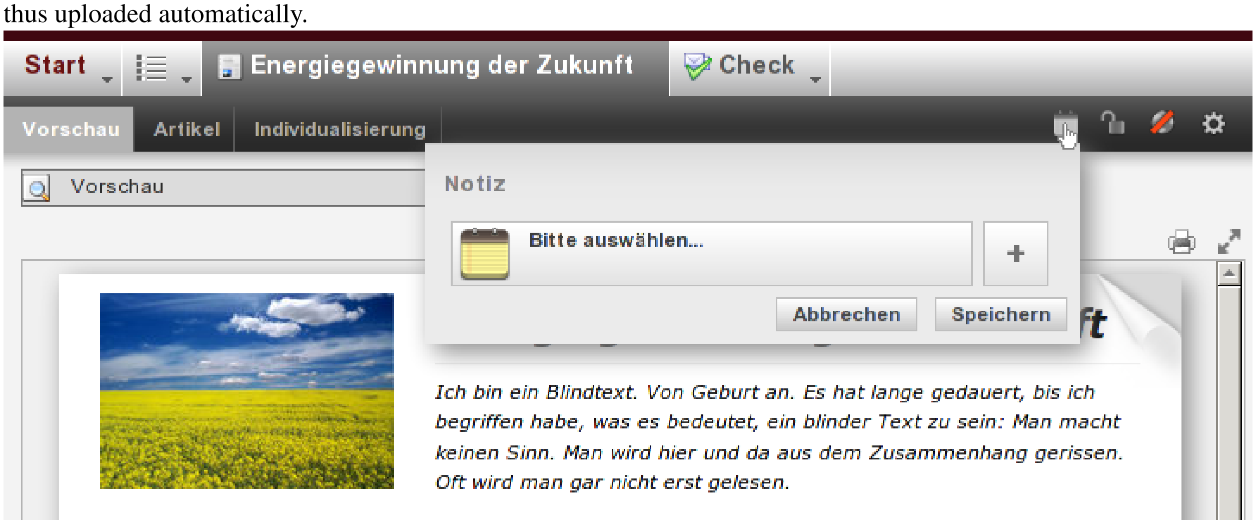
You can access all objects of the system using the folder structure described above. After clicking on one of the folders - e.g. Images - a window will open where the objects contained therein are displayed in list form. The system starts this list display with a general standard configuration. This configuration can be customized individually with regard to sort order and column display if required. The order of the lines can be customized with the Up/Down arrows next to the headings in ascending or descending order. In the upper left corner of the lists there is an icon for the representation of the columns. After clicking on this icon, a popup window will open in which all columns can be selected individually per checkbox. By clicking on the *Save permanently* button, the standard configuration for this list type will then be replaced by the customized configuration.



-  **Notice available:** This object has an attached notice.
-  **External access locked :** External access to this object is locked.
-  **Object lock:** An object lock is active for this object.

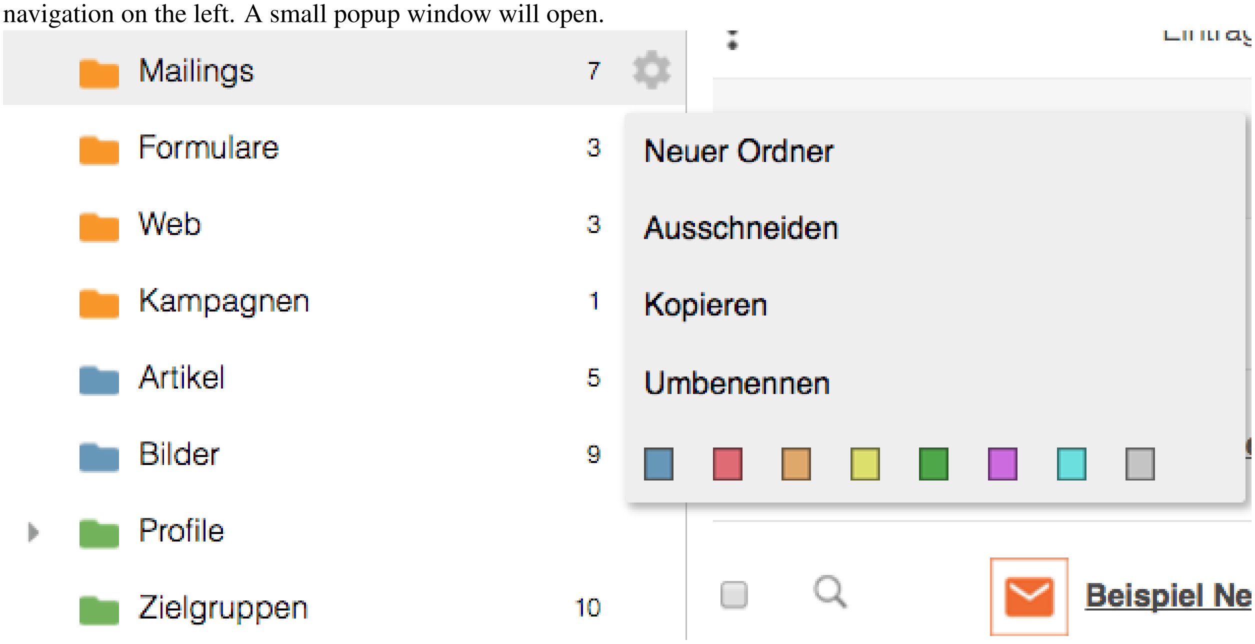
These functions are also available directly in the editing mask for nearly all object types.

Objects such as images, files or web pages can be moved from the desktop to the overview list by Drag & Drop and thus uploaded automatically.



1.20 Creating and editing folders

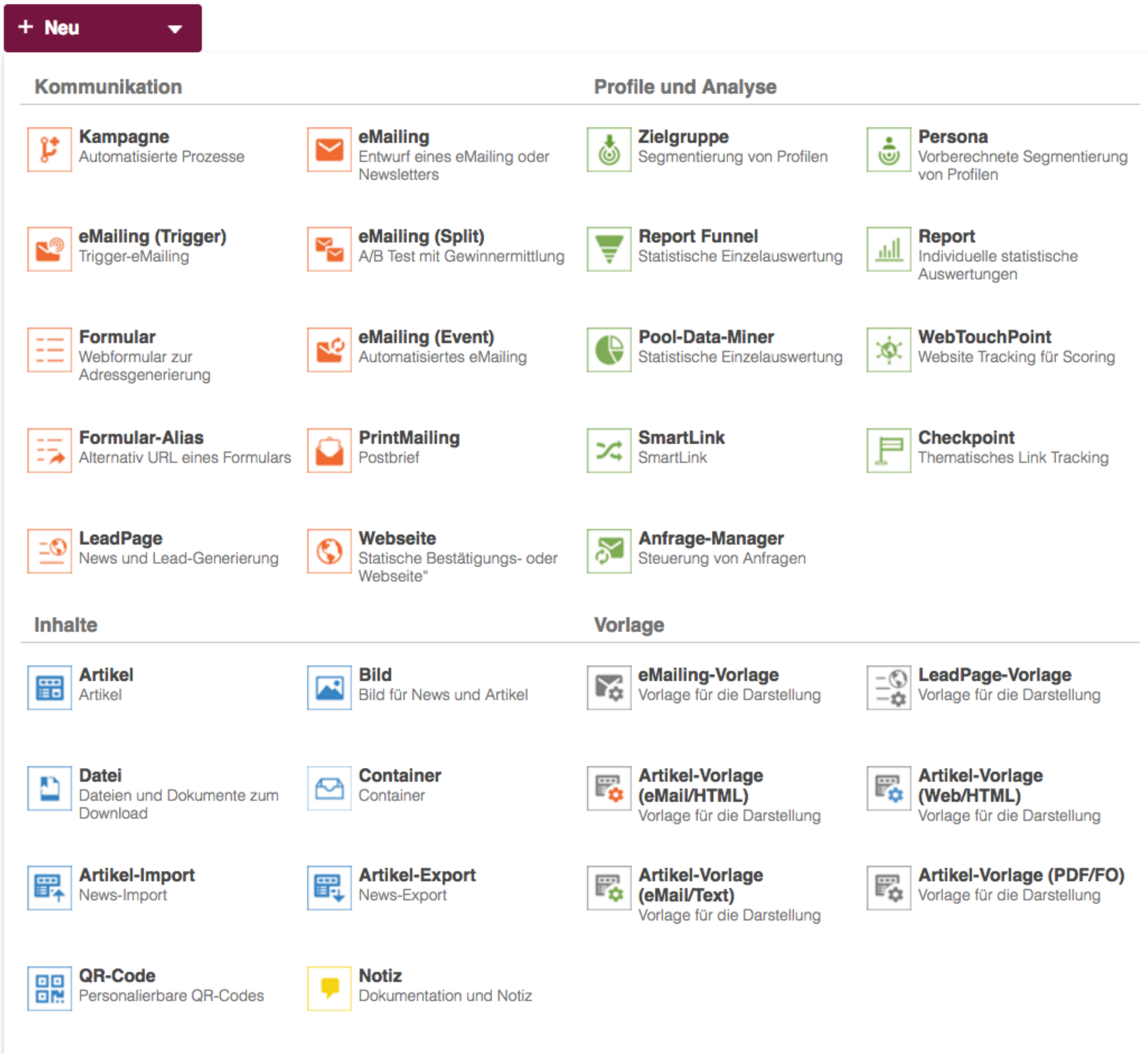
To create new folders and to edit existing ones, click on the stylized cogwheel behind the selected folder in the navigation on the left. A small popup window will open.



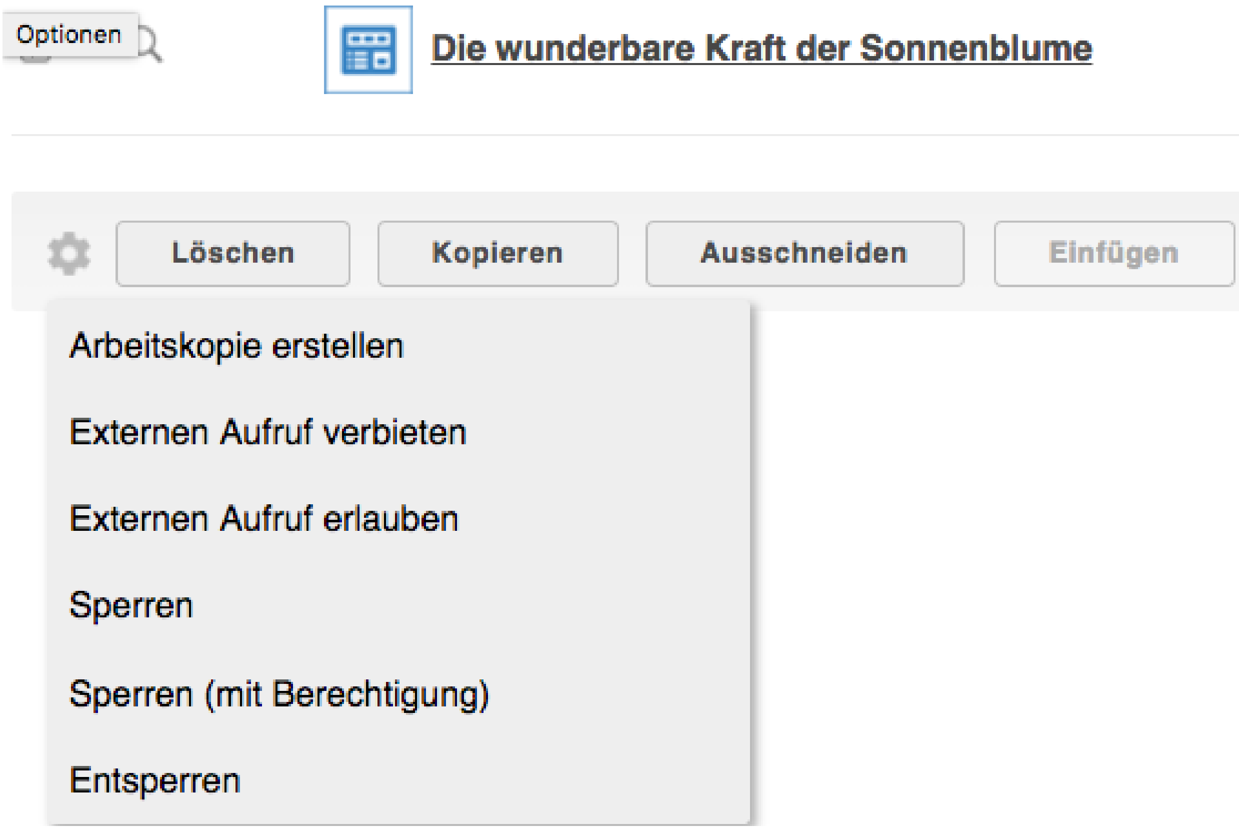
- **New:** If you want to set up a new folder, click on *New folder* in the popup menu. A folder is always created below the category currently selected.
- **Cut/Copy/Paste:** Folders, together with the objects contained within them, can be copied or moved. This also works for all clients.
- **Rename:** To change the name of the folder, click on *Rename* in the popup. All folders can be renamed with the following exceptions: The main client folder, the main group folder, the shipping folder, and the shipping subfolder.
- **Assign color:** The appearance of a folder in the tree can also be determined by assigning another color. This makes it easy for you to create visual groups and to find important items in the navigation created by you in the system.
- **Delete:** If you want to delete a folder, this can be done in the popup window by clicking on *Delete*. A folder must be empty in order to be deleted.

1.21 Creating and editing objects

- **New:** To create new objects, a page will open with fields for entering the object name and for the import or selection by clicking on a dropdown list.



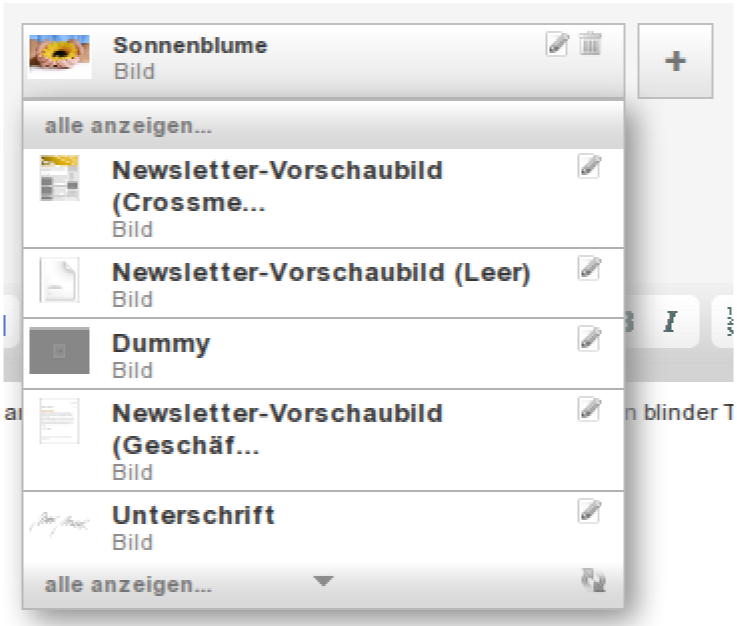
- **Delete, copy or cut:** Here, you can copy, move or delete objects within the system. First, select the object or objects by clicking on the checkbox. When deleting, the system will ask you once again before it performs the action. After performing the action *Copy* or *Cut*, change to the desired target location and click on *Paste*.



- **Create working copy:** With this function you can create a working copy of the objects selected (not possible for all object types).
- **Prohibit / permit external call:** This function allows you to lock external access for the selected objects (object lock).
- **Lock / (with authorization) / Unlock:** Provide / unlock all selected objects with an object lock.

> **Warning:** In many cases, the processing of objects (e.g. newsletters, statistics and templates) goes beyond several tabs. When a tab is changed, no data is stored. This happens only after explicitly executing the action *Save*. Clicking on *Save* completes the editing of the object. In a few exceptional cases, early data transfer is possible for reasons of clarity - in these cases, the system informs you of this.

1.22 Managing object relations with the object picker



The central input element of all object relationships within the system is the so-called object picker. Relationships between objects, such as the image of an article or the target group of an eMailing, are not only assigned quickly and easily in this way, but can also be created or edited directly on the spot.

The object picker behaves similarly to a select box. The objects that were last edited are displayed in a list with a small preview by clicking on the large button. Within the list you can either scroll through the list at the start and end of the list using the arrows or simply use the mouse wheel.

If you do not want use the object from the list for fast access, but would prefer to select an object from the tree and overview list in the usual view, then click on `Display all` or simply click a second time on the large button. Afterwards, you will see a miniature view of the usual image from the system with the option to select the object via tree and list.

The icon of the selected object as well as those in the list display the preview of the object as usual. Thus, for instance, the current configuration and statistical information of a target group. This will help you to find the right object more easily. Just hover the cursor briefly over the image.

In the currently selected object as well as in the list of the last objects there is an icon behind the name for editing the object.

If there is still no appropriate object in the system, i.e. you want to upload a new image for an article, for example, then click on the large + of the current selection. Some pickers allow you to define which type of object you would like to specifically create (e.g. you can select various article types for one article). The system now displays a window where you can enter the data for the new object. Save the object and then apply it. The new object is only selected directly in the picker.

If you want to cancel an existing assignment, e.g. remove an image from an article, then click on the icon to delete the current selection. Only the relation of both objects will be deleted here, but not the object itself that is affected. Also note that some assignments are mandatory and thus only another object can be selected.

1.23 Create and manage clients

Clients can create and manage only if they have particular access rights. For this purpose, click on the icon at the upper left next to the account name in order to switch to the global level.

You access the form for entering the new master data via the *New client* tab. After saving the data, the system configures a complete pilot environment for this new client.

You can edit the name via *Edit* and remove the entry again via *Delete*. When removing an entry with *Delete*, you will be prompted by the system whether you really want to delete the account for security reasons.

1.24 Global transmission overview

A transmission overview is provided on the start page of the global overview. The eMailings of all clients sent are summarized in a list. Clients that have already been deleted are ignored.

1.25 Automatic logout

If following login you are not active again for a long time, the system logs you out automatically. After this and before you can continue your interrupted work you have to log yourself in again. The data entered up to this point are not lost as a result. The system backs up the current state so that you can continue your work seamlessly. Only files that you uploaded and did not explicitly back up prior to the automatic logout are lost. Therefore, check after logging in again whether your data are correct and re-upload affected files if necessary.

1.26 Access authorization

If you do not have authorization to access particular sections of functions, the system indicates this to you in the login screen. Only a system administrator with sufficiently higher user rights can set up access authorizations for you to privileged sections and functions.

1.26.1 Security rules

In order to protect your email marketing system from unauthorized accesses, please follow - completely and without exception - all of the following security rules. If you have questions, please turn to our Service & Support staff or to your personal client adviser.

Important:

- Each time before leaving your workplace, always **LOGOFF** from the system without fail (click on the icon on the top right). Afterwards, close all browser windows.
- Never divulge your password to another person - our Service & Support staff will never ask for your password.
- If you have forgotten your password, our Service & Support staff will assign you a new one.
- Change your password routinely - or let our Service & Support staff change it for you.
- Immediately delete emails that contain passwords - also from the recycle bin of your email client and operating system.
- Delete user accesses immediately that are no longer valid or have these deleted by our service and support staff.

- Routinely check the rights structures of users‘ accesses for their being up to date or permit users to check.
- Inform our Service & Support staff immediately in the case of inexplicable results, e.g., upon altered or missing data, upon faulty functions or display problems.
- Avoid logging in on the system to outside computers. You should always use a demo account in the case of presentations and immediately change the associated access data at the conclusion of the presentation.
- Recommended browser: Mozilla Firefox in a current version

1.26.2 Defaults for date inputs

Instead of an absolute value you can also enter a value for date entries with the aid of a text representation. To do this, specify a time relative to the current day in English:

- *today*: Today
- *today midnight*: Today at midnight
- *today noon*: Today at noon
- *-7 days*: Seven days ago
- *+1 month*: Next month
- ...

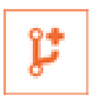
Caution: Please note that target groups with date ranges should be specified so that an entire day is covered especially when using event mailings. This means that you should never mix different input formats and in the preview always make sure that the time displayed is always covered by a complete day or by the corresponding time period.

CHAPTER 2

Objects and Media

The following overview displays every object and medium that you require for your eMailing and manage in the system. Depending on the selected price and service plan, you may not be able to use or have only restricted use of particular objects.

2.1 automation

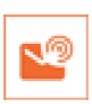
 *Campaign Designer* - Area for communication lines, which integrates the already existing system components and extends them by a time-controlled process flow.

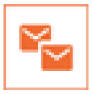
2.2 Communication

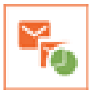
 *eMailing* - Area for managing and editing eMailings that you create based on a template. The eMailing contains all relevant information necessary for the transmission to a target group.

 *Transmitted eMailing* - Area with statistical evaluations of eMailings after their transmission. The system generates a copy of the original eMailing automatically during transmission.

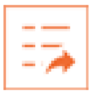
 *Event Emailing* - Is an eMailing transmitted recurrently to a target group within a specific time period - e.g. birthday eMailing.

 Trigger-eMailing - is an eMailing that cannot be sent manually. Trigger-Mails must be sent from other objects(forms, etc.).

 *Split campaign* - simultaneous transmission of different eMailings to a part of a target group. The most successful eMailing is the basis for the transmission to the other profiles of the target group.

 *Transmitted split campaign* - Statistical evaluation of all eMailings that were part of a split campaign . Here, in a transmission determining a winner, you can also find a comparison of the current statistics, which determine the winner.

 *Form* - Area for managing and editing web forms. These are used for generating addresses and creating profiles e.g. by registration, survey, order or competition forms as well as opinion polls.

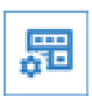
 *Form alias* - Are additional links to an existing form. If you use a form at several locations at the same time e.g. for competitions, you can then define additional source links in order to assign entries later according to origin.

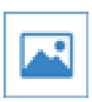
 *LeadPage* - Area for creating a web page for integration into external websites. This allows you to embed a list of articles directly on the website and display them individually.

 *Website* - Object for saving a static web page (e.g. landing page).

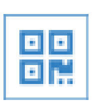
2.3 Content

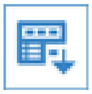
 *Article* - Area for managing and editing articles and news displayed, for example, in the newsletter or on the news site based on special templates.

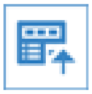
 *Article type* - Configuration of article types and their fields. They are the basis for articles.

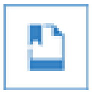
 *Image* - Area for the storage and management of image material for use in articles. Image size and scaling are freely selectable during upload. Permitted formats are .jpg, .png, .tif. The system carries out all subsequent transformations and compressions automatically.

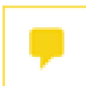
 *Dynamic image* - Object for the dynamic generation of personalized images (SVG).

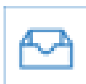
 *QR-code* - 2D barcode (QR code) for the transmission of text and/or links to suitable viewing devices (e.g. smartphones).

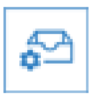
 *Article export* - Area for creating and managing individual RSS or Atom Feeds for integration in eMailings and web pages.

 *Article Import* - Area for importing RSS/Atom feeds with direct display as an article in the eMailing.

 *File* - Object for saving any kind of data (e.g. PDF, CSS etc.).

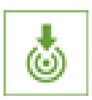
 *Note* - Object for creating documentations.

 *Container* - Content object for implementing translations, different eMailings - designs, etc.

 *Container type* - Configuration of container types and their fields. They are the basis of containers.

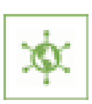
2.4 Profiles

 *Pool and Profiles* -The pool is the central database for the storage of profiles. You can adapt and extend the data structure to suit your individual needs depending on your requirements.

 *Target group* - By defining a target group you group entries of your pool on profile information (regions, male/female, preferences, etc.). Target groups are used for the selective transmission of eMailings, for customization of contents as well as for Inquiry-Management, for web personalization and for generating differentiated reports.

 *Persona* - Personas represent target group-like profile segmentations, but the criteria can also be configured to be out of focus and the results are automatically precalculated.

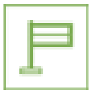
 *Inquiry-Management* - New incoming form requests (e.g. purchase orders, bookings, service enquiries) can be forwarded by means of the Inquiry-Management to the persons responsible by email, who then process these enquiries.

 *WebTouchPoint* - Track external websites and identify profiles that have visited them.

2.5 Analysis

 *Reporting* - Area where you can prepare, compile and evaluate reports on various types of media and actions for a specific time period.

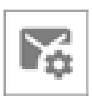
 *Pool Data Miner* - Area for grouping profiles from the pool with specific characteristics. You can use these for more complex statistical evaluations in which groups of pool options are grouped hierarchically.

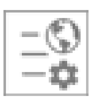
 *Checkpoint* - Tracking object - used for personal identification of users for post click tracking as well as link-tracking.

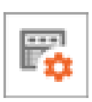
 *SmartLink* - Tracking-Object for enabling personal identification and target group-dependent linking (e.g. link tracking, web personalization, profile enhancement)

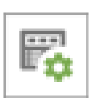
 Funnel - statistical object to display different TargetGroups in a graphic relative to each other. Used to graphically display e.g.: Lead qualification progress.

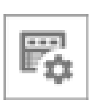
2.6 Templates

 *Emailing template* of an eMailing** - Emailing template for the following media: HTML, TEXT, WEB, LANDINGPAGE, PDF.

 `LeadPage-template` of a LeadPage** - Template of the LeadPage for the following media: HTML, LANDINGPAGE, PDF.

 *Article Template* E-mail (HTML) within an HTML mailing** - These article templates can be placed in e-mailing templates (HTML) via slots ''{SLOT_x}''.

 *Article Template* E-mail(text) within a TEXT mailing** - These article templates can be placed in e-mailing templates (TEXT) via slots ''{SLOT_x}''.

 *Article Template* PDF (FO) within a PDF document** - These article templates can be placed in e-mailing templates (PDF) via slots ''{SLOT_x}''.

Article Template Web (HTML) for Web Version in the browser** - - These article templates can be placed in e-mailing templates (WEBSITE) via slots ‘’{SLOT_x}’‘. This type of article template can also be used in news sites.

2.7 Settings

Users - Area for setting up and managing users of a client. You can assign different roles to each user. The system derives the necessary user rights from this.

Role - Rights configuration that you can assign to the individual users, e.g. *editor*, *web designer*, *management*, etc...

Theme - Configuration of the theme of the admin user interface. E.g. individual logo, definition of colors, etc...

2.7.1 Pool and Profiles

All pools are always created by the system in the main *Profiles* folder. There can be customer profiles in each pool. A single pool containing all customer profiles is normally sufficient for anyone.

Profiles are either entered by imports from existing address data, by manual input or via web forms. In addition to the standard fields - such as name, address and email address - which are already predefined, you can freely define and set up additional fields for your purpose. You define a data type for each of these fields to save the desired information appropriately there.

By clicking on the pool folder, a page opens listing the profiles contained within it. To edit individual entries, simply click on *Details*.

You get access to the pool configuration by clicking on the *Configuration* tab. After that a page opens with the current configured fields and field names of the pool. You can edit the pool fields by clicking on *EDIT* and save the changed data by clicking on *Save*.

Profile list

You can display the profile list grouped at 10, 25, 50, 100, 250 or 500 profiles and browse page by page. Alternatively, you can search the list specifically via a search field above the list. If the search is successful, the entries found are displayed in a partial list. If you delete the search entry and click again on *search*, you return to the complete list.

By selecting the *Delete* checkbox and then clicking on *delete* you can delete profiles unrecoverably from the pool. Furthermore, you can *edit* each profile of the list manually or click on *New entry* to enter a new profile via an input mask. Likewise, the history (entry, confirmation, logout, etc.) of each profile can be viewed in the editing mask.

In the profile history you can view the last actions of a profile (newsletters, forms, WebTouchPoints, SmartLinks, checkpoints, etc.).

Important: Profiles entered manually always contain *Opt-Out* as default permission. This means that these profiles will not be listed in a Inquiry-Manager.

Configuration

All configured fields of the pool are displayed here grouped together. This grouping can be adapted by the user. To do this, click on + at the end of the tabs of the groups. You will see all groups that have already been created and you can create new groups as well as rename or delete any groups available here. To change the sequence order of the groups, keep the stylized lines on the right pressed down completely and drag the line to the required position. After changing the groupings, the pool must be saved for these changes to take effect.

You can change the field configuration by clicking on *Edit*. You have the option to insert more fields into your pool. A *New field* button at the start and end of the field list per group is provided for this purpose. After clicking on the button, a dialog appears for entering the data. The name serves as a description and is not subject to any restrictions. This name is used as a field label in forms and so forth. The internal name is normally generated directly from the name displayed and may only consist of capital letters and numbers and is used later as a field name and placeholder.

You can define a separate type of your own for each field. This determines which options (contents) are entered here. Furthermore, the subsequent input options in the form are thus predefined to a certain extent. The following field types are provided:

- *Single-line input*: A line of free text can be entered.
- *Multi-line input*: One or more lines of free text can be entered.
- *Single selection*: One of these options must be selected from a list of options.
- *Multiple selection*: *0* to +n* options can be selected from a list of options.
- *Salutation‘, ‘Country‘, ‘State‘, ‘Mobile area code*: Predefined single selections.
- *Date‘, ‘Date with time*: Fields for entering a date (with time). Alternatively to the static entry of a date, a dynamic one in the form of text-to-date can be entered as well (e.g. “today - 5 days” for: “five days ago”).
- *E-mail*: Text input with validity check for a valid e-mail address.
- *True/False*: An input field with a yes/no option
- *Whole number*: Number that is simply incremented in the form and that can be used there for scoring methods. Attention: due to the limited numeric interval (0 to 4,294,967,295) this field is not suitable for long IDs or customer numbers.
- *Postal code*: An input field for numbers
- *IP address*: Read-only field for the IP address of a profile

In the case of fields of type *multiple selection*, *single selection* and *True/False*, the following functions can also be used: - *Content Profiling*: Assignment of profile features for articles to the automatic sorting in eMailing slots and for determining the hit ratio of personas. - *Profile Tagging*: options of this field can be set automatically by the opening behavior of the article recipient.

You can remove existing fields with - at the end of the line - but only if this field is not being used in any form. If this is the case, you will get a list of the object where the field is still used. You cannot generally remove standard fields of a pool.

Mandatory fields define that this field must be contained within each form or must always be specified by manual entry of a profile via the profile list. Fields that are already embedded in forms and are not marked as mandatory fields cannot be marked later in the pool as a mandatory field. For this purpose, a field must first be marked in all forms as *mandatory* before it can be declared as a mandatory field.

When using single or multiple selections, you can specify the type and the options for the entries. To do this, you must extend the options by entering one or more options (one for each line) and append them by clicking +. Existing options can be removed with - if no profiles currently exist in the pool. You can also specify the default order here by entering a number before the option.

You can also assign a *default* in order to also allocate a default value already at pool level for creating new profiles - identical to the form.

To move fields within the group, keep the field at the right end of the line pressed down and then move the line to the required position. The sort order will be saved immediately in the system.

Internal fields

The internal fields are a constituent part of each pool. These are always created and cannot be deleted. The system saves (automatically in some cases) information of the profile here, which is important for the internal processes.

- *Language*: Language of the profile (e.g. the form language during registration)
- *Profile ID*: Internal ID of the profile. This ID is distinct for all the profiles of a pool and is automatically assigned during creation.
- *External ID*: Free field for storing an ID from a third-party system. The third-party system is responsible for ensuring the distinctness etc.
- *Origin form*: Form in which the profile was initially entered.
- *Time of the subscription*: Time at which the profile was created.
- *Time of last change*: Time of the last change to the profile in a form
- *Permission*: Current permission of the profile
- *Unsubscribed*: Boolean field whether the profile has been unsubscribed.
- *Time of unsubscription*: Time at which the profile unsubscribed
- *Number hard bounces*: Number of incorrect delivery attempts of e-mails in succession
- *Status*: Status field at one's own disposal
- *Change (IP)*: IP-address of the last change.
- *Permission (IP)*: IP-address of the permission determination.
- *Sign out (IP)*: IP-address of the logout.
- *Persona*: Makes the profile for the persona (for article-matching and preview-individualization).
- *Tracking disabled*: Deactivates the personal data storage for the profile. This means that no more profile activity data is collected (neither personalized nor anonymous). This can affect tracking statistics.

Important: The *Persona* field has a special meaning. Profiles marked with this flag are automatically offered for testing. These tests are e.g.: article-relevance or eMailing-test-transmission.

Integration

Here, you will find a list once again of all fields with their unique IDs. Developers mainly use this list to get information on using the TAL functions or XML RPC interface.

Statistics

Displays the development of the pool in different time periods. Here, you can also see a statistical evaluation of the fields within the pool. You can also display this evaluation in a report by attaching the pool.

Field evaluation

Here, you will find a list of the single and multiple selection fields available in the pool with their characteristics as well as the distribution to the profile in the pool.

Map

By clicking on the *Map* tab above the profile list, you can display the geographical position of individual or all profiles on a map. In so doing, it is possible to zoom in down to the house number level. The geographical information is generated from the address data if available. If no address data is available, the IP address of the subscription is used if the profile was entered via a form.

Important: For the map display you need a distinct right in your role. If you do not see the *Map* tab, then you do not have this right.

Import

You can find a detailed description in the chapter entitled Profile import.

Backup

The Backup Center allows you to recover deleted profiles independently for the most part. You can find a detailed description in the chapter *Backup*.

Deletion dependencies

In forms, target groups or Pool Data Miners, used pool fields cannot be deleted. A corresponding note can then be found in the field configuration including a list of the objects where the field in question is used. The same applies, of course, to options of single or multiple selections.

2.7.2 eMailing

 You start an eMailing using an eMailing template in the *eMailings* folder.

The eMailing is created using a corresponding eMailing template with appropriate article templates and a list of articles. If the template is allowed, you can also create in this way a freely designed source eMailing (HTML) in full. When doing so, however, you have to forego the ease of the article templates, but you do have every degree of freedom across all media when designing the eMailing.

When creating an eMailing, you can choose from all newsletter templates available. Global templates (if available) are in a separate tab, here.

Preview

diverse Ausgabe-Formate: Für jede Art der Ausgabe eines eMailings (HTML, TEXT, LANDINGPAGE, PDF, WEB, RSS/Atom Feed) wird eine Option in der Vorschau gelistet.

Content: visual

Here, among other things, you can replace the header image. For this purpose there is a mechanism that accesses the *Header* folder and displays all images contained therein in a preview. The corresponding image is added as header by clicking on it.

In the visual mode you can also easily mount articles right in the eMailing's display or deal with contents of the text fields right where they are displayed in the eMailing. The system presents every article with an easy help menu that is used to move, delete and edit the article - in fact, right in the slot in which it is displayed. You can mount as many articles as you wish in each slot and even change their sequence order in the slot. You can even specify a target group for every article as well. These articles are only displayed to recipients within this target group. With this kind of customization, you can, for example, integrate additional information just for a specific group of recipients in your eMailing. (compare customization in the article).

Furthermore, it is possible via drag & drop to sort articles within a slot and to move them between slots as well. If you move an article between two slots, the article is adapted automatically to the standard article template of the new slot. In addition, the article images can be replaced by new images simply via drag & drop.

Slot configuration

A stylized gear is to the right of the slot name. This can be used for automatically sorting the articles based on specific criteria. See also *Profile Matching and Profile Tagging*

Article configuration

Next to the internal name of the article on the right is another stylized cogwheel. With this you can edit the article or remove it from the eMailing. Furthermore, you can set individual target groups or article templates here.

Content: Content

Here, in a list view you can edit and manage all slots with their article templates as well as the articles along with their target groups that may possibly vary. Here you can later assign to the articles an article template that varies from the standard template or a special target group.

To assign the template, remove the checkmark in the *Standard templates* checkbox. An overview of the article templates that can be used will open. Select the template you require and the article will be displayed in the eMailing with this new template.

To assign a special target group, click on the object picker "For all profiles". A list of all available target groups will open sorted by date modified. Select the target group required and the article will only be sent to this target group during transmission.

The enhanced editor also allows you to edit the text fields used in the template directly. In the lower area, you will find the corresponding input masks.

Content: Settings

Here you can specify additional content or make settings if a container type was specified in the eMailing configuration.

Configuration

Here you can set a wide variety of basic configurations in the eMailing:

- **Sender:** Emailing's sender (which can be seen in the email client).
- **Customized Salutation:** Customized salutation on the basis of the *Salutation* pool field.
- **Domain:** Dispatch domain that is used for all images and link URLs.
- **Confirmation URL:** URL for the confirmation page after logging out from or logging in to the newsletter as well as for disabling the tracking.
- **Recipient Filter for Transmission:** Option to disenable a wide variety of filters for transmission (do not disenable by default!).
- **eMailing format:** Option of sending an eMailing only as a text version. This option is appropriate for profiles that do not use any graphic email clients.
- **Tracking Pixel via https:** The tracking pixel for tracking an opening is generated with the *https* protocol, which can result in a better evaluation with some mail programs.
- **Test Transmission Prefix:** This text precedes the subject of a test transmission (*Check* tab) to make it easier to recognize if it is a test and not already the actual transmission.
- **Container type:** Here you can select a container type. The container fields are displayed under Contents: Settings and can be filled with content there. To display the contents in eMailing, you can use the getContentContainer valley method (for details see *TAL-API*).
- **Date:** Free date fields for noting a release date or deadline.
- **External Tracking:** Ability to specify the tracking code of an external tracking tool. This code is integrated into all URLS of the e-mailing. If URL parameters (e.g. in external tracking) are to be set via replacement variables, the variables {$TEXTAREA_xx_VALUE} must be used.
- **External Content:** Option to process external content in the eMailing. For example, an RSS-Feed could be transferred to the eMailing in this instance. Read details for this under *Programming in templates*.
- **eMailing Header** Wenn es sich um ein Transaktionsmail handelt, muss ein "List-Help-Link für Transaktionsmail" angegeben werden. Dieser Link ist eine URL, welche auf eine Webseite führt, welche Auskunft über die Begründung gibt, weswegen dieses Mail ein Transaktionsmail ist. Dieser Webseitenlink ist entscheidend, wenn es zu Beschwerden zu dem eMailing kommt.

Subject

With this you can specify the subject line (Subject) of the eMailing. Furthermore, it is possible to define different subject variants depending on target group.

File attachments

Here, you can define one or more file lengths that are to be sent with the mail.

2.7.3 rollout

Here you can configure and execute a rollout if this function is activated in the eMailing template and the eMailing is on the global level. By a rollout this eMailing can easily be copied into several clients. Articles or articles and article pictures can also be copied.

Important: A rollout is only possible in eMailings which are on the global level.

Link tracking

In order to track which links are clicked most frequently in your mailing, all links from the templates and contents of the eMailing are checked by a so-called "Linktracker" and replaced by a separate link (...goto.php...). A click on this link is counted and then leads the clicker to a corresponding target page. If a link should be ignored by the linktracker, simply add the *rel* attribute to your link with the value *notrack*. (Example: *Example*)

Check

Here you will find a selection of different options for checking the eMailing prior to transmission:

- **Display & Spam** provides an extensive check of the eMailing in various mail clients (desktop as well as web), spam filters and mobile terminals. This check involves additional costs.
- **Display & Spam (Quick)** offers a free quick check for checking the contents, layout and spam probability.
- **Subject** provides a free preview of the subject in various mail clients (desktop as well as web).
- **Eye-Tracking** offers an overview of the most likely eye movement of the recipient. This check involves additional costs.
- **Text Analysis** analyses the text of the eMailing for comprehensibility. This check is only available for German eMailings and involves additional costs.
- **Links** shows all default and individual links integrated in the eMailing and checks them for validity, i.e. whether the target can be reached. Correct links are highlighted with a green OK and possible errors are highlighted red.
- **Individualization** displays the representation of individualized and personalized variables for the target group set here (if no target group is selected, all profiles with the flag: *Persona* are displayed). To set it, select a suitable target group and save the object. By clicking on "Preview" at a profile of the target group, the eMailing will be adapted to the data of the profile concerned - e.g. salutation, individualised articles, etc. This allows you to check the correctness of individualizations and personalizations.
- **Test Transmission** allows the transmission to a small target group of "rest recipients" prior to the actual transmission. This allows you to check the display quality, customization/personalization and deliverability for particular email clients under real conditions. A list of emails as recipients can also be entered directly. These are then only customized for a sample profile, however, and thus according to the default preview. The prefix "Test" precedes the subject line of individually transmitted mails so you can distinguish them more easily from the subsequent transmission.

The Eye-Tracking check and free display check are deactivated in eMailings that are purely text.

Transmission

The transmission is initiated by clicking on the *Transmission* tab. This is always *now* in the default setting. This means that the system would send the eMailing immediately after final confirmation of the transmission. In addition to the start of transmission, the *Target Group* to which the eMailing is to be sent is also set here.

During the transmission, the system always uses extended criteria to filter out profiles that are no longer or not yet applicable. The eMailing is not sent to these profiles. This is referred to under the keyword *Target group*.

Please also note here that the time is always set as UTC or you should set an alternative time zone here for which the time entry should be interpreted. The time zone is saved in the system in your user profile and pre-selected again during the next call.

To transmit an eMailing not with the full performance of the system, you can also set the transmission performance here. A change to smaller values is particularly useful if you are concerned that the server might not be able to cope with with increased access to your website in the event of a high volume of transmissions - e.g. a shop system. The set performance is regarded as maximum that may be transmitted and the emails are distributed over a longer period according to the volume recipients. Example: an eMailing to 10,000 recipients is set to 25,000 mails/hour. The transmission takes place within a period of approx. 20 minutes. The selected transmission rate can later be viewed in the statistics.

By limiting the mailing quantity, you can specify the maximum number of recipients of a mailing. The profiles to be sent are randomly selected.

You can have an optimized shipping start time calculated for each individual profile. This only has an effect if previous profile activities are available. Selecting 24h+ selects the shipping time for a single profile at which the profile has made the most clicks in previous emails. This shipping time can be on the day of the shipping start or on the next day. A maximum delay of 47h is possible. By selecting 7 days, the shipping time (see above) as well as the most active weekday of the profile is selected for shipping.

The dispatch confirmation allows you to send a dispatch confirmation to one or more email addresses as well as a statistic link after a defined delay, which leads to the statistic link of the dispatch.

In addition, the expected mail size is displayed here in KB. The mail size comprises the size of the source code of the eMailing as well as any possible attachments.

After clicking on *Transmission Preview*, the entire eMailing is displayed once again in an overview. After clicking on *Start transmission*, a final confirmation prompt appears. When you confirm this, the eMailing will be transmitted from the time that you set.

You can view eMailings that have been transmitted or are expected to be transmitted in the *Transmissions* folder and on the start page in the extended view under *your last transmissions*. If you delete an eMailing from the *Transmissions* folder that has not yet been transmitted, it will not be transmitted.

Important: By deleting a transmission from the transmissions folder that has already been completed, you delete all contents of this eMailing and the recipients will no longer be able to call any links, contents and images.

2.7.4 Transmitted eMailing

Once you have started the transmission of an eMailing, a copy of the mailing is created automatically by the system and moved to the *Transmissions* folder.

As long as an eMailing has been released for transmission but the system has not yet started transmitting it, this is still handled as an *eMailing (Queue)*. These eMailings are displayed for you in the extended view of the desk with the time of the transmission in the area "your next transmissions".

Important: As long as an e-mail has not yet been sent, it can be deleted and the transmission canceled. A canceled transmission can be reactivated simply by sending it again.

You can find detailed statistical information on each transmitted eMailing in the *Transmissions* folder. For this purpose a complete list of the determined data - such as opening rate or click rate - from the transmission is displayed in the preview of the object.

The system provides you information with the statistical data. This includes among other things:

- Progress of transmission - start of transmission, duration of transmission, end of transmission
- Addresses filtered by the system - i.e. not transmitted
- current bounce numbers - failed dispatch due to unavailability of e-mail address(es)
- detailed impression and click statistics.
- Reading time measurement (engagement analysis) How many recipients have read the eMailing or have only scanned it

The transmissions in the overview are made available as live reporting within the first 14 days and then generated only 1 x daily later. During the actual transmission, you can always see the current data.

The live tab within the object to be sent is available within the first 24 hours.

This evaluation can also be presented complete or parts of it can be separated in a report. For this purpose, you can assign the eMailing directly to a report during transmission or first create the report after the transmission. Since it is only a view and the data is always collected, there is also no loss of data for the statistical data.

In addition to the evaluation via the application, the most important parameters of the report are also available as Excel (xls and xlsx format) for download (Tab *Transmission* -> *Excel-Export* top right).

Reading time measurement (engagement analysis)

The reading time measurement (user engagement) tracks how long an opener of an eMailing has opened the eMailing. A distinction is made between:

- t < 2s / Not read (or deleted immediately)
- 2s <= t < 8s / Scanned
- t >= 8s / Read

Industry comparison

Within this statistics tab you will find an evaluation of your data compared with the industry average. The type of industry of the account is specified when creating the client and cannot be changed later.

Important: There is no guarantee for the correctness, completeness and actuality of the comparable data. As a basis of assessment, all campaigns of all customers of the previous year from the corresponding industry are analyzed. Campaigns that have extreme values are excluded.

Clickmap

Here, the links used in the eMailing are marked and the number of corresponding clicks displayed.

These statistics are not available in text eMailings.

Important: Owing to the various applications and systemic differences such as the limited view of the clickmap in comparison to the complete dispatch, the numbers of the clickmap are not compared on a one-to-one basis with the numbers of the link, article or dispatch statistics.

Clients

Here you can see which end devices, browsers and mail programs were used by the recipients for interaction. You can also export individual device, browser, and mail program users.

Conversion

This point is only of interest, for example, for webshops that also sell their products by means of eMailings. Here you will see what sales the recipients of the eMailing have generated.

Scoring

This point is only visible when scoring is activated. Here all scoring activities are listed, which took place by the dispatch.

Export

You can export most statistics and create direct target groups from the information collected - provided that you use a personalized data management.

2.7.5 Event Emailing

 An event eMailing is a special form of eMailing. It is not transmitted once at a set date but is always transmitted to a fixed target group at a set recurring interval that you set. This can be birthday eMailings, for example, or eMailings that should be transmitted to the corresponding profiles in the target group over a certain period of time after the registration. The eMailing is created using a template - as with a normal eMailing - in the *eMailings* folder

 You can recognize a deactivated event eMailing based on the fact that there is another icon on the extended start page and in the list view of the *eMailings* folder.

> **Warning:** In the case of a misconfiguration of the target group, you can very quickly annoy your customers if they receive the same eMailing every day. For this reason, you should test these eMailings or the target group to be used thoroughly.

Transmission

All settings - until the transmission - are identical to the eMailing. The only difference is in the configuration of the transmission time. Here, you also have to define the intervals in which the event mailing is to be transmitted. You start the transmission by setting a checkmark to *active* and by removing the checkmark the transmission is stopped. The benefit of this process lies in the fact that a previously existing configuration will not be discarded once you stop the transmission. Unlike normal eMailing the event mailing is still on its original position and can be reedited there (after deactivation) at any time.

The system displays ongoing event eMailings below the *transmissions* folder. There, you can view the statistics exactly the same as with the normal transmission.

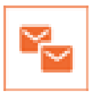
Statistics

The time periods of the integrated opening/click and deregistration statistics are extended up to one year with the event mailing.

Delete data

You can reset the entire statistic, e.g. to delete opening or click data from function tests.

2.7.6 Split campaign

 Split campaigns are created under *eMailings*.

In a split campaign at least two eMailings already available in the system - which normally resemble each other -are transmitted to a specific target group. A text briefly describing the corresponding eMailing or what makes the selected variant special can optionally be specified for each eMailing.

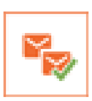
These eMailings, whose number you can define, are sent to some of the recipients during the transmission.

The eMailings can either be split up so that they are first sent only to some of the recipients and the rest are then sent to the winners after a certain period of time (automatic determination of the winner eMailing) or completely to the same parts each (classic split campaign).

Hint: If at least one eMailing that must be accepted by an authorized person prior to the transmission is stopped, then the spilt campaign is also subject to the acceptance process. You can find details on the acceptance process from the information on regular eMailing.

 Split campaigns with acceptance/release necessary for the transmission

 split campaigns are waiting for release

 split campaigns with release granted in transmissions folder

---

The recipient filters for the transmission - such as blacklist - are set directly during the split campaign because these relate directly to the target group to be set for the transmission. Settings made during the eMailings have no relevance for the transmission.

Transmission

The first part of the transmission is identical to the transmission of an eMailing. A target group, the transmission time and speed are set.

When splitting up the test transmissions, set how many recipients the first part of the transmission should be sent to. During the selection, a proportion of 5% to 100% of the recipients is set. This proportion is then distributed to the number of existing emails. E.g. in the case of a proportion of 30% for two eMailings, each eMailing is sent to 15% of the recipients. If 100% is set, a classic A-B split is made for two set eMailings - i.e. eMailings are transmitted to all recipients at the set time.

Hint: During transmission, the profiles are distributed randomly to the individual eMailings.

Afterwards, define the criteria by means of which an eMailing is determined as "Winner". Here, you have statistical parameters available such as the opening rates or conversion rates. If a winner should be determined, at least one criterion must be set. If several criteria are selected, these are then added up.

Hint: An exception to this is the unsubscribe rate. It is not the eMailing with the most but the eMailing with the least number of unsubscribes that wins here.

Then you set how much time the system still has for determining the winner. The standard here is one hour. The decision can also be delayed by up to one day, however.

Hint: If several criteria are selected, the criteria are brought together internally and a ranking list and the winner are determined from this.

Hint: You can set 15 minutes here for the purpose of testing. This is only recommendable for internal use, however.

When the specified time period has elapsed, the eMailing that performed the best in total over all criteria is transmitted automatically to the remaining recipients. If no winner can be determined because the target group e.g. is too small - any eMailing is taken from the list. But bear in mind that a tie is rather unlikely in a real transmission.

Warning: The winner eMailing is transmitted to the target group that was set originally at the start of the complete transmission. This means that profiles that have been unsubscribed in the meantime or are incorrect are still being sent eMailings.

If all settings for the transmission have been made, the transmission can be started after a preview with all settings and a final control dialog comparable with the eMailing. After the start of transmission, a copy is created of the split campaigns and all contained eMailings below the *Transmissions* folder.

Hint: At the start of transmission, the split campaign - like an eMailing - can be attached to a report. Please make sure here that you carry out the corresponding settings yourself for the correct display in the report.

 Split campaigns wait in the transmissions folder until the start of the transmission (compare transmission time).

Hint: If you might have accidentally started a split campaign, then delete this from the *Transmissions* folder.

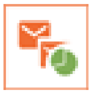
At the set start of transmission, the system now generates the internal information necessary for the transmission. The set filters as well as the blacklists, duplicate check, etc. are first checked in this step. The actual number of addressees is first definite afterwards.

After this step, the transmission to the set recipients of the eMailing starts. If there are still recipients available for the winner eMailing after that, they are transmitted after the set time period has elapsed.

Hint: Owing to the usually very small number of recipients when testing, it may happen that no profiles remain for the transmission to the winner eMailing due to hard bounces, duplicates etc.

Warning: When waiting for the winner, the transmission of the winner eMailing is hindered due to deletion of the split campaign of the transmission. As a result, all information is deleted however - even the eMailings that have already been transmitted! If you do not want a winner transmission to be made, then notify the support immediately.

2.7.7 Transmitted split campaign

 Statistics of the split campaign after starting transmission can be found in the *Transmissions* folder.

After transmission of the eMailings, which are a component of a split campaign, different statistical information is collected by the system automatically. This is necessary for determining a winner.

In the *Transmissions* folder you can see the transmitted split campaigns corresponding to their status. As long as a split campaign has not yet been completely transmitted or is still in the acceptance, it will be indicated by corresponding icons. The statistical data - unlike normal eMailing - is not yet displayed at this stage.

You only see the statistical parameters for the test transmissions by means of which a winner is determined. The criteria which actually determine this are indicated by a checkmark in the headings column. Therefore, this corresponds to the setting of the criteria during transmission. Whilst a split campaign is still waiting for the winner to be determined, you will also see here which eMailing would be the current winner eMailing.

After determining the winner eMailing, this is transmitted and appears above the test transmissions. At the same time, the statistical parameters are displayed. In addition, you can find all eMailings - both the test transmissions as well as the winner eMailing - in separate tabs including a detailed description of all statistical information - just like with a normal eMailing.

2.7.8 Form

 A form is created in the *web form* folder. Forms are pool-linked. This means that when creating a form you have to assign a pool in which the entries are made via the form.

A form is a configuration of fields available in the pool. Thus, for example, you can create new profiles or update existing profiles in the pool. In doing so, you define which fields of the pool are to be used - i.e. filled out. Besides the simple default form - in which the layout is limited to a few options - you can also design an individual form with its own template or with its own CSS instructions if necessary.

In the course of data entry in forms errors often occur, which can be detected and corrected using plausibility checks:

- Validation error: entry of meaningless values in a check field (e.g., an invalid email address)
- Mandatory error: Missing input in a field marked as *mandatory*
- Uniqueness errors: In the case of form errors with a uniqueness criterion, the overwriting of contents is prevented - along with appropriate error message.
- Identification errors: If in the case of forms that validate user data the data entered do not correspond to those saved in the profile, a corresponding error message also appears.

Important: If all standard pool fields used are empty when filling out a form, the form is considered empty and is not entered. However, if this is desired, the following tag must be inserted in the form. <input type="hidden" name="allow_empty" value="1" />

Preview

You receive with the preview a WYSIWYG view of the form - how recipients would see it per browser.

Configuration

Apart from the general information for identification of the form, you define the central configuration of the functions and the workflow for the form here. The most important elements are the *forwarding*, *uniqueness criterion* and *registration procedure*. With these components you control the flow of information during and after the registration. A corresponding *confirmation email* is required for registration forms with newsletters that have double-opt-in procedures, which the system generates and sends automatically by means of a created template.

Uniqueness criterion:

- **-** (no selection): It is not checked whether the profile already exists. A new profile is created each time.
- **eMail**: It is checked whether a profile already exists with the e-mail. If yes, the existing profile is updated. If not, a new profile is created.
- **Profile ID**: It is checked whether a profile already exists with the internal profile ID. If yes, the existing profile is updated. The profile ID is used, for example, when you have forms that are called up personalized - in other words pre-filled with existing data of the profile - and in which no e-mail has to be entered.

Prevent double entry:

If this checkmark is set, no change - if the form determines that the user already exists - will be made to the existing data.

Protect log offs:

If the checkmark is set here, this prevents a profile from being reflagged as registered when a form is sent.

Registration procedure:

- **-** (no selection): no registration procedure –> if the permission is not to be changed - e.g. for surveys or the like. If a new profile is entered, it gets the permission "unconfirmed".
- **Registration for double opt-in**: After registration, the user still has to click a link - which is only valid for this profile - to successfully complete the registration. This link is normally sent via an e-mailing to the e-mail address used for registration. As long as this link is not clicked, the profile remains unconfirmed and cannot be filled.
- **Registration as opt-out**: Here the permission for the profile is indirectly confirmed. This is the case e.g. when existing customers are manually entered in the database via a form. All profiles that are added to the system via an import have the permission "Opt-Out".
- **Registration as opt-in**: The permission is directly granted. An additional reconfirmation - as with double opt-in - is not required. After registration, the user is normally shown a confirmation of the successful registration.
- **Unsubscribe**: When the form is sent, the permission is automatically set to "Unsubscribed"
- **Confirm as double opt-in**: When the form is sent, the permission is automatically set to "Double Opt-In".
- **Only log-in**: Access protection for data that only the user or authorized persons are to see. This is for authentication. After the corresponding data has been successfully input, the user is transferred e.g. to a form with the personal data.

Important: The administrator of the forms is responsible to ensure that correct procedures are observed for generating permissions. The system provides tools for a variety of scenarios, but no standard guidelines for all your scenarios. You can find examples for this in the chapter entitled *Address generation*. Please use these examples and adapt them to your needs.

Address correction:

The automatic address correction checks address data for validity and corrects it if possible. This function can lead to additional costs.

Linking:

An external URL (e.g. a website) or the URL of an object - e.g. another form - can be used in the system as forwarding. The URL Picker with which you can easily select the object (form, web pages) to where the form must be forwarded helps you to use the Object-URL. To do this, click on the button next to the input element.

Important: If the forwarding form contains xxxxxxx, a UID is only attached to the forwarding URL if in the first form: - no uniqueness criteria is selected, - or if the uniqueness criteria is the profile Id, or if the registration procedure equals 'Only login'

Confirmation email:

Here, you can set which eMailing the user should receive after sending the form. If you do not set any eMailing, none will be sent either.

Important: With the *Double Opt-In* registration procedure an eMailing containing the replacement variable {$GRANT_URL} MUST always be sent. The individual confirmation link for each user is entered in this variable for the final newsletter registration. If the user does not receive this eMailing, he will no longer be able to successfully complete his registration.

Target group restriction:

Here, you can define whether the confirmation eMailing should only be sent to a specific target group or to all registrants. This makes sense, for example, if existing users as well as new users can make an entry in the form and only the new registrations should receive the confirmation email.

BCC Recipients:

If an email address is specified, a copy of the confirmation email is sent to the address. Multiple addresses can be entered separated by commas.

Report email

A mailing that is sent to one or more fixed email addresses can be stored here. The mailing is personalized with the data of the profile entered.

Report profile changes:

When activated, changes to a profile also trigger the report - otherwise only new profiles.

Recipients:

Email addresses that should contain the *Report email* can be entered here separated by commas.

Language:

Here, you can set the form language. This means that the name of all standard database fields is translated automatically into the set language. If you do not select any language **- (no selection)**, the system will use the default language of the browser.

Prevent double entry:

Prevents the form from being sent inadvertently several times, e.g. by multiple clicks. Requires JavaScript on the mobile device. Mobile devices without JavaScript can use the form normally, but multiple entries are not prevented.

.. important:: The option inserts dynamic HTML elements in the form source text and is therefore not effective for forms with active individual templates.

Protection against unauthorized entry:

Protects against unwanted automatic form entries. A confirmation cookie is set when the form is opened and checked again when it is sent. This option requires cookies on the end device. Mobile devices with deactivated cookies can no longer send the form. .. important:: The option inserts dynamic HTML elements in the form source text and is therefore not effective for forms with active individual templates.

Activate ReCaptcha:

Activates ReCaptcha support of the form to prevent automated entries.

Auto-Submit:

if the form only contains hidden fields and the profile ID is set to uniqueness criterion, a field will appear here in which you can set a checkmark. If the checkmark is set, this form will be sent off automatically when called without the user knowing anything about this. This makes sense if you want to change additional characteristics of his profile in order to assign him to a special target group, for example.

Important: The legal requirement for using this function should be examined in detail first.

Login form:

All forms that have the registration procedure "Only Login" are displayed for you here. You can then use them as access protection for the actual form. If you retrieve this form, the login form is first displayed and you will only access the actual form after correctly entering the data in the login form.

External tracking:

This code is appended to each URL from the system for use by campaign tracking tools like Google Analytics, eTracker, etc... Do not input "?" or "&" as prefixes.

External content:

Link to an external XML file. For integration of the external content, consult the TAL API documentation: doc:*../developers/tal-api*

Fields

Important: If you want to use your own fields in a form, you first have to create these in the pool. Only then can they be integrated into a form.

The field configuration represents the content-related definition of the form. The field list is divided vertically into two areas: The first area defines all fields of the pool that you want to use - and the second part contains all fields that are not contained in the form. You can provide individual fields with integrated checkboxes by clicking on the button +, other fields can be removed from the form by clicking on -.

Important: When you use fields from the pool in forms, you can no longer delete these in the pool, only edit them. Should fields be deleted from the pool, then you have to ensure that they no longer occur in a form.

Important: The pool‘s mandatory fields always have to be included in the form. You cannot remove them. For this purpose you would have to configure the respective field in the pool differently beforehand.

You specify the *sort order* of the fields with an *X* and *Y* value and thus define where the field is located in the default form. The same *X* values mean the same line. When saving, the system corrects these values automatically and adjusts the positioning of the list accordingly.

The *labeling* field refers to the name for the field in the form. It has no influence on the name of the field in the pool.

Depending on the type of pool attributes in the form, you can remove particular options for single or multiple choice fields from the selection of possible options in the form.

Display

You can select various so-called widgets for displaying the entries in a web form. This enables a web designer to work with appropriate input elements for each case. There are various display options for many pool attributes:

- *Text input*: Direct text input; is displayed as `input type="text"` according to the pool attribute.
- *Text input with trim*: Single-line input (e.g. for e-mail), in which any white spaces are stripped from the beginning and the end.
- *Text input with initial uppercase letters*: All words are begun with a capital letter. For example, the input *on hollywood boulevard* changes to *On Hollywood Boulevard*.
- *Text input all capital letters*, *Text input all lowercase letters*: Inputs are completely changed to capital or lowercase letters.
- *Text input (also blank input)*: Normally, blank inputs in non-mandatory fields are ignored and existing values are not overwritten. With this field, blank inputs are then also imported.
- *Password input*: Text entry with hidden display. **Please note that such a password field does not provide sufficient protection for sensitive data.**
- *Password generator*: Hidden input that generates a password. The password is not generated in accordance with a procedure that makes a dictionary attack difficult, but an easily memorized password is generated notwithstanding.
- *Hidden input*: The default for this attribute is directly transferred. The user has no direct influence on this input.
- *Dropdown*: Single selections are shown using a so-called dropdown selection (`select`). For this widget the first option is always assumed as default (i.e. if another default is not explicitly specified).
- *Date inputs*: Date and time inputs in a dropdown menu; a relative input can also be made the default by using a simple text (see DateDefault). There are also widgets that only allow the input of a date in the future (e.g. reservation wish) or in the past (e.g. birthday).
- *Radio buttons*: Single selections are shown with a list of radio buttons. This single choice can begin with a blank by default.
- *Checkboxes*: Multiple selections are shown with a list of checkboxes.
- *Checkboxes (only attach)*: Only set values are added to the values in the profile. No values are deleted, regardless of what the user checks.
- *Only display*: Here the existing data is only displayed. It cannot be changed by the user.

Important: When using *hidden inputs*, the unused options must be removed in the case of multiple selections - otherwise they will be set blank explicitly. This could cause loss of data because existing data in the profile will be removed.

The selection of a widget also determines how a *default* value can be defined for this field. If a corresponding value is set there, this is displayed when the form is called up. If this default should always be used no matter what data might possibly already be in the profile, you can then achieve this with the *Default priority "default-setting"* in the *advanced settings*.

In the advanced settings, you can define an individual *error message* for most fields, which is displayed in the event of validation errors of field inputs. Validation checks are not possible for all fields of a form, however.

Default Priority

- *Profile data*: Fields are prefilled with the profile values if the profile (profile ID) is recognized.
- *From the call*: Fields are prefilled with values from the URL. **Important:** Please note that the parameter names correspond to the namen/ids of the form fields. In the case of true/false fields, an appropriately named parameter always means a prefill (True). Omitting the parameter does not mean prefilling (False).
- Default setting: Fields are always prefilled with the values specified as default in the field configuration.

Limit

- *Input limitation*: Here you can define a limit for how many inputs can be made via the form. For example, for sweepstakes in which there is an upper limit for participants.
- *Max. inputs*: Maximum total number of inputs before the input limitation goes into effect.
- *Redirection*: URL to which the program redirects in the case of limitation.
- *Time limitation*: Here you can use a limitation to define the time from which this form is to be accessible.
- *Start time*: Here you specify the time starting at which this form is to be accessible.
- *Redirection*: Here you specify the URL to which the program is to redirect before the set time.
- *Time limitation*: Here you can use a limitation to define the time until which this form is to be accessible.
- *End time*: Here you specify the time until which this form is to be accessible.
- *Redirection*: Here you specify the URL to which the program is to redirect after the set time.

Theme

Here, you can configure the visual theme of your default form within a specific framework and determine which messages the system should display for specific input fields. These settings refer primarily to the use of the default form.

Individual template

If you want to use a freely configured form instead of the default form, you can upload an appropriate HTML template. This template may also include TAL instructions. This opens up much freer design possibilities without having to give up the possibility of using simple options from the default configuration. Here, advanced functions such as linktracking or ReCaptcha integration can be managed as well.

Important: When using an individual template, the integration url from the Integration tab must always be inserted manually into the form tag. `<form action="Integration URL">`

Log

In this log all form actions of the last 12 months are displayed. Entries are preserved even if the associated profiles were deleted.

Checkliste: Integration von Formularen

Da Formulare auf unterschiedliche Arten in Webseiten integriert werden können, ergeben sich daraus verschiedene Vor- und Nachteile.

Evalanche Formulare in IFrame:

- Vorausfüllbare Formulare bei Identifizierung des Benutzers
- Progressive-Profiling (bereits befüllte Felder nicht abfragen)
- Schutz vor Fremdeintragungen (CSRF-Token)
- Änderungen nur an einer zentralen Stelle in Evalanche
- separate Gestaltung des Formulars
- schwierigere Anpassung bei mobilen Darstellungen

Integration als Spiegel-Formular:

- Gestaltung des Formular geschieht durch Website
- leichte Anpassung bei mobilen Darstellungen
- keine vorausfüllbaren Formulare
- kein Progressive-Profiling (bereits befüllte Felder nicht abfragen)
- kein Schutz vor Fremdeintragungen (CSRF-Token)
- Änderungen auf Website und in Evalanche

Integration Spiegel-Formular via dynamischer Konfig: *hierfür wird die Formularkonfiguration über die URL:* **INTEGRATIONS-URL,f=1103** *abgerufen und es kann im Anschluss via JavaScript dynamisch ein Spiegelformular erzeugt werden.*

- Gestaltung des Formular geschieht durch Website
- leichte Anpassung bei mobilen Darstellungen
- Änderungen müssen nur an einer Stelle bearbeitet werden (Evalanche)
- keine vorausfüllbaren Formulare
- kein Progressive-Profiling (bereits befüllte Felder nicht abfragen)
- kein Schutz vor Fremdeintragungen (CSRF-Token)

Integration

Here, you can find numerous code snippets with which you can integrate the form into different objects such as eMailings or external web pages.

Flow chart

An overview of the currently set behavior of the form. The configuration is represented as a flow chart depicting the sequence of a registration. The objects in the flow chart can be clicked on. This makes navigation easier and ensures that the correct objects such as target group, confirmation email and linking page are set.

Check

Here, you can find - exactly the same as with eMailing - a link check and eye-tracking check.

Statistics

The following statistical data are displayed:

Öffungen Number of times the form is opened

Eintragungen Number of registrations and the relation between registrations and openings

Links Number of clicks on the links embedded in the form as well as their conversion

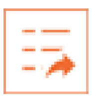
Eintragungsfehler How often did errors occur during registrations because of validation, mandatory fields etc

Formular-Alias Openings, registrations etc. of the form aliases (if available)

Browser An evaluation of the browsers used by those opening the form

Ursprungsseite A list of the pages from which the openers accessed the form (if available)

2.7.9 Form alias


A form alias is in the *web form* folder. When creating a form alias, this is linked to a form that already exists.

With a form alias you can integrate, for example, the same form into different objects - eMailings, web pages - and measure the success of the form separately for each object. The system then assigns the corresponding statistics of the form - such as new registrations, call ups, etc - to the corresponding alias. In this way, separate evaluations are possible for the same form.

2.7.10 LeadPage


The LeadPage is created in the *Web* folder.

A LeadPage is used to display content (e.g. articles, news, product information, etc.) on a website or as a landing page to generate addresses. The LeadPage is created using a corresponding LeadPage template with matching article templates and a list of articles. Like eMailing, the LeadPage has various display formats such as PDF or RSS/Atom feeds. It offers an easy way to integrate articles dynamically on a website without a content management system. In addition to a display independent of the eMailings, you can also use the LeadPage to create an archive of articles of sent newsletters.

The configuration of the LeadPage is identical to the maintenance of the content of an e-mailing in the extended and visual editor. Articles are appended to the corresponding slot of the article configuration. The slot for displaying the articles can be configured for the different media with a different article template.

Preview

various output formats: For each type of output of a LeadPage (HTML, LANDINGPAGE, PDF, RSS/Atom Feed) one option is listed in the preview.

Contents: Visual editor

In the visual mode you can simply add articles directly to the view of the LeadPage or maintain the contents of the text fields directly where they are displayed in the LeadPage. Furthermore you can use the replacement variable: *{$RENDER_FORM:FORMULARID}* to integrate a form directly into a TextArea e.g.: the imprint of the LeadPage (make sure that the checkbox *Set Action Parameters* is set in the configuration of the used form). The system displays all articles with a simple help menu, which serves to move, delete and edit the articles - directly in the slot in which it is displayed. At each slot you can add as many articles as you like and also change their order in the slot. You can also specify a target group for each item. These articles are only displayed to a recipient if he is in this target group. With this type of individualization you can, for example, integrate additional information into your LeadPage that is only visible to a certain TargetGroup. (see Individualization for Articles).

Furthermore, it is possible via drag & drop to sort articles within a slot and to move them between slots as well. If you move an article between two slots, the article is adapted automatically to the standard article template of the new slot. In addition, the article images can be replaced by new images simply via drag & drop.

Slot configuration

A stylized gear is to the right of the slot name. This can be used for automatically sorting the articles based on specific criteria. See also *Profile Matching and Profile Tagging*

Article configuration

To the right of the internal name of the article is another stylized gear. This can be clicked to edit the article or remove it from the lead page. Individual TargetGroups or article templates can also be entered here.

Contents: Enhanced editor

Here you can edit and manage all slots with their article templates as well as the articles with their possibly deviating target groups in a list view. You can also assign an article template that deviates from the standard template or a special TargetGroup to the articles at a later date.

To assign the template, uncheck the checkbox *Standard templates*. An overview of the usable article templates opens. Select the desired template and the article will be displayed in the LeadPage with this new template.

To assign a specific target group, click on the object picker "For all profiles". A list of all available target groups will open, sorted by modification date. Select the desired target group and the article will only be displayed to profiles from this TargetGroup.

The enhanced editor also allows you to edit the text fields used in the template directly. In the lower area, you will find the corresponding input masks.

Configuration

Here you can see the used LeadPage template. Click on the name of the template to open it for editing in a new tab/window.

Scoring

Here you can specify scoring values for the *LeadPage opening* and the *LeadPage click* that deviate from the standard definition.

Integration

Here you get the direct URL to the LeadPage and some further display variants e.g: Leadpage in the IFrame, as well as the URLs to the RSS and Atom Feeds.

To call up a lead page and retrieve the current geocoordinates of the client and store them temporarily in the profile, there is a URL with extended parameters. The following parameters can be used:

- **f=28:** Causes the geocoordinates to be collected first and then to switch to the regular view (f=7).
- **vp=3600:** Indicates for how many seconds the current coordinates should be stored (default is 3600 seconds).
- **lat=XX:** Latitude for manual specification of coordinates.
- **lon=XX:** Longitude for manual specification of coordinates.

Important: The automatic recognition of the geocoordinates depends on the settings of the client browser and, if necessary, the operating system.

Check

Here you will find a selection of different ways to check the functionality of the LeadPage:

- **Individualization** shows the representation of individualized and personalized variables for the target group entered here. To set, select a suitable TargetGroup and save the object. If no TargetGroup is selected, the defined personas are offered for each pool. By clicking on "Preview" at a profile of the target group, the LeadPage will be adapted to the data of the profile concerned - e.g. individualized articles, etc. This allows you to check the correctness of individualizations and personalizations.
- **Eye tracking** provides an overview of the probable eye movements of the visitor. This check is subject to additional costs.
- **Text analysis** serves the textual evaluation of the content of the LeadPage for comprehensibility. This check is only available for German-language content and there are additional costs.
- **Links** displays all standard and individual links integrated in the LeadPage and checks their validity - i.e. whether the target can be reached. Correct links are marked with a green OK, possible errors are marked in red.

Statistics

- **Statistics** here you will find the number of clicks directly on the LeadPage and its landing page, as well as the RSS and Podcast version.
- **Article** shows the number of clicks on different areas of the article as well as the frequency of use of cross-media content such as the read aloud function.
- **Social** offers an evaluation of how often articles were distributed in social networks and also how often they were called up in social networks.
- **Forms** displays the number of openings, submissions and, if necessary, resulting new entries of an integrated form. Forms are replaced with the replacement variable: {$RENDER_FORM:FORMULARID} are integrated into the LeadPage template.

2.7.11 Website

 A website can be created in the *Web* folder or in a subfolders created by you. It only contains HTML code. A website can serve as a landing page for an article if the display options of the templates are not sufficient for this because the page has to be designed very individually, for example. You can also use a website to replace the standard confirmation pages for the subscription or unsubscription of newsletters by confirmation pages in corporate design.

2.7.12 Article

 An article is created in the *Article* folder.

An article is an easily maintained information unit comparable with a newspaper article. It consists of text parts and one or more optional images - specified by a *Article type*. The article acts as abstract storage of information. It can be edited by the user without knowledge of the subsequent display. Thus, the input is always neutral. The display is not determined till later via the usage site. At the same time, the article template is used to define the display as in a newsletter, for example.

Initial generation

To create a new article, click on the + *New* button in the list view of the *Article* folder. If there is more than one article type in the system, all usable article types will be displayed for you in the next step. After selecting one, you will be brought to the input mask of the contents. If there is only the standard article type in the system, no selection will be necessary and you will be taken directly to the input mask.

Article

Here, all fields of the selected article type are displayed grouped in tabs. The tabs and fields displayed is dependent on the settings of the article type.

The default article type provides a tab with base fields like the *Headline*, an *Picture*, the *Short* and *Longtext* and a *Link List*. In addition, there are other tabs with their own input options for adapting headings, images and texts to the respective display formats (PDF, RSS, social networks, etc.).

If you want to refer to an existing landing page (e.g., website) instead of to the long text of the article, the *Landing Page Link* field can be used for this purpose. However, the article template used must take this into account.

With the *Hit Checkpoint* selection, the clicks on the links in the mailing can be assigned an existing checkpoint and counted in order to view the long text.

Using the function *Text quality* (below the check tab), it is possible to check automatically the readability of your text inputs, for example, based on different criteria.

Warning: If this detailed evaluation is used, this will result in additional costs.

Important: If you copy text from external sources, then depending on the operating system and the source application, HTML transformations may be used. In order to avoid problems with this, the HTML editor in the system provides the option of importing text directly as plain text or from Word. For all media, the system will only interpret the HTML tags that are actually permitted in the editor as well. All other data included from different sources fall under the responsibility of the editor. A correct display outside the HTML view (text email, PDF, ...) cannot be guaranteed.

Profiling

In the profiling tab you can set the interests that this article is relevant for. *Profile Matching and Profile Tagging*

Persona

In the persona tab you can see how well the content of the article relates to individual personas. This requires that the article possesses criteria that has configured relevance and the profile of the pool is marked as persona.

Customization

When using articles in an email, an alternative article or a string of alternative articles can be used instead of the actual article and also depending on the target groups. If you specify them under the item *Customization* and set a target group for each of them, you will get a finely adjusted customization of the contents. This technique is appropriate if you want to ensure that an article is always to be displayed, though a specialized article should sometimes be given precedence in the case of particular target groups. An example application would be a general article on sport. If the salutation - i.e. woman or man - is known, an article for women or men should be displayed instead of the general article.

Important: As individual articles, no article can be used, which in turn, was individualized. It is possible, however, to use the single individualized article dependent on the target groups and thus perform further individualizations.

Translate

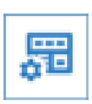
In the Translate tab you can select the languages into which you want to translate an article and start the automatic translation.

further information can be found here: `../getting_things_done/translation`

Statistics

Here you can see in which news managers or sent eMailings the article is integrated. After clicking on the "Details" button, you will be shown further statistical parameters.

2.7.13 Article type



An article type is created in the *Article types* folder. You can find this in the left-hand folder navigation in the *Settings* folder.

This object type is a container containing a *Article configuration*. This allows articles to be freely configured in full - beginning with the data fields and up to labels, help texts and entry limitations.

The article type represents the actual configuration of the article. The fields that should appear in the subsequent article object are saved, edited and grouped here.

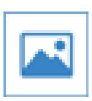
The following field types can be added to an article:

- One-line input - Provides a simple, one-line input
- Multi line input - Generates an input field for multi line text
- Multi line input (formatted) - like the multi line input but integrated in a WYSIWYG-Editor
- Multi line input (HTML-Editor - like the multi line input (formatted) but with a complete WYSIWYG-Editor
- Single selection - Generates a dropdown list
- Multiple selection - Generates a checkbox list
- Boolean (true/false) - Generates a checkbox
- Image - Generates an image picker
- Image (List) - Generates a list with images
- Url - Generates a one line input with URLpicker
- Links (List) - Generates a list with links
- Date with time - Generates an input option of data and time with calendar
- Colour - Generates an input field for a color with color picker
- Container - Generates a container-picker
- Container (List) - Generates a list with containers
- Values (List) - Generates a list of values

Furthermore, there are also special field types intended for the design and documentation of the article type.

- Documentation - Displays the text that was entered in the default field
- Separator - generates a line for separating the fields

2.7.14 Image

 An image is created in the *Images* folder.

It can be integrated into different objects - eMailings, articles, websites etc.

Preview

Bild Direct preview of the image. You can display the image in three supported image formats using the dropdown menu.

Information / Image

To upload an image, enter an internal name as well as either a *File* or a *URL*. If required, you can upload a group of images. To do this, click the + button and enter a *Name* and either a *File* or a *URL* here too. Every file is created as a new image object. The system normally supports various image formats. If contrary to expectations, problems should occur with an image format, try uploading using another format if necessary. The file size of the image is limited to 10 MB, regardless of whether it was uploaded when the object was created or by drag & drop.

Important: The system only permits images of the .jpg, .png and .gif type

Editing

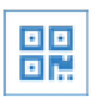
With this function you can crop an uploaded image to a specific area, reduce the size as well as rotate it. If you use this function, you change the uploaded image irrecoverably. The size of an image is limited to 5000px x 5000px after editing.

Integration

To integrate an image in an article template or freely in the template, let the system generate a dynamic or static URL here. At the same time, various options can be set in order to scale or crop the image, for example. Furthermore, it is possible to round off the corners or to represent the image in grey scales.

Important: If images are still being used in objects - e.g. articles or eMailings - they cannot be deleted.

2.7.15 QR-code

 A QR-Code object is created in the *Images* folder.

The QR code is a two-dimensional barcode that can include texts and links. This barcode can then be photographed and evaluated with the cameras of current Smartphones (a bar code Scan App might possibly be needed).

Preview

Direct preview of the text entered as QR code.

QR-code

Here, you store the text to be converted. When doing so, you can also enter personalization variables of the profile to personalize the QR code. You can find a list of the variables available in the dropdown menu.

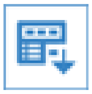
If the personalization variables in a profile are empty, a fallback text - if defined - will be displayed as standard text. If no fallback text is stored, the QR text is always displayed.

Important: The number of characters in a QR code object is limited to 600

Integration

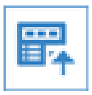
Here, you will find the link to the QR code object for integrating the QR code in eMailings, websites etc. The image can be enlarged if necessary by means of the optional *width* parameter. A reduction of the image is not possible, since the code would otherwise be illegible for the cameras.

2.7.16 Article export

 **Article Export** is created under *Web*.

An article export enables you to export articles from the system as RSS feed. The articles set in the *Article* tab can be used as the source for external feed readers by means of the URL from the *Integration* tab for display on your own homepage, for example.

2.7.17 Article Import

 **Article Import** is created in the *Web* folder.

With an article import, not only can you import articles within the system from one client to another, for example, but also from outside simply by specifying a URL to an RSS or Atom Feed. After configuration of a corresponding URL for the feed, the articles of the feed are displayed in the Preview and then can optionally be imported. To do this, you select the articles and click on *Import selected articles*. If the import was successful, you can now view a list of every article that can also be clicked on directly for further editing. The imported articles are copies of the originals so you can extend the articles yourself.

Article type

An article type has to be saved in order to import an article from a feed. The data of the feed (*heading*, *image*, etc) is then assigned to the fields in the article by means of the set roles in the article type.

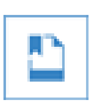
Tip: Simply use the Feed reader to import from your Content Management System article. Popular systems like *Typo3* or *Web-Edition* provide direct news feeds.

Tip: Make content centrally available to your clients or departments. Create an RSS/Atom Feed where you mount all of the messages of your interested parties and then communicate the URL. Then communicate the URL.

Important: When importing articles, you should always be aware of an article‘s copyright. If in doubt, indicate the source being cited.

Important: If the specified URL has errors, then the system will display them as well. Therefore, please first check in advance whether it is a valid feed. The system provides a direct link to the so-called Feed Validator of the W3C for this purpose.

2.7.18 File

 A file object can be created in the *Web* folder or in subfolders that you have created. This involves a direct upload of a file from a computer to the system The file is then returned back from the system by calling the link via

the *Integration* tab. During the upload, the system attempts to detect the file type (PDF, jpg, etc.) and then to return the file in this form as well.

A typical sample application is the hosting of additional files for a newsletter directly on the system., e.g., a PDF document or a ZIP file without having to provide a download for it in the Internet.

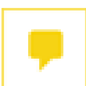
After uploading the file, you can still customize the file type yourself and configure the display type after clicking on the link:

- As download - with a prompt as to where the file is to be saved on the recipient‘s computer
- Or with direct display of the content.

Hint: Store files like style sheets easily and directly in the system. Make sure that the file type is set correctly (e.g.‘‘text/css‘‘) and that the download is *deactivated*.

Important: Note that the system reassigns the same name to the file when downloading, which it had when uploading.

2.7.19 Note

 A notice can be created below the *Notice* folder or in subfolders created by you. Internal documentations and descriptions of objects can be set up using notices.

A variety of objects support the direct assignment of a notice. Consequently, direct descriptions / work orders can be attached directly to an object.

Preview

View of the complete notice or its content.

Note

All core configurations of the notice can be managed within this tab.

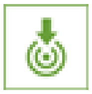
- **Name**: Internal name in the note
- **Headline**: Headline of the note. This value is also displayed in the sidebar / start page when activated
- **Note**: Note text
- **Display in page navigation**: If this option is activated, the current note is displayed in the tree structure on the left in the “Notes” area as well as in the expanded view in the area “Your documents and notes”.

Links

All objects that the current notice is linked to are displayed here. Existing links can be removed immediately here, or new links to objects created.

You can tell whether a notice was assigned to an object from the object itself as well as in the overview lists by means of the corresponding icon.

2.7.20 Target group

 A target group is created in the *Target groups* folder.

The target group is similar to a *Persona* a user-configured selection of profiles. A target group configuration consists of any number of comparison criteria (properties such as *hobby* or *talk*), which you can logically link together. This can also be done across pools. Target groups are used for the following areas:

- **Transmission**: The target group defines the recipient group (e.g. Test addresses, all addresses, special groupings etc.)
- **eMailing customization**: You can display and hide certain articles depending on the target group.
- **Article customization**: You can replace articles by alternative articles depending on the target group.
- **Inquiry Management**: You can redirect or deliver contents of web form queries depending on the target group.
- **Smartlink**: Depending on the target group, it is possible to route to a specified URL.
- **Pool-Data-Miner**: You can delimit special statistical evaluations for a target group.

Important: Target groups are a filter! No profile is assigned directly to a target group. The profiles are only delimited from a target group by means of their characteristics. Several target groups can contain the same profile. Each modification of a profile can immediately influence the membership to a target group.

Preview

Various statistics on the progress of registrations and profile numbers are represented here as well as the configuration details for each pool.

Configuration

The generation of a target group can proceed across several pools. If there is only one pool, then this option is unavailable. After a pool is added, it is initially configured in such a way that every profile contained is transferred to the target group as there is no limitation criterion yet at this time. You can modify the profile itself by attaching additional criteria. New criteria can always be added after any criterion by clicking on the 'plus' in the respective line. Any lines can also be deleted again by clicking on the 'minus'. In the case of more than one criterion, you can link them logically using a connecting operator and bracket them appropriately.

Important: The operators *And* and *Or* behave neither in a mathematical sense nor in a sense familiar to programmers. Please always use brackets when mixing both operators.

Important: In der Konfiguration werden ab und an auch Profil-Zahlen angezeigt. Diese werden jedoch aus Performancegründen nur vereinfacht berechnet (geschätzt). Es werden u.a. keine Verschachtelten Statements ausgeführt. Die Zahl kann daher von der (immer korrekten) Zahl in der Zielgruppen-Vorschau abweichen.

There are the following criteria for the configuration of target groups:

- **Pool:**
  - *All entries in the pool: All profiles in the pool.

- **Behavior:**
  - *E-mailing received*: Assignment of whether a profile has received a particular e-mailing.
  - *E-mail openings*: Assignment of whether a profile has opened a particular e-mailing.
  - *E-mailing hard bounce*: Assignment of whether a profile has been designated as a hardbounce in a particular e-mailing.
  - *E-mailing soft bounce*: Assignment of whether a profile has been designated as a softbounce in a particular e-mailing.
  - *E-mailing landing page*: Assignment of whether a profile has viewed a landing page of a particular e-mailing.
  - *E-mailing multiple openings*: Assignment of whether a profile has opened a particular e-mailing multiple times. (Impressions > 1)
  - *E-mailing clicks*: Assignment of whether a profile has clicked a link in a particular e-mailing.
  - *E-mailing multiple clicks*: Assignment of whether a profile has clicked several links in a particular e-mailing. (Clicks > 1)
  - *Shipping frequency*: Assignment of whether a profile has received more or less than X mailings in the last Y months.
  - *Received last e-mail*: Check when a profile was last e-mailed.
  - *PrintMailing received*: Assignment of whether a profile has received PrintMailing.
  - *PrintMailing erroneous*: Assignment of whether a profile should have received PrintMailing but could not be delivered.
  - *SmartLink*: Assignment of whether a profile has gone through a SmartLink.
  - *SmartLink target*: Assignment of whether a profile has arrived at a particular SmartLink target.
  - *WebTouchPoint website*: Assignment of whether a profile has visited a specific Web page.
  - *LeadPage opening*: Assignment of whether a profile has opened a LeadPage.
  - *LeadPage click*: Assignment of whether a profile clicked in a LeadPage.
  - *Formular abgeschickt*: Zuweisung, ob ein Profil ein Formular abgeschickt hat.
  - *Scoring*: Assignment based on a field of the Scoring Matrix (A1 to D4).
  - *Profile Scoring*: Assignment based on the Profile Scoring value.
  - *Activity Scoring*: Assignment based on the Activity Scoring value.
  - *Persona Matching*: Assignment of whether a profile is assigned to a specific persona.
- **Basis:**
  - *Entry date*: Time at which the profile was created.
  - *Date of change*: Time at which the profile was last changed. For new profiles this point equates with the creation.
  - *Unsubscription date*: Time at which the profile was unsubscribed, e.g. following an e-mailing.
  - *A/B group split*: Every profile is assigned to a group A or B; there is no overlap within these two groups and the distribution is 50% on average.
  - *Language*: The system transfers the language transmitted along with the profile in the form and saves it with the profile data. With this, one can subsequently address a profile in its correct language.

   - *Origin form*: Assignment of whether a profile was generated via a particular form or form alias. **In this case, only new inputs are considered - changes via a particular form are ignored**
   - *registration procedure*: Assignment of whether a profile has a certain permission (*Caution: The use of this statement cancels the shipping protection due to the permission!*).
   - *Incorrect e-mail address*: Assignment of whether a profile is faulty and and is no longer being filled (three hardbounces).
   - *Number of Hardbouncese*: Checks how many hardbounces a profile has already created.
- **Miscellaneous:**
   - *Area of*: Assignment whether a profile within an area with a specified radius is at a specified point. The coordinates are specified in decimals and not in minutes and seconds.
   - *Checkpoint*: Assignment of whether a profile has been run via the named checkpoint.
   - *Target group*: Use of other target groups in the current target group. **Attention: No multiple nested target groups are considered (target group in target group in target group)**.
   - *E-mail duplication*: Provides all profiles that occur two or more times.
   - *Number of options*: Provides all profiles with a special number of selected options (only applies to multiple selections).
- **Profil:**
   - *Permission*:
   - *Persona*: Assignment of whether a profile was marked as "persona".
   - *Change (IP)*: Assignment of whether a profile was last changed using a specific IP address.
   - *Discharge (IP)*: Assignment of whether a profile was "unsubscribed" via a specific IP address.
   - *Permission (IP)*: Assignment of whether a profile was "confirmed" via a specific IP address.
   - *Deleted*: Assignment of whether a profile is marked as "deleted" (revoked permission).
   - *Tracking disabled*: Assign whether a profile is marked with a tracking revoke.

0 Extruded profiles:

- *Unshipped without shipping protection*: Checks for discharge and *removes shipping protection*.

Important: If a field is to be checked for an IP address, note that a CIDR suffix is always required.

Examples: 192.168.2.7/32 describes a single IP address 192.168.2.7 192.168.2.7/24 describes all IP addresses from 192.168.2.0 to 192.168.2.255

Apart from these internal fields, you can also configure target groups by means of all other fields available in the pool

---

Important: Please be aware of the following points during the use of particular target group options:

If a target group is used for the transmission, the following options are automatically applied via the system:

- **Registration procedure** is not confirmed
- **Incorrect email address** does not exist
- **Unsubscribe** does not exist
- **email** includes @

If it is desired that these options are not to be used during transmission, the appropriate options have to be explicitly specified in the target group.

In addition and depending on the pool‘s configuration, there are still other comparison functions. The respective options can be selected for all single and multiple choices.

Profiles

This is a list of all profiles that comes under the target group configurations currently saved. This list is used to quickly check whether certain profiles exist in the target group. It can also be used in conjunction with an extended configuration with a form as a front desk application in which the employees are only allowed to process a specified list of profiles.

Map

Below the *Profiles* folder as well as at pool level you can visualize the profiles on a map. A zoom function until house number level is therefore possible.

Important: For map display a right of your own in your role is required. If you do not see this tab, then you are lacking the right.

Delete profiles

Removes profiles permanently from the pool. The target group is also used here for filtering the profiles concerned. In this way, for example, you can delete all incorrect email address from a pool once a month. Follow the instructions in the dialog to delete the profiles permanently.

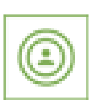
Important: The deletion of profiles removes these permanently ** from the pool** (and thus from the target group as well). The profiles will also then be missing in every other target group, however. It is possible to recover the profiles in the *Backup*

Deletion dependencies

Like every object, the dependency of target groups on other objects is always checked during deletion. Particular attention should be paid to their dependencies on transmitted eMailings. Thus, target groups used in the subject configuration and as a transmission target group are regarded as a hard dependency for up to two weeks after transmission and can therefore not be deleted. A soft dependency is displayed here for up to four weeks after transmission.

hidden persona

2.7.21 Persona


Personas are created under *Target Groups*.

A persona represents a target group-like profile filter, which is precalculated in contrast to a target group. Another difference is the *matching threshold*. This indicates the percentage of matching criteria from which a profile is to be assigned to a persona. This makes “fuzzy” definitions possible.

A *matching threshold* of 100% means that all criteria must be met. A *matching threshold* of 0% (deactivated) means that all criteria are ignored and all profiles from the *data source* are also assigned to the persona.

In general, a profile can be assigned to any number of personas.

The following field types are currently available:

- single selections
- multiple selections
- Salutation (system field)

Important: In order to guarantee a performant and fluid workflow even with complex persona configurations, the persona affiliations of the profiles are precalculated overnight. If a profile is changed, e.g. by a form, a new assignment of the profile is triggered immediately.

---

field configuration

In addition to the criteria to be fulfilled, a *weighting* must be configured for each criterion. This specifies how large the share of the criterion in the final result should be. A *weighting* of 100% means, so to speak, that this field is included in the calculation with 100 percentage points.

The sum of all weightings gives the maximum possible percentage points that a profile can achieve. Multiplied by the *matching threshold*, the result is the number from which a profile is assigned to the persona.

If several options are selected for a merge selection, the total weighting of the field configuration is evenly distributed among the individual options. With 4 selected options in the persona, each option in the profile would correspond to a quarter of the weighting, with 100% weighting 25% per option.

Important: To be able to use your own fields in the field configuration, the checkbox "Profile Matching" must be ticked for the respective field in the pool configuration.

---

purpose of use

Personas can be used around:

- Move profiles into automated campaigns or check within a campaign whether a profile is assigned to a persona.
- To select profiles in simple target groups in order to realize e.g. article individualizations.

2.7.22 Inquiry-Management

 Inquiry-Management is created in the *Inquiry Mgmt.* folder.

With the management query you can forward frozen queries to the corresponding departments (responsible person, Account Manager etc) for further processing via forms. Queries can have a connection to both a pool's existing and new profiles. Moreover, inquiry-Management is also useful for generating backups of every profile or synchronizing profile data with an external CRM system.

Configuration

In addition to the system default fields, the inquiry-Management configuration includes the actual configuration of the object. Here, you can configure time intervals in which reports are to be transmitted to specific recipients and which profiles the system should contain. To determine precise specifications for the configuration, example cases can be found under *Working with the System*.

Fields

Here, you not only configure the file type for a grouped transmission, but which fields in the file are to be included as columns, too. All fields are defined directly and correspond to the fields of the selected pool. *Internal Fields* references every field saved for a profile, but is not configured directly in the pool. They are an internal ID, the permission and additional fields among others.

eMail

These settings specify the appearance and the addressees of the email sent. Here the pool‘s placeholders for the text can also be used (if a single report is involved). For a report from recipients, depending on which options for the profile are selected from an open field, who is to receive a report in the case of which options can also be configured here under *Grouping* after selecting a single or multiple choice field. The profiles are only then communicated to the recipients who apply to the options. In the case of multiple choices, the emails are sent for every applicable option.

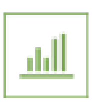
Important: If queries do not arrive and there is nothing to report, then an email will not be generated and sent either.

Report email

A separate mailing can also be set as a report email.

Important: Bei allen gruppierten Versandarten muss die Ersetzungsvariable „{$REPORT_BODY}“ in der eMail verwendet werden, da sonst in der versendeten eMail u.a. kein Downloadlink für die Profilliste angezeigt werden kann.

2.7.23 Reporting


Reports are saved in the *Reports* folder.

A reporting is a central evaluation of difference statistics of various objects from the system. You can assign the following objects to a reporting:

- *Pool* (Time restriction filters profiles by entry date)
- *Pool-Data-Miner* (Time restriction filters profiles by entry date)
- *Target group* (No time limitation possible)
- *eMailing* (Only shipments sent within the time limit)
- *Split-Campaign* (Only shipments sent within the time limit)
- *Event-eMailing* (no time limitation possible)

- *SmartLink* (Time restriction filters tracking entries)
- *Checkpoint* (No time limitation possible)
- *Formular* (Time restriction filters by entry and opening date)
- *Leadpage* (No time limitation possible)
- *Funnel* (No time limitation possible)

You can display as many objects in a report as you wish. The objects themselves or specific elements of their statistics can be displayed or hidden as required. These are e.g. enhanced graphic representations. You can compare certain objects (e.g. eMailings) with each other using the *Compare* checkbox. The result of a reporting is displayed as a complete page - with the list of all objects one below the other and their evaluations. At the same time, it is also possible to display certain objects anonymously (without labeling) - e.g. the Pool Data Miner.

Pool

The system displays the evaluation of a pool as a contingency table. Which fields in the pool have been created as single and multiple choice can be observed in the process. For each of these options the system indicates how many profiles have set this option. From which form the entries with the respective options originate is displayed via the forms. If required, you can have graphics shown for the options, which display a relative and absolute allocation of the options across the entire profile count.

Target group

With this, one can have the particular number of profiles in the target group be displayed. Optionally, the reporting displays them directly as well.

Pool-Data-Miner, eMailing, LeadPage, Form, SmartLink and Checkpoint

This display corresponds with the object‘s respective statistics.

Split campaign

Corresponds to the settings of the eMailing. You can also decide whether the data of the winner and/or of all test transmissions should be displayed. The representation can only be defined once for all displayed eMailings, however.

2.7.24 Pool Data Miner


Pool-Data-Miner are stored under *Reports*.

The Pool-Data-Miner is a tool for visualizing data of a pool - i.e. the profiles - and their characteristics according to their distribution. At the same time, most of the pool fields can be combined in different ways. In each attachment, these fields are nested more deeply into a hierarchy level. The sum of the profiles with the corresponding characteristic is then displayed for each hierarchy level in the preview or in the reporting. Each level is further represented with a bar showing the level‘s proportion of the entire group of profiles. If you do not want to view the complete pool and thus all profiles of the pool, you can also specify a target group to be used instead.

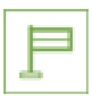
The following pool field types are supported:

- Single or multiple selections with their options

- Number fields with:
  - Value (e.g. 1, 3, 39, 204)
  - Range in tens (e.g. 0-9, 30-39, 200-209)
  - Range in hundreds (e.g. 0-99, 200-299)
  - Range in thousands (e.g. 0-999)
- Email address with domain
- Date entries
  - Month and year (e.g. 04/2012, 05/2012)
  - Month (e.g. 04, 06, 12)
  - Day of the month (e.g. 25, 31)
  - Day of the week (e.g. Monday, Friday)

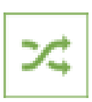
Tip: If the Pool-Data-Miner is used in a report and anonymous display is set there, one or more lines can be highlighted by request. Extend the link found in the *Integration* tab by `?pdmhl[]=pdm_34223_0_4736` (as often as you wish). You can best get the value for the variable from the preview (HTML-Code) or generate it from the IDs of the pool fields concerned and their options.

2.7.25 Checkpoint

 A checkpoint is created under *Tracking*.

The checkpoint is used for additional tracking of user activities and can be regarded as a universal object. You can use checkpoints to measure openings (image tracking) and to track links as well. If, for example, you are tracking a list of links at a checkpoint in the eMailing, you can query every profile that has passed across this checkpoint later in a target group. At the same time, the checkpoint detects single and multiple calls and can also be used in locations outside the system - e.g. as additional counters for AdWords campaigns.

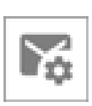
2.7.26 Target group link

 Target group links are created under *Tracking*.

A target group link is used for linking to different pages - depending on membership of the profile to another target group. A click on this link leads to different targets - e.g. a page for women and another page for men. If, however, a profile is not assigned to any target group or is unknown to the system, a general fallback link is then used.

Target group links are therefore usable simply for customization based on profile data - even for linking to pages that are outside of the system.

2.7.27 Emailing template

 Newsletter templates are created under *Templates*.

A newsletter template is the basis of every eMailing and can already be populated with static contents - such as the editorial or imprint - in advance. With a newsletter template you specify the display types of an eMailing. A copy of this template is created during the creation of an eMailing and the actual template is not affected by any changes to the eMailing.

Preview

Here you can view directly and check every task format that has been configured.

Configuration

Here you define the basic settings and configuration of the eMailing.

- Sender
- Reply to email address
- Personalized salutation

In addition to the default contents, a container type can be selected in the configuration of the eMailing template. The field configuration of the container type creates a tab in eMailing: Contents/Settings. There the contents can be filled.

See also *eMailing*

Automatische Kopie: Geben Sie eine oder mehrere (kommagetrennte) Email-Adressen an. Jede Mail, welche an ein Profil versendet wird, wird nun auch als Kopie an diese Adressen versendet. Dadurch vervielfacht sich die Anzahl der versendeten Mails. Die Adressen müssen eine Einwilligung erteilt haben und existieren (Andernfalls wird für jede generierte eMail ein Hard-Bounce generiert).

eMailing Header

Die Option "Als Transaktions-Mail nutzen" ermöglicht es, eMailings auf Basis dieser Vorlage zu erstellen, welche keinen Abmeldelink benötigen, da es sich nicht um werbliche Inhalte, sondern sogenannte Transaktions-Mails handelt.

Wenn kein gültiger Abmelde-Link im Mail-Head verwendet wird, muss stattdessen unter "List-Help-Link für Transaktionsmail" ein List-Help-Link im Mail-Header angegeben werden. Dieser Link ist eine URL, welche auf eine Webseite führt, welche Auskunft über die Begründung gibt, weswegen dieses Mail ein Transaktionsmail ist. Dieser Webseitenlink ist entscheidend, wenn es zu Beschwerden zu dem eMailing kommt. Die URL welche in der Vorlage angegeben wird, kann in einzelnen Mailings abgeändert werden.

Die Option "Alternatives Abmeldeszenario (List-Unsubscribe)" ermöglicht es andere Abmelde-Szenarien zu ermöglichen. Genaurere Möglichkeiten dieser Funktion werden in Zukunft bekannt gegeben.

Design

The following display types are possible:

- *eMail (HTML)*: The e-mailing is shown in HTML format in the e-mail client.
- *eMail (text)*: For e-mail clients that cannot or should not display HTML. In this case, plain text is used.
- *Web/Mobile*: The e-mailing is shown in a display optimized for smartphones. The source is HTML.
- *Landingpage*: The display of the expanded contents of an article on the landing page. The source is HTML.
- *PDF*: The e-mailing is shown as a PDF in a print-optimized form.

Important: A separation of the HTML code between email and non-required web displays appears in the first view. However when creating templates, you should always be aware of the fact that the eMailing's HTML code in the various email clients cannot be identical to that in a browser. Therefore, the system does not permit any mixing of the two types of HTML code. This also applies to the *Article Template*.

The internal HTML editor is available in the respective tabs for editing task formats. Checking off *Can be changed in the eMailing* permits a modification of the task format even in the eMailing. Alternatively, an alternative template whose code can then be used for displaying the eMailing can also be set for each type of presentation.

Warning: Also, if you allow editing in the eMailing, configuring slots can only be done in the newsletter template. However, you can influence the positioning of slots in the eMailing via the source code in the eMailing.

- *Static text variant*: This function helps you to generate the text version of the mailing. There are two different generation variants to choose from.

Article templates

This view corresponds to the "Enhanced Editor" in the "Contents" tab of the eMailing. Here, you configure the name of the slots used and define their default article templates. Only slots previously configured in the eMailing template that was used can be populated.

If it should be necessary to use different article types in a slot, you can define them a separate set of article templates per article type and slot used. If unknown article types are used or if specific article types are not set in an article type, the article templates from the default configuration take effect.

Default Contents

Here, you configure the demo articles for the slots used and demo contents for the input fields used (e.g. HTML-Area, Text-Area, etc.). Consequently, you can test how eMailings will later look that were created using this template.

Release

Here, you can set whether the Release Management should be activated or not for eMailings that were created by means of this template.

All users created in this client are available as users. In the case of global templates, all global users are thus offered. The same holds true for the role selection.

If the release process is activated, select here the users of specific roles that can release an eMailing prior to the transmission.

Important: At local level, only local users or roles can be selected, at global level only global users or roles can be selected.

You can find more information here: *Release process*

2.7.28 rollout

Here you can activate the rollout and define a default configuration. By a rollout an eMailing, which is based on this template, can easily be copied into several clients. Articles or articles and article pictures can also be copied.

Important: A rollout is only possible in templates and eMailings which are on the global level.

Update

Changes to the template do not directly affect eMailings that have already been created. You must first assign changes to an eMailing so that they can also become effective for this. At the same time, all source fields, the slot configuration as well as other selected settings are transferred to the eMailing. These protection measures prevent any accidental or incorrect changes to eMailings which possibly should be transmitted at this time.

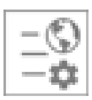
In the upper area you see eMailings that have already been created by means of this template and therefore are also the most likely targets of an adjustment. Below you can also display all other eMailings in the client but not based on this template. Select the corresponding eMailings with the checkbox before the name of the object and click on *Save and assign* at the end of the page.

Checks

The CSS check checks the CSS classes used and style instructions for their acceptance of distributed email clients. This is a traffic light system that gives a rough clue as to which of the functions used in the code cause problems in certain clients. The results must always be interpreted with regard to the planned implementation of the eMailings.

The HTML validation of the W3C helps you to generally discover problems with the created HTML code. Please note that the different quality of email clients used can make it necessary to use unconventional HTML technologies. For this reason, not every error shown really has to represent a problem or risk for the transmission of the actual eMailings.

2.7.29 LeadPage-Template

 LeadPage templates are created under *Templates*.

A LeadPage template is the basis of every LeadPage and can be filled with static content - such as the editorial or the imprint - in advance. In the LeadPage template, you define the display of a LeadPage. When a LeadPage is created, a copy of this template is created; the actual template remains unaffected by changes to the LeadPage.

Preview

Here you can view directly and check every task format that has been configured.

Configuration

- **LeadPage new version** defines the preview image of the LeadPage template, which is displayed when you select the templates when you create a new LeadPage.
- **Feed image** specifies the image to be used for display in a feed.

Design

The following display types are possible:

- *LeadPage*: The LeadPage defines the display of the LeadPage in a browser. The source is HTML.
- *Landingpage*: The display of the expanded contents of an article on the landing page. The source is HTML.
- *PDF*: The LeadPage is displayed as a PDF in a print-optimized form.

The internal HTML editor is available in the respective tabs for editing the output formats. If you check *Can be changed in LeadPage*, the output format can still be modified in the LeadPage.

Warning: Even if you allow editing in the LeadPage, you can only configure slots in the LeadPage template. However, you can use the source code in the LeadPage to influence the positioning of slots in the LeadPage.

Ersetzungsvariablen speziell für LeadPage und Webseiten

Es gibt die Möglichkeit ein Formular innerhalb einer Leadpage-Vorlage via Ersetzungsvariable zu platzieren. Hierbei gibt es zwei Varianten:

{$RENDER_FORM:ID}**: Bindet ein Formular in ein LeadPage-Objekt ein. Erkennt die LeadPage ein Profil (durch einen URL-Parameter oder durch ein Cookie), so wird das Formular das erkannte Profil aktualisieren. **{$RENDER_FORM_UNPERSONALIZED:ID}: Bindet ein Formular in ein LeadPage- oder Website-Objekt ein, ohne das Formular mit einem u-parameter zu versehen, dadurch sind auch Neueintragungen möglich, obwohl bereits ein Profil erkannt wurde. Diese Variante ist besonders für Standalone-Leadpages vorgesehen, welche auch von unbekannten Personen besucht werden soll.

Article templates

This view corresponds to the "extended editor" in the tab "Contents" of the LeadPage. Here you configure the name of the slots used and define their standard article templates. In a LeadPage, you can only fill slots that were previously configured in the LeadPage template used.

If it should be necessary to use different article types in a slot, you can define them a separate set of article templates per article type and slot used. If unknown article types are used or if specific article types are not set in an article type, the article templates from the default configuration take effect.

Default Contents

Here you configure demo articles for the used slots and demo contents for the used input fields (e.g. HMTL-Area, Text-Area, etc.). This allows you to test how LeadPages that are created using this template will look later.

Scoring

Here you can specify scoring values for the *LeadPage opening* and the *LeadPage click* that deviate from the standard definition. These values can be defined differently in the LeadPage.

Update

Changes to the template do not directly affect lead pages that have already been created. You must first assign changes to a LeadPage so that they also take effect for this LeadPage. All source fields, the slot configuration, and other selected settings are transferred to the LeadPage. This protective measure prevents accidental or incorrect changes to LeadPage that may already be online at this time.

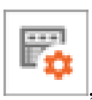
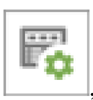
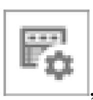
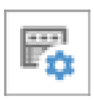
In the upper area, you see the LeadPages that have already been created using this template and are therefore the most likely targets of a comparison. Below this area, you can also display all other LeadPages in the client, but you cannot base them on this template. Select the corresponding LeadPages with the checkbox in front of the name of the object and click on *Save and assign* at the end of the page.

Check

Here you will find a selection of different options for checking the functionality of the lead page template:

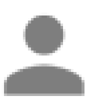
- **Individualization** shows the representation of individualized and personalized variables for the target group entered here. To set, select a suitable TargetGroup and save the object. If no TargetGroup is selected, the defined personas are offered for each pool. By clicking on "Preview" at a profile of the target group, the LeadPage will be adapted to the data of the profile concerned - e.g. individualized articles, etc. This allows you to check the correctness of individualizations and personalizations.
- **Eye tracking** provides an overview of the probable eye movements of the visitor. This check is subject to additional costs.
- **Text analysis** serves the textual evaluation of the content of the LeadPage for comprehensibility. This check is only available for German-language content and there are additional costs.
- **Links** displays all standard and individual links integrated in the LeadPage and checks their validity - i.e. whether the target can be reached. Correct links are marked with a green OK, possible errors are marked in red.

2.7.30 Article Template

, , ,  Article templates are stored in the *Templates* folder in the subfolder *Article templates*.

Article templates determine the display of an article having image and text within a slot. The editing of content can be performed easily in the text editor and images can be uploaded via the integrated content management system. Depending on the application, various display formats can be derived from an article template, e.g., HTML, text, PDF etc. In *Preview* you can test an article template with various contents and thus check the display for the recipient in advance.

2.7.31 Users

 A user corresponds to a user account for the system and is stored in the *User* folder below the *Settings* folder.

Users can be created at client level or global level. A user created at client level can only log into this exact client and therefore only has access to data of this client. Users created at global level, however, can access the data of all clients.

General

Information

Here, you enter general information about about the user. The main purpose of this data is to better address the user within the system.

The email address specified here is used to provide the user with information that he can request in the system - such as profile exports. You should therefore make sure that the email address stored here is valid and written correctly.

Settings

All settings made here apply to the user. He can also edit this data himself - either by navigation in the folder tree or by clicking on the user name at the top right. The selection of the theme always refers to the level of the user here - global themes can be assigned to global users only, global as well as local themes can be assigned to local users.

Roles

The origin of every selectable role is displayed as well. A distinction is made here between *System*, *Global* and *Client* roles. *System* roles are defined by the system (e.g. Accounts, Administrator), *Global* roles are at global level.

One or more roles can be assigned to a user. Through the assignment of such a role a user receives all rights that were defined in a role. A user - provided that he has the right - can assign the roles to other users that he himself has.

Login details

A user can log into the system using a username and password. The username can be generated by clicking on *Propose* - the email address entered is normally proposed here. Names that are defined too generally are not accepted by the system.

The password is entered either directly by entering a password twice or automatically by the system. A generated password is sent to the set e-mail address. After the first login the user has to assign a new password. This temporary password is valid for 24 hours.

Under *Restrictions* the user can be restricted to a demo phase. In the course of this, this user is denied access to the system automatically after 30 days. To immediately lock the account manually, checkmark *Account locked* here. In the event of too many unsuccessful attempts when logging in, the system will set the checkmark automatically here.

Sicherheitsrichtlinien

Bei der *Sicherheitsrichtlinie* werden alle verfügbaren Richtlinien angezeigt. Gibt es keine definierte Rcithlinie, so wird die systemeigene (default) verwendet. Wenn der Zugriff auf Personenbezogene Daten nur in einem bestimmten IP-Bereich möglich sein, so müssen mindestens zwei IP-Einschränkgung vorgenommen werden.

In der ersten muss der IP-Bereich festgelegt und die Rechte “Profile anzeigen” und “Login” gewählt werden. In der zweiten müssen alle übrigen IP-Bereiche (0.0.0.0/0) mit dem Recht “Login” definiert werden.

Login history

Here, you will find a list of the last 20 login (attempts) of the user with information about the browser and IP addresses from where he logged in. The user can also view this information about himself in the settings. If unsuccessful logins have occurred, the user will be informed about this once when logging in.

Restrictions

Global users can be limited to clients here. These users can then only change to the clients for whom they are also activated. The other clients are only displayed in the navigation but are no longer clickable.

> **Warning:** Please observe the current legal framework – particularly with regard to the data protection of your employees – when using this data beyond the purpose intended for system administration and which serve the purpose of maintaining data security.

2.7.32 Role

 Roles are created below the *Settings* folder in the *Roles* folder.

A role is a collection of rights that can be assigned to a user. A role can have any number of rights and any number of roles can be assigned to a user. Thus, you have every freedom for replicating the rights management in the system - regardless of whether you use few roles with the descriptions of the users, such as *Web designer*, *Editor*, etc. or whether you set up very finely adjusted roles that target more individual functions (such as *article management*).

> **Warning:** You can only assign privileges of a role that you yourself have. No other rights are displayed. This means that you can take a right but no longer regain it yourself. For this reason, it is imperative that you consider two things:
>
> - Always retain at least one user with the role of *System Administrator*. All rights are assigned automatically to this role.
> - Only assign the right for role management to trustworthy persons - particularly at the highest level. **This person must be trained accordingly with regard to the logic of users, roles and rights.**

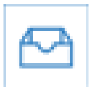
Overview of rights/privileges

The system separates rights into two groups. For the one there are the rights that are linked directly to the system‘s functions. For the other there are the rights that are linked directly to individual objects and media. For all objects there is a detailed list that appears identical in function per object. Every object has the privileges:

- *Log-in*: Every user requires this right to be able to log into the system. If one wants to revoke the login option for a particular group of users on short notice, it can be done quickly and easily with this privilege.
- *Create customer/mandator*: The user can create new customers/mandators.
- *Edit customer/mandator*: The user can edit existing customers/mandators.
- *Delete customer/mandator*: The user can delete an entire customer or mandator.
- *Create customer/mandator (demo)*: The user can create a demo customer or mandator.
- *Create folder*: Creation of a new folder in the navigation tree.
- *Manage folder*: Renaming and deleting a folder in the navigation tree.
- *View profile list*: View lists of profiles e.g. for pool or target groups.
- *Export profiles*: In the profile list, the user may export every profile in the current view.
- *Delete profiles*: In the profile list, the user may permanently delete individual profiles.
- *Import profiles*: The user may request to import into the system in the profile list.

- *Restore profiles*: The user may create and restore backups in the pool.
- *Display / edit blacklist*: View and edit the blacklist.
- *Send e-mailing*: The user can release an e-mailing for transmission.
- *Send e-mailing (global level)*: The user may send an e-mailing on the global level.
- *GeoDB services*: The user can have the profiles displayed on a map down to the street numbers.
- *Create e-mail display check*: Allow use of the premium display and spam check (additional charges may apply).
- *Show e-mail display check*: View existing premium display and spam check analyses.
- *Create EyeTracking check*: Allow use of the EyeTracking service for e-mailings and forms (additional charges may apply).
- *Display EyeTracking check*: View existing eye tracking analyses.
- *Create text analysis check*: Allow use of the text analysis for e-mailings and articles (additional charges may apply).
- *Display text analysis check*: View existing text analyses.
- *Manage users*: The user can create, edit and delete users.
- *Manage roles*: The user can create, edit and delete roles.
- *Manage themes*: The user can create, edit and delete themes.
- *Service center access*: Display of the *Service center* link in the main navigation and access to this part of the page.
- *View global accounting*: View the accounting on the global level.
- *Web services 1.0*: This right is assigned to users that are used for external system calls via the XML-RPC interface.
- *Display image personalization*: Allow use of image personalization (additional charges may apply).
- *Access to global data*: Access to the global area of a group of mandators.
- *Block and unblock objects*: Block and unblock objects within the system for other employees.
- *Block external access*: Prohibit the display of objects through access by outsiders (e.g. website, form, ...).
- *Export conversions*: The user may export conversion data of a transmitted e-mailing.
- *Safety guidelines*: The user may edit safety guidelines.
- *Editor configuration*: Defines whether a user may adjust the editors in the system
- *Display object*: The object can be displayed and linked externally; this right also permits the display by non-authorized users.
- *Manage object*: The The object can be deleted, copied and moved by the user.
- *Create object*: An object of this type can be created by the user from scratch.
- *Change* object: The object can be edited and changed by the user (editing and saving); without this right the user can only see a preview and detailed view of the object.

2.7.33 Container

 A container is created under *Templates* in the *Container* folder.

Containers are merely objects for containing data and can only be queried via TAL. These are used, for example, to carry out translations in a newsletter. With the aid of containers, it is thus possible to map all language versions via one single eMailing template in future.

Configuration

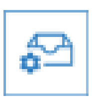
The fields that are shown in the container and that can be used in the e-mailing are defined via a *Container type*.

Read container contents via JSON

If the integration url of a container is called, the contents are output in JSON format. This allows container contents to be read out and, for example, further processed with JavaScript.

Important: The only exception is container lists. If a container list contains a container that is based on the same container type as the queried container, the container list will not appear in the JSON array

2.7.34 Container type

 A container type is created in the *Container types* folder.

The container type is an object containing container configurations. This configuration is freely definable and extendable.

The container type behaves like the *Article type* the only difference being that there are no roles within the container type.

The following field types can be assigned to a container:

- One-line input - Provides a simple, one-line input
- Multi line input - Generates an input field for multi line text
- Multi line input (formatted) - like the multi line input but integrated in a WYSIWYG-Editor
- Multi line input (HTML-Editor - like the multi line input (formatted) but with a complete WYSIWYG-Editor
- Single selection - Generates a dropdown list
- Multiple selection - Generates a checkbox list
- Image - Generates an image picker
- Link list - Generates a list with links
- Date with time - Generates an input option of data and time with calendar
- Colour - Generates an input field for a color with color picker

Furthermore, there are also special field types intended for the design and documentation of the container type.

- Documentation - Displays the text that was entered in the default field
- Separator - generates a line for separating the fields

Deleting container types

Container types can first be deleted if there are no more containers that were created by means of this type.

2.7.35 SmartLink

SmartLinks are created under *Tracking*.

A SmartLink can contain several links. For each configured link, it provides an integration URL that can be used in various contexts. For each link, several redirecting URLs can be defined that redirect a call of the integration URL. Redirecting can be based on membership to target groups. A SmartLink can be used as a switch in this manner (see the areas of use).

At the same time, a SmartLink can manipulated a profile depending on the profile properties wants an integration URL is called. That is, specific flags, selections or dates can be set or removed.

If profile data is to be changed during a visit to a target group link, this can be set for each target in the *Profile* tab. The profile of each identified visitor is thus updated with the configured data.

The option “`Combine old and new values`” is only relevant if it is a multiple selection. If this option is activated, the configured options are added to the possibly existing options. Otherwise the existing options will be overwritten with the configured options.

Important: The legal requirement for using this function should be examined in detail first.

The individual functions can be activated/deactivated per target as required. This allows, for example, the profile update to be deactivated temporarily without deleting the complete target configuration.

Areas of use

Since the SmartLink functions like a URL switch, any type of URL can be used.

In particular, the following URLs can be used:

- **IFrames:** The integration URL of a SmartLink can be used as an IFrame source URL of an IFrame. Depending on the target group, different IFrame content is displayed for different profiles.
- **Pictures:** These can replace an integration URL as well as an image source URL. The SmartLink then offers different pictures depending on the target group.
- **PDFs:**in the case of PDFs or other files, the integration URL can redirect to different download URLs. Different files can be output depending on the target group affiliation.
- **Adds:** The SmartLink can also be used to output individual advertising banners. This involves the same principle as when pictures are used.

2.7.36 Dynamic image

 A dynamic image object is created in the *Images* folder.

A dynamic image is an image in SVG format that can be enriched by means of TAL and personalization variables. At the same time, the SVG is converted automatically into a conventional image format for output.

Preview

Direct preview of the SVG as a raster graphic.

Dynamic image

Beside the input of the XML source code, it is possible to integrate predefined personalization variables into the source code by means of a dropdown menu. The TAL interface is documented in the developer area.

Additional images can also be integrated into the SVG: e.g.: *<image x=”0” y=”0” width=”64” height=”64” xlink:href=”data:image/png;base64,BASE64CODE”/>* and *<image x=”0” y=”0” height=”64” width=”64” xlink:href=”{$IMAGE:OBJECT_ID}”/>*

Container:

Select a container to use its contents via TAL in the source code of the Dymanic image.

Integration

A tool for generating an individual link to the image is already available here for integration. Thus, the width, height, format etc. can also be changed dynamically, similar to a normal image.

2.7.37 WebTouchPoint

Websites outside of the system can be used in a scoring assessment by means of WebTouchPoints.

To do this, a tracking code (`Integrations-Tab`) must be integrated in the desired websites. The websites with a tracking code are either automatically displayed after the first call in the touchpoint, or you deactivate the *automatische Webseiten-Anlage* in `Konfigurations-Tab`. In this case, you need to indicate the relevant websites manually in `Konfigurations-Tab`.

Scoring with WebTouchPoint

An individual score can be issued for each added website. In addition, you can also indicate which score is to be issued starting at which opening time. This is useful in order to distinguish whether a visitor has actually read specific content, or whether the website was briefly opened and then closed.

Statistics

The statistics tab provides an up-to-date overview of the activities of the last month.

2.7.38 Campaign Designer

Preamble

The Campaign Designer enables the configuration of communication lines that integrate existing system components and exten

- Automation of 1:1 marketing campaigns by e-mail (primary), letter, etc. based on behavioral, decisional and conditional components.
- Mapping of complex marketing campaigns without deep technical knowledge (programming) via drag and drop, independent of the workstation.
- Sales-supporting as an emotional factor due to currently very strong presence of the topic in the media.
- Natless integration into the basic system regarding familiar object configuration, optical appearance and uniform user authentication.

system interface

The size of the workspace is determined by the window size when a new campaign is created and is saved as part of the campaign.

With the button Workspace the size of the workspace can be adjusted manually. The Workspace button as well as a thumbnail of the campaign for navigation and a possibility to zoom in and zoom out can be found at the bottom left of the workspace.

User permissions

To use the full functionality of the Campaign Designer the account needs the following price model rights

Send eMail -> Send eMailing Send PrintMailing -> Send PrintMailing Change Profile -> Show Profiles Activity Score -> Scoring / Activation Notification -> Send eMailing

Basic elements

Campaign elements serve to visually model campaigns and are grouped into three basic types:

- Start and end points: Serve as active and passive import elements for profile data and for the formal conclusion of a campaign.
- Actions: Perform campaign actions such as sending an eMailing, changing a profile, or delaying a campaign.
- Conditions: Serves as decision elements to navigate to different process lines, e.g. click in eMailing, website visit, or form entry.

Campaign start points

Starting points represent entry or import points for profiles within a campaign. There must be one starting point per campaign. Multiple use of starting points is not possible. Each element start point can serve as a target point for manual profile entry, for example, for object actions, elements for copying or moving profiles in campaigns or Web services. All profiles that enter a campaign via a starting point are checked for the multiple run configured in the element. If multiple passes are deactivated, a profile may not re-enter the campaign if it already existed in the past in the campaign.

Element basic functions

The following functionality is available to all campaign elements:

- Placing an element
- Removing an element
- Change Settings
- Placing and removing connection lines
- Move profiles

Element Connections

Elements can be connected with each other to define the path of a profile in chronological order. There are two types of connecti

- Input: Each element (except start objects) has an input. At this input 1-n elements can dock.
- Output: Each element (except end objects) has 1-n outputs, depending on the element type. Only one element can dock per output.

Element Delete or Edit

If elements are removed from an active campaign, this is only possible if there are no profiles on the corresponding element. The

- Move profiles to another element.

Profiles in campaigns

The following restrictions apply to profiles in a campaign:

- A profile may only occur 1 time per campaign
- If a profile moves in the campaign, it cannot be imported/acquired again.

campaign states

A campaign can have the following two states

- Processing (paused): Campaign does not execute any actions. The status is not checked for validity when it is saved. When you create a new campaign, this is the first state.
- Active (live): Campaign settings and elements cannot be changed. Campaign actions are executed without restriction. The status is checked for validity before activation.

Campaign starts

Beim Wechsel des Kampagnenzustands in „Aktiv" öffnet sich ein Modal welches abfragt ob die Kampagne gestartet werden soll und darauf hinweist, dass vor dem Start der aktuelle Stand gespeichert und die Kampagne validiert wird. Es erfolgt eine Status und Fehlerrückmeldung bzgl. Validierung und Zustand (erfolgreiche bzw. fehlerhafte Speicherung und Start). Die Kampagne wird umgehend gestartet.

Campaign active

If the campaign is active, the tree changes to the minimized view and all elements are deactivated (same style as the deactivated

- The number of profiles on the node is displayed below the elements.
- No more elements can be added, moved or changed.
- The save button remains visible, but is deactivated.

Reimport campaign

Die Funktion steht nur bei Verwendung des Startpunktes „Kampagne" zur Verfügung im Zustand „Aktiv". Beim erneuten importieren der Kampagne werden die Profile der ausgewählten Zielgruppe erneut in die Kampagne geladen. Die Akzeptanzkriterien sind dieselben, wie bei einem regulären Start der Kampagne. Die Optionen Mehrfachdurchlauf und Sperrfrist werden berücksichtigt. Profile, die sich aktuell in der Kampagne befinden, werden nicht beeinflusst.

Element Validation

Elemente: Das Speichern von Element-Konfigurationen wechselt den Elementzustand in „konfiguriert", sofern die Konfiguration syntaktisch valide ist und die Pflichtangaben vorhanden sind.

Campaign Validation

Validierung bei Aktivierung (große Validierung) Beim Wechsel des Kampagnenzustands in „Aktiv" wird die Kampagne auf Validität geprüft. Folgende Kriterien sind dabei mindestens maßgebend:

- At least one start element exists.
- All element connectors have connections
- Alle Elementzustände sind „konfiguriert"
- All campaign strands end with an endpoint

Validation during storage (small validation)

When a campaign is saved, a minimal validation takes place if there is no change in the campaign status. Errors can occur in both validations and must be reported to the user.

Deletion dependencies

All objects used in the Campaign Designer have a hard deletion dependency. This means that the object used cannot be deleted as long as it occurs in an element configuration. This also applies to scoring groups and pool fields.

Elements Overview

The following elements are available for configuring the campaign:

start points

- *Campaign* - Unique campaign with professional import from target group

- *Dynamic* - Continuous campaign with profile import from field definition
- *Persona* - Continuous campaign with professional import from persona assignment
- *Neutral* - Ongoing campaign without active professional import
- *Form* - Continuous campaign Profilimport after form submission

Endpoints

- *End* - End of a campaign line

Action

- `Send eMailing` - Send an e-mail
- *Send letter* - Send print letter via partner
- `Waiting` - Time delay of profile transport
- *Change profile* - Targeted change of existing profile properties (fields)
- *Activity Score* - Sets a value for Activity Scoring
- *Notification* - Sending a notification to deposited recipients
- `Copy in campaign` - Copies a profile into another campaign

Condition

- *eMailing open* - Checks if the profile has the selected email open
- *Article open* - Checks if the profile has opened an article of the selected email
- *eMailing responds* - Checks if profile opened or clicked the selected email
- *eMailing receive* - Checks if profile has received the selected email
- *Check profile* - Checks profile for certain field values
- *Filed form* - Checks if the profile submitted a form
- `Check persona` - Checks if the profile is assigned to a persona
- *Website visits* - Checks if the profile has visited the selected website
- `Leadpage visits` - Checks if the profile has visited a LeadPage
- *Link clicked* - Checks if the profile has clicked a SmartLink
- *check scoring* - checks if the profile has reached a certain scoring value
- `In the vicinity` - Checks whether the profile is located in the vicinity of certain coordinates
- `A/B Split` - Split profiles into random groups
- *External condition* - Calls an external URL (optionally with profile variables) and then decides based on the response of the call
- `Max. Runs` - checks how often a profile has run through this campaign

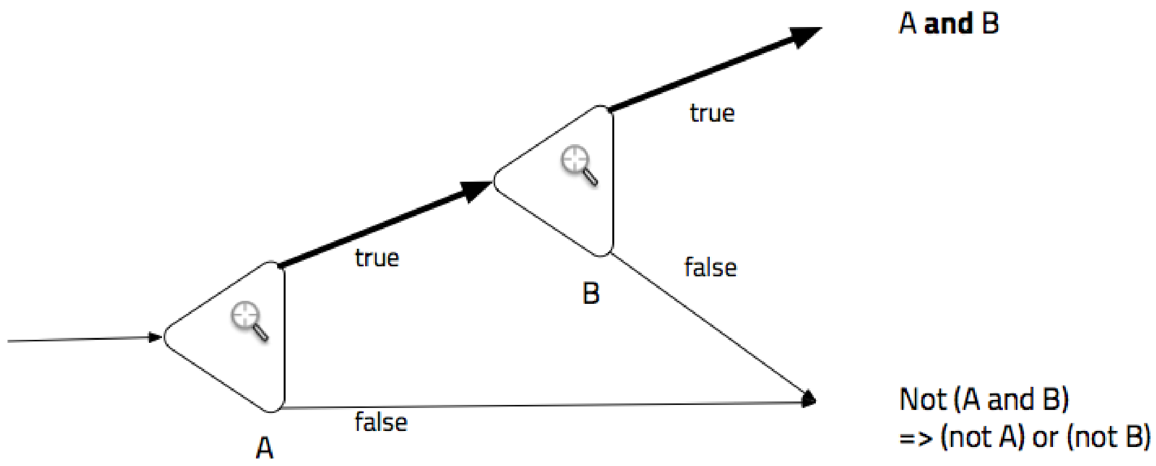
You can drag and drop these elements from the element tree on the left edge of the window onto the workspace. By clicking on the gear of an element or by double-clicking on an element, you can open the configuration window of the element and display or change the configuration.

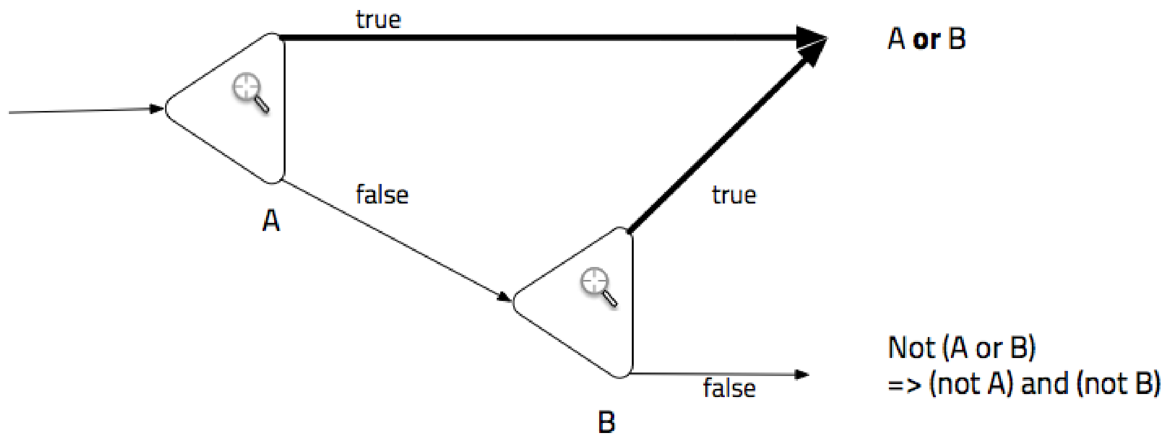
Connections

- Each element has so-called inputs and outputs with which you can connect the elements to each other. The connections indicate to which element a profile, which runs through the campaign, will be transported next.
- Each input of an element can receive one or more connections. It is possible to transport from different elements of profiles to one element and thus to merge different rods.
- Each output can have only one connection so for each element there is always only exactly one following element.
- **Start points** have no inputs and exactly one output. As soon as profiles enter a starting point, they are transported to the first element.
- **Actions** have exactly one input and one output. Profiles remain on the element until the action is successfully completed and are then transported to the next element.
- **Conditions** have exactly one input and exactly two outputs. Profiles are transported to the next element either via the positive output (green) or the negative output (red) as soon as the condition is fulfilled or after an optional waiting time has elapsed.

Linking conditions

- Conditions can be linked by placing them one after the other. In analogy to target group configurations one can speak of AND and OR connections.





Campaign Designer elements

Starting points

Starting point: Campaign The "*Campaign*" starting point imports profiles from a specified target group into a campaign once. This is suitable, for example, for classic campaigns such as competitions, voucher campaigns, events, etc..

Configuration

Targetgroup Under Targetgroup you can select the desired target group. All profiles of this target group are imported once at the start of the campaign.

Multi-pass If the option "*Multi-pass*" is activated, profiles can be re-imported. If this option is disabled, profiles can only run through this campaign once and will be ignored when re-imported.

Blocking period If the "*Multi-pass*" option is enabled, you can optionally specify a expiration date. A profile can be re-imported into this campaign after the end of this campaign at the earliest after the expiration of the blocking period.

Additional Information

reactivation If the "*Campaign*" starting point is used and the campaign has already been activated, the "*Reactivate*" function is also available. This function is displayed next to the button for activating a campaign. When the campaign is re-imported, the profiles of the selected target group are reloaded into the campaign. The acceptance criteria are the same as for a regular start of the campaign. The multi-pass and blocking period options are taken into account.

Starting point: Dynamic The starting point "*Dynamic*" continuously imports those profiles which have the configured field properties. A multi-pass is not allowed at this starting point. Each profile can therefore only run through this campaign once.

Selection Under *Selection*, you must first select a pool, then the single or multiple selection field to be checked, and finally the value that the selected single or multiple selection should have. All profiles that have this value will be imported when the campaign starts. If the campaign is activated, further profiles are imported as soon as they receive the selected option.

Starting point: Form The starting point "Form" automatically imports the profiles that have submitted the form. A multiple run is not allowed at this starting point. Each profile can therefore only run through this campaign once.

Formular Under "Form" you first have to select a form. In addition, you can specify whether only newly entered profiles should be imported into the campaign or also profiles that have been updated via the form.

Starting point: Persona The starting point "*Persona*" automatically imports those profiles which have been assigned to a certain persona. A multiple run is not allowed at this starting point. Each profile can therefore only run through this campaign once.

Persona Under "Persona" you only have to select one persona. All profiles that can be assigned to this persona are automatically imported into the campaign.

Starting point: Neutral Neutrale Startpunkte können sich nicht aktiv um Profile bemühen und diese importieren. Stattdessen wartet ein neutraler Startpunkt darauf, dass andere Objekte oder Kampagnen dieser Kampagne Profile zuweisen.

Mögliche Elemente, welche die Profile der Kampagne zu werfen können: - SOAP-API - Kampagnen-Element: Kopiere in Kampagne

End-point Each input and each output of an element must be connected to another element. At the end of the campaign at least one endpoint must be used to connect the last action. Otherwise, endpoints have no configuration options.

Actions

Action: send eMailing The action "*Send eMailing*" sends an eMailing (trigger) to the profile which enters the element.

Configuration

eMailing Under "*eMailing*" select the eMailing (Trigger) to be sent to the profile. Make sure that the eMailing (Trigger) is configured correctly. If the eMailing (Trigger) cannot be sent, e.g. because of missing sender address, the action is ignored and the profile is immediately transported to the next element.

Recipient filters If the option "Recipient filters" is activated, only those profiles will be sent which have a valid permission, are not unsubscribed and are not on the blacklist. If this option is disabled, profiles will be sent regardless of permission and possible unsubscribe. Please note that disabling this setting may have legal consequences.

Action: Delay The action "*Delay*" delays the transport of a profile for a certain time.

Three variants are available for this purpose:

- Time period
- Date & time
- Weekday & time

Important: Example of the effects of pausing campaigns: A profile that moves at 12:00 p.m. on a waiting item with a configured waiting time of 4h moves at 16:00 p.m. on. This also does not change if the campaign is paused and restarted in the meantime. If the campaign is paused until after 16:00 and then restarted, the profile starts immediately at the time of reactivation.

Configuration

Time period If you have selected the "*Time period*" variant, you can specify the number of days and hours that a profile should wait after reaching the element until it is transported to the next element.

Date & time If you have selected the "*Date & time*" option, you can enter a date and time. The profile then waits until the specified time of the specified date before being transported to the next element.

Weekday & time If you have selected the "*Weekday & time*" option, you can select a time and any days of the week. The profile then waits until the specified time of the next selected weekday before being transported to the next element.

Action: Send PrintMailing The action "*Send PrintMailing*" sends a PrintMailing to the profile that enters the element.

Attention Please note that there are costs when sending a PrintMailing.

Configuration

PrintMailing Under "*PrintMailing*", select the PrintMailing that is to be sent to the profile.

Recipient filters If the "*Shipping protection*" option is activated, only profiles that are not on the blacklist will be sent. If this option is disabled, profiles will be sent regardless of the blacklist. Permission and advertising consent are not checked for PrintMailings.

Action: Update profile The "*Update Profile*" action changes the value of any field in a profile. This can be used, for example, to set interests or save the current date.

Configuration

Update Field First select a pool. (If your campaign is used for profiles from different pools, you must use a separate "*Update Profile*" element for each pool.) Select the profile field you want to change. Finally, specify the value you want to save in the field. Depending on the field type, corresponding input fields are available. For the field type "*Multiple selection*" there is an additional switch: "*merge*" available in addition to the option selection. If this button is activated, selected options are added to the current options. If this switch is deactivated, all currently available options will be removed before the selected options are saved.

Action: Activity Score The action "Activity Score" offers the possibility to give the current profile an activity scoring value in a scoring group.

Configuration

Scoring group Select the scoring group for which you want to set a scoring value.

Scoring-Points Enter the number of scoring points you would like to give the profile.

Description Enter a description of the scoring points here. You will see this description in the scoring history of the profile.

Action: Notifications The action "Notifications" sends a selected eMailing (Trigger) to one or more email addresses, whereby the eMailing (Trigger) is individualized with the data of the profile which triggers the notification.

Configuration

eMailing Select an eMailing (Trigger), which should be sent as notification mail.

Recipients Specify one or more email addresses to receive the notification. Multiple email addresses can be specified comma-separated.

Action: Copy in campaign The action "Copy in campaign" copies a profile into another campaign. The multiple run rules of the target campaign apply.

Configuration

Campaign Select the targetcampaign to which you want to copy a profile.

Conditions

Condition: eMailing opened The condition "*eMailing opened*" checks the recipient behavior with regard to openings of a sent eMailing during a definable check period for mapping behavior-based communication paths.

Configuration

eMailing to check Select the eMailing (Trigger) to be checked for openings.

Test period before reaching the node Choose a possibility to define the test period:

- **Openings** that have taken place since the profile entered the campaign are taken into account.
- **In the last** openings are taken into account which x days and y hours before reaching this element took place.

- **Date & time** Openings that have taken place since the specified date and time (UTC) are taken into account.

Waiting time after reaching the node

- **Waiting time** If no opening has taken place when this item is reached, a maximum of x days and y hours is waited before a decision is made for the condition. If the receiver creates an opening, the waiting time ends immediately.

Condition: Article opened The condition "*Article opened*" checks the recipient behavior with regard to openings of a sent e-mail during a definable test period for mapping behavior-based communication paths. This checks whether internal or external landing pages of articles have been opened within the selected eMailing (Trigger).

Configuration

eMailing to check Select the eMailing (Trigger) which should be checked for article openings.

Test period before reaching the node Choose a possibility to define the test period:

- **Since inception** Article openings that have taken place since the profile entered the campaign will be considered.
- **In the last** article openings are considered, which x days and y hours before reaching this element have taken place.
- **Date & time** Article openings that have taken place since the specified date and time (UTC) are taken into account.

Waiting time after reaching the node

- **Waiting time** If no item has been opened when this item is reached, a maximum of x days and y hours will be waited before making a decision for the condition. If the recipient creates an article opening, the waiting period ends immediately.

Condition: eMailing reaction The condition "*eMailing reaction*" checks the recipient behaviour with regard to eMailing or article openings of a sent eMailing during a definable check period for the illustration of behaviour-based communication routes. This checks whether internal or external landing pages of articles have been opened within the selected eMailing (trigger).

Configuration

eMailing to check Select the eMailing (Trigger) which should be checked for article openings.

Test period before reaching the node Choose a possibility to define the test period:

- **Since inception** eMailing or article openings that have taken place since the profile entered the campaign will be considered.
- **Date & time** E-mailing or article openings that have taken place since the specified date and time (UTC) are taken into account.
- **Date & time** E-mailing or article openings that have taken place since the specified date and time (UTC) are taken into account.

Waiting time after reaching the node

- **If no e-mailing or article opening has yet taken place when this item is reached, a maximum of x days and y hours will still be waited before a decision for the condition is made. If the recipient generates an e-mailing or article opening, the waiting period ends immediately.

Condition: Receive e-mail The condition "e-mail received" checks the delivery of a sent e-mail during a definable check period for mapping communication routes.

Configuration

eMailing Select the eMailing (Trigger) whose delivery is to be checked.

Test period before reaching the node Choose a possibility to define the test period:

- **Since inception** eMailing deliveries that have taken place since the profile entered the campaign will be taken into account.
- **In the last** e-mailing deliveries are taken into account which x days and y hours before reaching this element have taken place.
- **Date & time** eMailing-deliveries are considered which have taken place since the given date and time (UTC).

Waiting time after reaching the node

- **Waiting time** If no e-mailing delivery has taken place when this item is reached, a maximum of x days and y hours is waited before a decision is made for the condition. If the email is delivered to the recipient, the waiting time ends immediately.

Condition: Web form submitted The condition "*Web form submitted*" checks whether a web form has been sent successfully. This serves e.g. to measure the success of registration forms in the context of content downloads such as whitepapers, etc.

Configuration

Web form Select the web form that you want to check for activity.

Test period before reaching the node Choose a possibility to define the test period:

- **Since inception** Activities that have taken place since the profile entered the campaign are taken into account.
- **In the last** Activities that took place x days and y hours before reaching this element are taken into account.
- **Date & time** Activities that have taken place since the specified date and time (UTC) are taken into account.

Waiting time after reaching the node

- **Waiting time** If no activity has taken place when this element is reached, a maximum of x days and y hours is waited before a decision is made for the condition. If the receiver creates an activity, the waiting time ends immediately.

Condition: Check profile The condition "*Check profile*" checks profile fields for specified conditions and field values.

Configuration

Conditions Here you can select a pool, a field and the comparison operator to be used and specify the value to be checked.

Condition: Persona Check The condition "*check persona*" checks a profile for belonging to a selected persona.

Configuration

Conditions Here you can select the persona to be checked for.

Condition: Website visited The condition "*Web page visited*" checks whether a profile has been identified by a WebTouchPoint during a definable period of time, which corresponds to the visit of the page. The prerequisite is the prior integration of the tracking code for WebTouchPoints in the corresponding website.

Configuration

Website Here you can select a WebTouchPoint and optionally a web page. If no Web page is selected, all Web pages of the WebTouchPoint are checked.

Test period before reaching the node Choose a possibility to define the test period:

- **Since inception** Website visits that have taken place since the profile entered the campaign are taken into account.
- **In the last** Website visits are taken into account that took place x days and y hours before reaching this element.
- **Date & time** Website visits that have taken place since the specified date and time (UTC) are taken into account.

Waiting time after reaching the node

- **Waiting time** If no website visit has taken place when this item is reached, a maximum of x days and y hours will be waited before deciding on the condition. If the recipient creates a website visit, the waiting time ends immediately.

Condition: Leadpage Visited The condition "*Leadpage visited*" checks whether a profile has visited a lead page during a definable period of time.

Configuration

Leadpage Here you can select a LeadPage for which you want to check the visit.

Test period before reaching the node Choose a possibility to define the test period:

- **LeadPage visits that have taken place since the profile entered the campaign are taken into account.
- **In the last** LeadPage visits that took place x days and y hours before reaching this element are taken into account.
- **Date & time** LeadPage visits that have taken place since the specified date and time (UTC) are taken into account.

Waiting time after reaching the node

- **Waiting time** If a LeadPage visit has not yet taken place when this element is reached, a maximum of x days and y hours is waited before a decision is made for the condition. If the recipient creates a LeadPage visit, the waiting time ends immediately.

Condition: A/B Split The condition "A/B Split" divides the profiles in the specified ratio.

Configuration

ratio Here you can define the ratio of the distribution.

Condition: In the vicinity The condition "Within a radius" checks whether a profile is located within a configurable radius around configurable coordinates.

Configuration

Coordinates Here you can enter longitude and latitude.

Circle * Here you can define the radius in which the coordinates are to be searched.

Condition: External condition The *External Condition* condition calls a URL and transmits the profile ID and a unique campaign ID. The condition is true (green path) if the external condition gets back a JSON string with the content "true". The condition is false (red path) if the external condition gets back a JSON string with the content "false". If a response is returned without JSON (true or false), the external URL for this profile will be called up to 10x again before the profile stops permanently.

Format of the request:

```
{
    "Profile": [{
        "ProfileId": "xxx",
        "ApplicationTypeId": "1"
        }]
}
```

Format of the valid JSON String Responses:

```
{
    "Result": true|false
}
```

Configuration

URL Here you can enter the URL to be called.

Condition: Max. Runs The condition ""Max. Runs" checks how often a profile has run through a campaign. The condition is true (green path) if the number has been reached or exceeded.

Configuration

Number of runs

Here you can specify the number of runs. The following comparison operators are available:

- Equal
- Greater than or equal to
- Bigger
- Less than or equal
- kid

CHAPTER 3

Working with the system

3.1 Presentation of information and transmission

3.1.1 Creating a new eMailing

Objective: Create a new eMailing - e.g. newsletter, event-mailing, info-mailing, etc. - with subsequent transmission.

Procedure:

1. To do this, open the page **Create eMailing from template**: on the **start page** click on the + icon of the *eMailing* button or open the *eMailings* folder and click there on the *New* button above the overview of all mailings in list form.

2. Give the eMailing a descriptive name in the *Name* field and define a text for it in the *Subject line* field. Then select a suitable template and click on *Save*. With this the system generates a draft for your eMailing and opens this in the *Preview* HTML view. Click on the *Contents* tab to change to the edit mode.

Important: Templates already contain every basic element for your eMailing and thus relieve you of routine tasks: unsubscription link for recipients to unsubscribe to future newsletters, customization variable for personal salutation, buttons for cross-media newsletters to select different types of display (HTML, text, PDF, readout per text-to-speech, Podcast and RSS-Feed), as well as every article short-version link to the corresponding long version.

---

3. Configure and edit the eMailing‘s contents:

- Replace the banner by clicking on it with the mouse. Select from the alternatives shown under it a suitable one for your eMailing. (This function is only applicable for the **Example eMailing (Cross Media)**)
- Insert the special greeting text of your email in the field below. The salutation customized for the recipients is already fully integrated in the template.
- You delete the placeholder article and teaser from the template by clicking on - above the slots “Articles” and “Teasers”.
- You then insert the articles or teasers for the mailing created from the list by clicking on +- here, you can also define another article template and specify whether the article should only be displayed in a specific target group.

- You can change the order of the articles by clicking on the UP and DOWN arrow.
- You can change the contents of the selected articles by clicking on the *Edit* button. A window will open with the article. Here, you can edit or replace the texts and image. The system automatically scales the images to the size of the placeholders in the template. Articles consist of a short and a long version. The contents for both are to be edited in separate fields.
- Various selection options for text layout are shown above the editing fields: for the display of attributes, colors and formattings. Moreover, you can import contents about it from other text and Word documents and declare portions of text as links to other pages, e.g., to landing pages. It is also possible to insert HTML source code.
- You save the changes by clicking on *Save* on the top right or bottom right. You will be taken back to the content editing of the eMailing. After you have added and possibly edited all articles and filled the predefined text fields, click on `Save editor` on the lower right and your changes will then be saved.
- In the *Configuration* tab, define the sender data and response data with names and email addresses, change the predefined customized salutation and define other references to external tracking systems as required. Save your entries by clicking on *Save*.
- It is possible to also attach files to each eMailing. To do this, go into the *File attachments* tab and select the corresponding file(s). It is generally recommended not to attach files, however, since these will most likely end up in the SPAM folder of the recipient or be will be deleted immediately.

4. Implementing quality check and test transmission: To do this, click on the *Check* button above the edit window. You will see your eMailing - with tabs for implementing the following tests:

- **Display & Spam** provides a preview for checking the contents and layout. Additional charges may apply here, however.
- **Display & Spam (Quick)** Firstly, a free check without any additional design tools is provided for you here for checking the display. Secondly, by means of independent, external software, you can calculate the probability that your eMailing would be declared as SPAM by the mail clients.
- **Eye-Tracking** allows you to view the eMailings for tracking and viewing the user. With this check, additional charges may apply.
- **Text analysis** allows you to evaluate the text of the eMailing for readability. This check is only available for German eMailings and additional charges may apply.
- **Links** displays a list of all standard and individual links integrated in the mailing and checks these. Correct links are highlighted with a green OK and possible errors are highlighted red. Check correct as well as broken links since correct links might still refer to wrong pages or faulty links are only marked as such because they only work properly during transmission.
- **Customization** indicates the display of customized and personalized variables for the profiles of the target groups set here. To set a target group, search for a target group and save the object. The list of the profiles from the target group (max. 25) allows you to test profiles and view the eMailing from the perspective as received by the corresponding profile. To do this, click on the *Preview* button to the right of the corresponding profile.
- **Test transmission** allows you to test the display quality, customization/personalization and deliverability for specific mail clients under real conditions before the actual transmission by means of a target group defined by you. You can also enter a list of emails directly in the input field as recipients. These are then only customized for the sample profile, however, and therefore only correspond to the standard preview. Mails sent directly to individual persons are marked in the subject line with “Test”.

5. Start transmission for a target group: To do this, click on the *Transmit* button above the edit window. A page with test criteria and windows will open where you can define the target group and enter the date and time (UTC) for the start of the transmission. When entering the time for the start of transmission, please make sure that you allow for a time lag of 1 - 2 hours compared with UTC and the local time in Central Europe. During winter the time lag for local time is -1 hour and during summer -2 hours.

6. Last check of all data and media of the eMailing before the transmission takes place by clicking on *Transmission preview*. A page opens with the following tabs:

- **Preview** displays the different display forms of the newsletter as well as addresses and target groups via the selection window
- **Customization/** displays which articles and teasers are configured for which target group in the newsletter
- **Profiles** displays the profiles of the pool for this eMailing in a list with the corresponding status - whether they are OK or not OK for the transmission
- **Transmission start** displays the configuration data of target group, subject line(s), sender and statistic for this mailing in an overview.

7. Start the transmission: After clicking on the *Start transmission* button on the lower right, a message opens with the number of profiles to whom the eMailing will be transmitted. Click there once again on the *Start transmission* button. The eMailing is then finally transmitted.

Tip: Create a target group of your own for your tests. Enter yourself in the pool and create a target group with your email address as criterion. Customizations and personalizations can be reliably tested using this method exclusively. In so doing, you can also include clients in another test group.

3.1.2 Customized articles and news in the eMailing

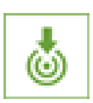
Objective: Two-step customization of the contents of an eMailing - without programming skills. During eMailing customization you can define which target group an article is selectively intended for. You can create special variants of such an article by profile during article customization, provided that the article is shown. This 2-step strategy provides top quality content customization.

Example:

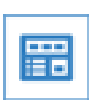
You want to display the article “Sport in the mountains” only to the target group “Sportsmen” (Interest = “Sport”). This article should not appear in the target group “Culture” (Interest = “Art”). If the article is displayed in the sportsmen target group, the relevant variants for individual sub-target groups should then be displayed automatically in terms of content and visual appearance, e.g. for “swimmers”, “climbers”, “bikers”, “hikers”.

Procedure: eMailing customization (1st step)

1. Ein eMailing aus den eMailing-Vorlagen auswählen: *Creating a new eMailing*



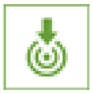
2. Create the necessary target groups for which the individual articles are to be displayed (culture, sport)



3. In the *Articles* folder, create all [articles] required for the eMailing and insert contents (culture, sport)
4. In the eMailing for the articles that should only be displayed for special target groups, define the corresponding target group. (Culture –> Culture, Sport –> Sport)

Procedure: Article customization (2nd step)

1. In the *Articles* folder, create a neutral main article (Sport), then create the target group specific alternative articles for the same topic. (swimmers, climbers, etc.)



2. Create the necessary *target groups* for which the relevant alternative articles are to be displayed. (swimmers, climbers)

3. In the *Customization* area of the main article, assign all alternative articles - with a definition of the target group for each alternative article.

4. In the *eMailing*, only insert the main article.
5. If the eMailing's recipient is not in any target group, then the main article is displayed.

Tip: Copy the main article (f necessary repeatedly) and adjust the contents to the individual target groups. In doing so, it usually suffices to adjust the image and the headline.

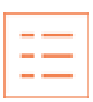
3.1.3 Profile updating in the eMailing

Objective: Recipients of an eMailing are to update their interest profile themselves via an integrated web form

Actions relating to how you create a web form for updating the profile and how you integrate it into the eMailing are described in detail in the chapter "Address generation via web forms".

Procedure: Direct editing

1. Create a form, see: *Creating a web form*

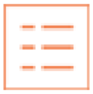


2. Only integrate fields in the web form, which the user is to update by himself.
3. Integrate the web form in the eMailing with contents prepopulated. See **Integration** tab in the **Web form** object.

Caution: Emailings for updating the profiles of interested parties are to be conspicuously provided with the note stating that the recipient may not forward the email in question. Otherwise the new recipient can view and change these data. Therefore, never let critical data be updated using this procedure.

Procedure: Secure editing

1. Create a form, see: *Creating a web form*



2. Only integrate fields in the web form, which the user is to update by himself.
3. Create a second web form that upon acknowledgement (send) sends a special email (confirmation email), in which is present the link to the first web form for editing the profile. This second form cannot contain any fields, only a note stating that an email is being sent for editing the profile in the attachment.
4. Integrate the first web form in the eMailing with contents prepopulated. See: "Inserting in the eMailing with prepopulated fields" under *Inserting web forms in the website or in the eMailing*

3.1.4 Social sharing for articles

Social Sharing means the sharing of information (news and articles) in social networks. You can integrate special links in your eMailings that the recipient can use to publish (share) the appropriate article in his preferred social network.

You just need to integrate the required replacement variables into the article templates used. Example: Sharing in Twitter

The list of all replacement variables can be found in the Service Center in the area "Personalization variables". It is not possible to use the function without article templates (source mailing) at present.

How does Social Sharing work?

The replacement variables are replaced by pre-defined links. These transfer the link to the social network on the landing page of the corresponding article. If a user from the social network now clicks on the transferred link, he will either be taken to the long version of the article and see the article just like the newsletter recipient or he will be routed to the destination of a preset landing page link.

Which social networks are supported?

The system currently supports sharing in the following networks:

- Facebook
- XING
- Twitter
- LinkedIn
- MySpace
- Delicious
- Google Bookmarks

Which data is transferred to the social networks?

This depends on the network. The headline of the article as well as the link to the landing page is normally transferred. Facebook is the only exception: Here, a text passage as well as the article image are transferred and displayed on the profile page.

Are statistics collected?

Each "Share" from the newsletter to the social network as well as each click on the links in the networks is evaluated. The evaluation from these actions can be viewed in the transmission statistics.

What must be borne in mind when using social networks?

To display the correct article and to ensure the most accurate tracking possible, the variable used for assigning a user must be attached to every link. For this reason, you should observe the following points when using social sharing:

- There should be no possibility to edit or view the profile data of a recipient on the landing page of an article.
- Links leading back from the landing page to the newsletter must be avoided.
- There should be no possibility to unsubscribe from the newsletter on the landing page.

After a share, an 'ems.li' URL appears?

We use our own "link shortner" for the links shared in the social networks. A link in the form "http://ems.li/u/abcdef" leads directly to the long version of the article or redirects the user to the destination of the landing page link.

Special case Facebook Sharing

The landing page of a newsletter in Facebook is shared by means of the Open Graph Protocol. For this purpose special tags must be integrated into the HTML code of the landing page and filled via TAL.

Example:

```
<tal:block tal:condition="THIS/getNewsletter/hasSelectedArticle">
        <meta property="og:title" content="TITLE" tal:attributes="content THIS/getNewsletter/getSelec
        <meta property="og:image" content="IMAGE" tal:attributes="content THIS/getSelectedArticle/get
        <meta property="og:description" content="DESCRIPTION" tal:attributes="content THIS/getSelecte
</tal:block>
```

The following attributes must be added to the <html tag> for this to work:

```
xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://ogp.me/ns#"
```

These codes have already been integrated into our sample templates.

Facebook does not share any images that are smaller than 200x200px even if these are specified explicitly.

3.1.5 Birthday eMailings

Objective: Creation of a birthday eMailing

Procedure:

1. The prerequisite is that you have address data in your pool that contain the birth date. Use this to create a new target group with the following configuration:

```
       Geburtsdatum ist gleicher Tag wie today
und   (Geburtsdatum ist kleiner          01.01.1970
oder   Geburtsdatum ist grösser          01.01.1970)
```

2. Create a new event eMailing in the *eMailings* folder. Alternatively, you can also derive this step via the *Start page* by clicking on *New eMailing*.

3. Give the eMailing a descriptive name and subject line. Search for an appropriate template and then click on *Save* to generate a draft for your planned eMailing from the template.
4. Configure the contents of the eMailing:

- Assign *contents to the articles from the article management via the ‘contents* tab using `+`.
- Determine the subject line.
- Check the base data.

5. Select the previously defined target group for your eMailing under *Transmission* and the interval for the transmission (daily). After that, set the `active` checkmark.
6. After clicking on `Save`, you will be taken to the statistics page. The eMailing has now been completely configured and the system will now transmit the greeting daily to the corresponding birthday children.

3.1.6 Content Syndication with RSS/Atom Feeds

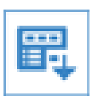
Objective: cross-client distribution of news/articles in the form of RSS/Atom feeds. You can thus provide centrally managed news to all subordinate accounts so that individual clients can display and import these by **Article import**.

Ranges of applications:

- Cross-client content providing.
- Integration of external news sources per RSS/Atom, e.g., news ticker.

****Procedure:** News source

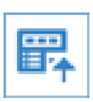
1. Create a new object article export in the *Web* folder.



2. Assign all relevant news/articles to this object and activate by *saving*.
3. Feed thus generated is immediately accessible right after.

****Procedure:** News importer

1. Create an object article import in the *Web* folder.



2. Specify the URL of the source feed in the *Feed URL* field.
3. When the RSS/Atom Feed readers is called, the system displays every feed article that you can import as an article. The images of the article are saved in the *Images* folder.
4. Every RSS/Atom Feed reader is automatically displayed on the system‘s start page.

Tip: You can find the source feeds in the *Preview* display below the preview window.

3.1.7 Emailing templates

The system provides several options for creating dynamic newsletter and eMailing templates. Every eMailing is based on an eMailing template. You copy this template when creating a new eMailing with the aim of subsequently designing content, including customized text and images, for your eMailing. Depending on the respective field of application, you can create templates for the following eMailing types:

- Emailings and Newsletter with articles from the CMS
- Individual HTML and text eMailings

Any combination for a wide variety of fields of application is also possible.

3.1.8 Emailing template components

In order to create a perfect template, it often requires the combination of diverse techniques and template elements. In principle, you can combine the following template elements in an eMailing template:

- Placeholder for any number of articles and news (article slots).
- Open input fields with indefinite specifications (text area or HTML area)
- Dynamic TAL components (properties programmed for automatic actions such as generating a table of contents in a newsletter)

3.1.9 Articles in the eMailing (article slots)

You can place every article in article management directly in the eMailing. The management of article contents mainly takes place in the eMailing, not in the template. If you create a new newsletter based on an eMailing template, assign it the requisite articles.

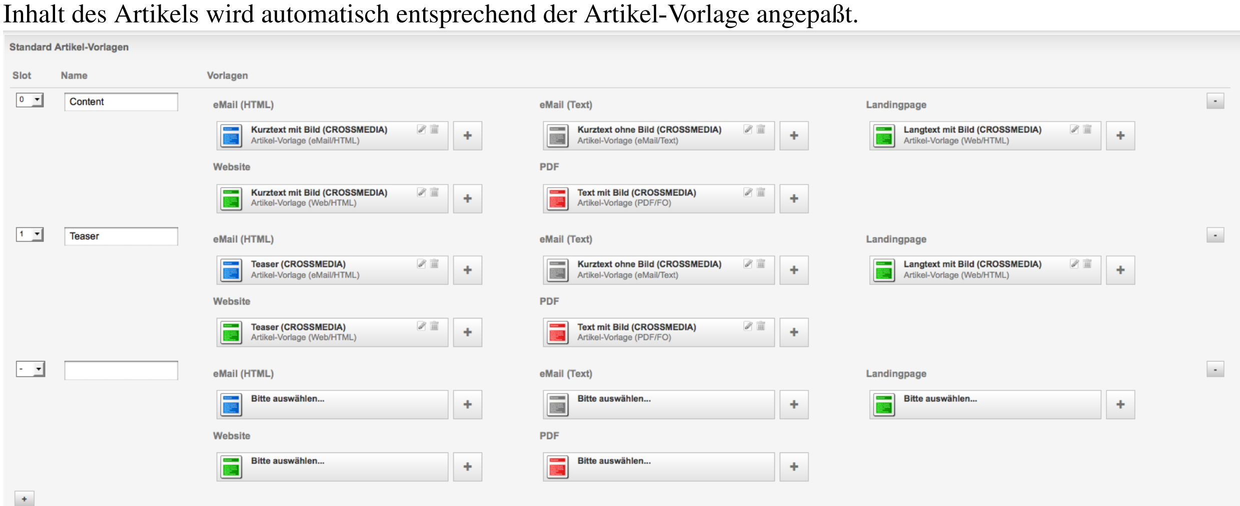
So-called *Slots* are defined in the eMailing template for this purpose. A slot is a placeholder for any number of articles. The exact type and display of an article is specified in the eMailing template. You can define up to 100 different `SLOTS` in an eMailing template. The placeholder variables `{SLOT_0} - {SLOT_99}` are provided for this purpose. You can integrate a slot in all media templates (EMAIL (HTML), EMAIL (TEXT), web/mobile, landing page, PDF, RSS feed and Text-to-Speech).

The system generally provides an automatic cross-medial display for all the information that you manage via article management.

Important information about slots

Individual articles can be subsequently assigned to the eMailing only if one slot at a minimum is available in the template.

In einer eMailing-Vorlage können Sie zu jedem Slot innerhalb eines Mediums (eMail (HTML), eMail (TEXT), Web/Mobile, Landingpage und PDF) eine entsprechende *Artikel-Standardvorlage* definieren. Diese werden dann standardmäßig verwendet, wenn ein emailing mit dieser eMailing-Vorlage erstellt wird. Somit muss sich ein Redakteur beim Hinzufügen eines Artikels keine Gedanken über die Darstellung eines Artikels im eMailing machen. Der Inhalt des Artikels wird automatisch entsprechend der Artikel-Vorlage angepaßt.



Every eMailing template contains a configuration of the display for the following media:

- Newsletter as EMAIL (HTML email)
- Newsletter as TEXT (TEXT email)
- Newsletter LANDINGPAGE (detail page for articles)
- Newsletter as WEB/MOBILE (in the browser or Smartphone as website)

- Newsletter as PDF

If you want to use a newly created article template in the eMailing as a default template, then you have to save this in the *Article Templates* tab for the individual display forms.

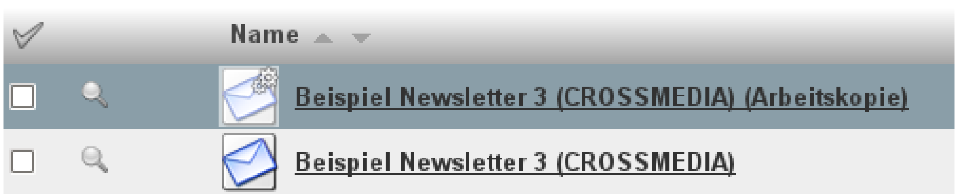
3.1.10 Working copy

A work copy is a copy of an existing object with special functionality. Its purpose lies in the fact that you do not make changes directly on an object but in a copy. These changes are then applied on the original. This prevents publically accessible objects (e.g. a form integrated on a web page) from temporarily becoming unusable due to a misconfiguration. Similar functionality can already be found in the eMailing template whose change must first be transferred to existing eMailings.

Creating a work copy

A work copy can be created from almost any object (exceptions are: query managers, pools, templates, files and images). If you create the work copy directly on the called object, you will be directed automatically to the work copy and can edit this. Alternatively, work copies from several objects can simply be created simultaneously in the overview list. To do this, select the corresponding objects and click on the stylized cogwheel below the checkboxes.

Work copies are visualized in the overview list by a slightly faded object icon with cogwheels (s. screenshot). In addition, the supplement "(work copy)" is attached to the name of the object.



Editing the work copy

The work copy can be edited quite normally like any other object. Just a few special functions are deactivated (e.g. you cannot send any work copy of an eMailing; no display in the flowchart). Furthermore, work copies do no appear in the object selection pickers.

Application on the original

After the changes on the work copy have been completed, they must be applied on the original. A button in the object functions on the top right is used for this purpose. There you will see two stylized sheets, one of which is yellow. After clicking on it, information on the original is displayed once again (name, time of the last edit) and the option of deleting the work copy automatically after use will be given. After clicking on *update original*, the changes will be saved and the work copy possibly deleted - if the user has the right to edit the original object.

3.1.11 Article configuration

By means of the **article type** object the article can be configured completely freely - starting with the data fields including the label, help and input restrictions.

Grouping

The grouping of the fields is organized via tabs. You can create and individually name as many tabs as required. The field types can then be assigned to the corresponding tabs. To do this, click on Edit for the corresponding field, select the required tab and save the field.

Creating fields

If a new field is added to an article type, then this will appear automatically for all articles created from this type.

Deleting fields

Each field can be deleted - even if there are still articles in the system that were created from this article type. Just make sure that no more roles are assigned to the field you are deleting.

Important: If the replacement variable of the field you are deleting was integrated in article templates, this will no longer be replaced after deletion.

Field configuration

Configuration

Name

Name (label) of the field. This is displayed as field name in the configuration overview and article.

Internal name

Name used by the system as a unique name. Is used for generating the replacement variable and for access via TAL. The internal name is generated automatically when entering the name

Help text

This text is displayed in the article overview when the user moves the mouse over the yellow question mark of the field concerned. If no text is entered, then no question mark is displayed

Input help

If set, this is displayed directly below the entry and should provide quick help for the input (e.g. optimum word count, etc)

Category

Category (tab) under which the field is to be sorted.

Mandatory

When activated, the article cannot be saved if this field is empty.

Role

System roles that this field should have. When creating a field type, only one role can be defined. If you edit the field afterwards, all system roles can be assigned (s. *Roles*).

Data type

The data type is the actual configuration of the article. The fields to appear in the subsequent article object are edited and grouped here. The type - provided that no article has yet been created with this configuration - can be changed later.

The following field types can be added to an article:

- Single-line input: Provides simple single-line input
- Multi line input - Generates an input field for multi line text
- Multi line input (formatted): Generates multi line input embedded in a WYSIWYG editor
- Multi line input (HTML editor): Generates multi line input (formatted) including a complete HTML editor
- Single selection : Generates a dropdown list
- Multiple selection: Generates a checkbox list
- Image: Generates an image picker
- Image (List): Generates a list with images
- Container: Generates a container-picker
- Container (List): Generates a list with containers
- Url: Generates a URL field
- Link list: Generates a list with links
- Values (List): Generates a list of simple values
- Date with time: Generates an input option of date and time with calendar
- Boolean (true/false): Generates a checkbox
- Color: Generates an input field for color with color picker

In addition, there are also special field types for designing and documenting the article type.

- Documentation: Displays the text input in the default field
- Documentation (Html): Displays the Html-formatted text that was entered in the default field
- Separator: Generates a line for design-related separation of the fields

Options

Same data types (single/multiple selection) allow additional options These can be configured here.

Extended

After saving a new data type for the first time or editing an existing data type, you will find two more tabs where additional settings are possible.

Default

Here, you can predefine a default value with which the field is prefilled when creating a new article based on this article type. In the case of selections - i.e. fields with a selection option - the value set here is preselected.

The documentation is a special case. The text entered here appears as plain text in the articles.

Fallback

Here, an existing field can be set as fallback. In this case, its value will be used if the original field remains empty.

Important: Depending on the fallback configuration, unintended events could occur under certain circumstances. The configuration should always be checked before use.

Restrictions

In the case of fields with text input, it is possible to define restrictions or defaults and to force these. The system provides the following types for this purpose:

- Character limitation: Specifies the maximum number of characters that can be input in the field. HTML tags are counted as well
- Only numbers: Enforces that only numbers can be input in this field
- Individual: It is possible to configure completely individual restrictions. These have to be defined by means of a regular expression

After creating a restriction, an error message can be specified which will be displayed if the restrictions are violated. In this case, the article is not saved.

If you specify several restrictions for a field, all restrictions must then be observed during editing.

Important: Restrictions are only possible for the input field types (single line, multi line). The "Only numbers" restriction is only possible with single line inputs.

Integration

Replacement variable

The content of the field in article templates can be accessed via the variable shown here.

Roles

Roles are used for notifying the system about which field should assume which task or function. If you assign the role `Headline` to a field, for example, the system will use the value of this field to fill in the `HEADLINE` replacement variable in article templates. If no value is entered in this field and a fallback is configured, the system would then use the value of the fallback for the variable.

In this way it is possible to store customized texts for all display forms or to write optimized headlines for all social networks. You can, however, just as well assign all headline roles to a single field, for example, which then represents the headline in all cases (article, Facebook, PDF, etc).

3.1.12 Release process

To achieve the double verification principle, the so-called "Approval process" can be used. In the course of this process, the eMailing still needs to be accepted by an authorized person prior to transmission and approval for delivery must be granted.

The release process is activated in the eMailing templates and then applied to the required eMailings. To do this, go to the template on the "Release" tab and set the checkmark to *Release process activated*. Then save the eMailing. You will now see a list of all users and roles who can be assigned the right to release. Select here the relevant user or role. When selecting a role, all users to whom this role was assigned have the right to release.

Important: At local level, only local users or roles can be selected, and at global level, only global users or roles.

Important: Please make sure that the approval process forces a double verification principle. This means that an approval cannot be requested and accepted by the same person.

In the eMailings for which the release is required, the *"Transmission" tab* is replaced by the *"Request release" tab*.

Sequence

As soon as the email is ready for transmission, click on the *Request release* tab. Once you have selected the transmission time and target group as usual, you will be taken to the transmission preview after clicking on *Release preview*. In addition to the normal information, there is a new *Release* tab. After clicking on this tab, you will find a list of all users from whom a release can be requested. If the user himself can also process the release, he will not appear in the list.

Once (one or more) users have been selected and the eMailing saved, it is copied into the *Transmissions* folder and waits for confirmation or rejection of the release. In this process, the eMailing in the *Transmissions* folder is marked by a separate icon.

The selected users are informed by email that an eMailing for release is ready and who requested the release. It is possible to jump immediately to the eMailing via a link in the email - after logging in, you come directly to the transmission preview and can view the eMailing as well as all relevant settings.

If the user considers the eMailing to be correct, he can issue the release in the *Release* tab. If the eMailing contains errors or anything of that kind, the user can reject the release in the same way. In both case, an optional notification can be specified.

Rejection of the release

In this case, the eMailing is deleted from the transmissions folder. The user who requested the release will receive an email informing him that the release was rejected, who rejected it and possibly why it was rejected. The user can jump back to the original eMailing via a link and make the necessary changes and then request the release again.

Issuing of the release

Here, the user who requested the release will receive an eMailing with the confirmation, the person responsible and possibly another message. The user can now return directly to the transmission preview via a link in the email and start the transmission as usual.

3.2 Scoring and profiling

3.2.1 Profile- and Activityscoring

New interested parties first register in a data pool (generally anonymously) using a form with a few items of information. At this point, an insightful evaluation of the quality of the acquired address is not possible. Consequently, a primary goal is to enrich the available, generally rudimentary information, which should preferably be automated. The system offers a tool for lead nurturing in the form of a dual scoring model in which the profiles are evaluating according to two primary criteria.

The system distinguishes on one hand between profile scoring that uses the explicit data of the profile for evaluation. Information on the position, company size, budget, requirements/time window (BANT analysis) , etc. are requested here at any desired depth, analyzed, and depicted in an evaluation matrix.

At the same time, an implied scoring is used that evaluates the individual behavior of the interested party. Information about the interest, activity, areas of interest and click behavior is identified and evaluated here in a freely definable diagram.

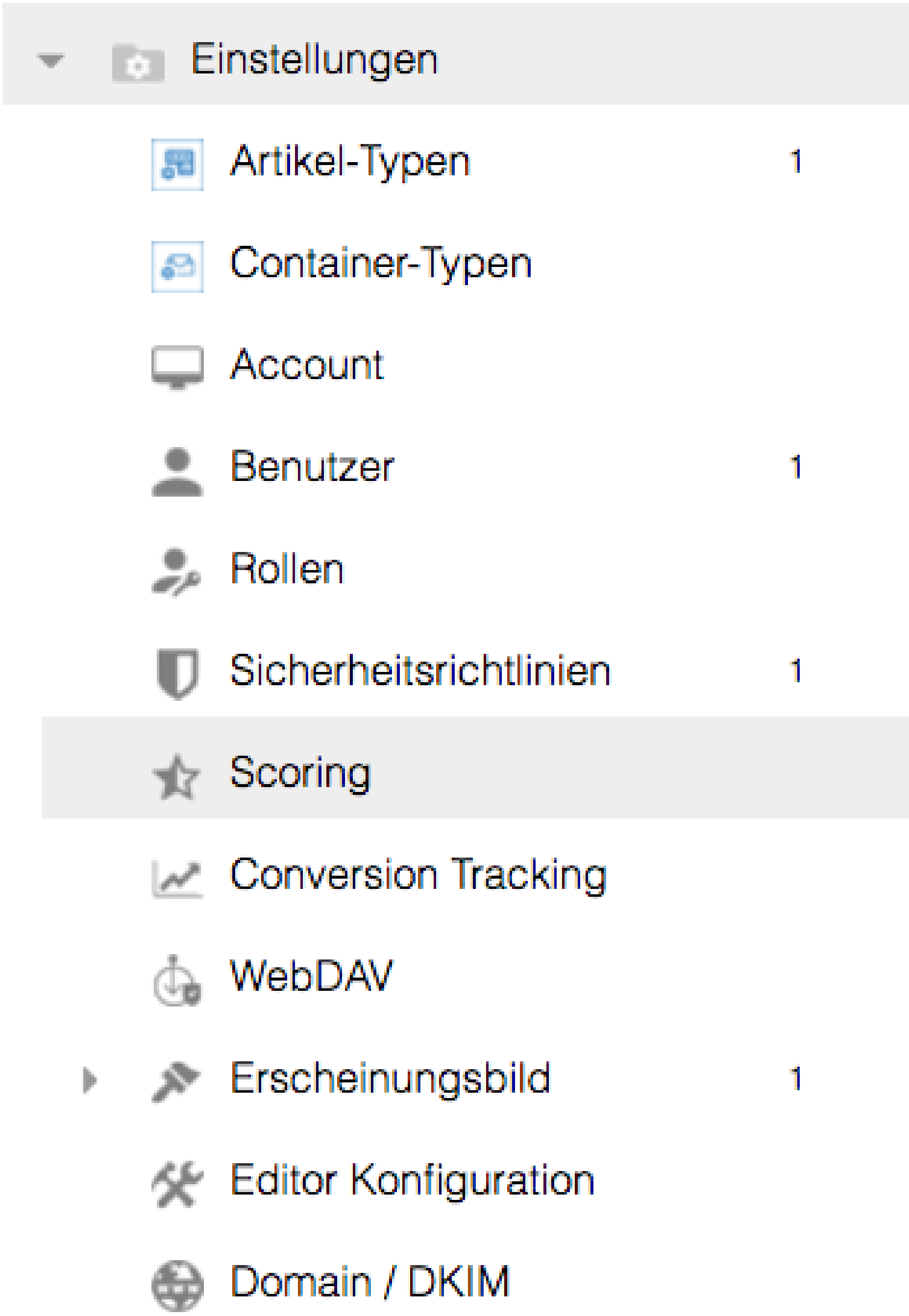
In the area of implicit scoring, it is possible to distinguish between the activity (click behavior, paths, downloads, interface) and interest (selected content). Lead generation can hence be qualified either preponderantly from quantitative considerations (such as the frequency of an action), or from qualitative considerations (what does the interested party find interesting).

Multiple scoring can also be generated, that is, several "products" (qualitative scores) can be combined by appropriately adjusting the configuration. This functionality enables different scores to be processed all at once.

Zu beachten ist, dass ohne ein personenbezogenes Tracking des Systems, die Scoringfunktionalität nicht nutzbar ist.

Technical description

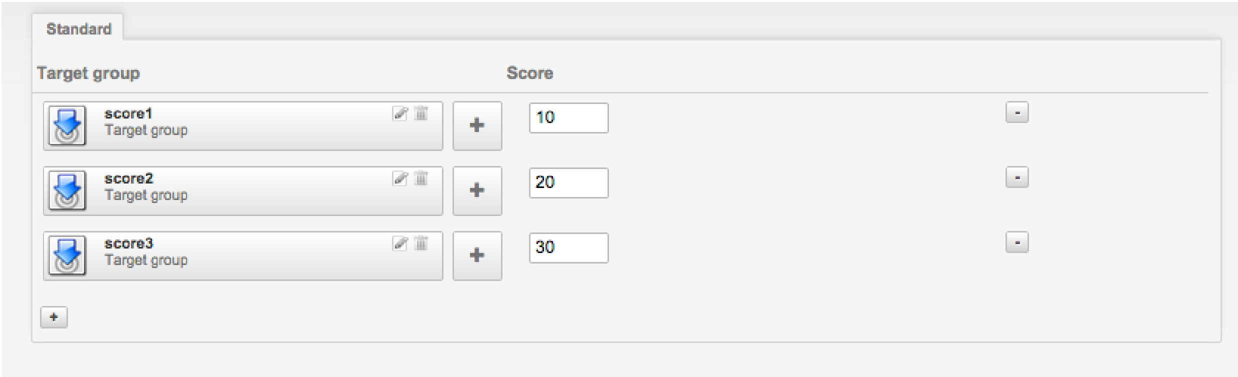
Default criteria are set up and adapted by choosing and overriding the proposed values in the area of scoring using settings on a global and client level.



Profiles scoring

The entered profiles are scored using the functionality of the target groups, which is correspondingly expanded for this purpose and provided with new technical features.

Each target group defined in this manner can be assigned a score.



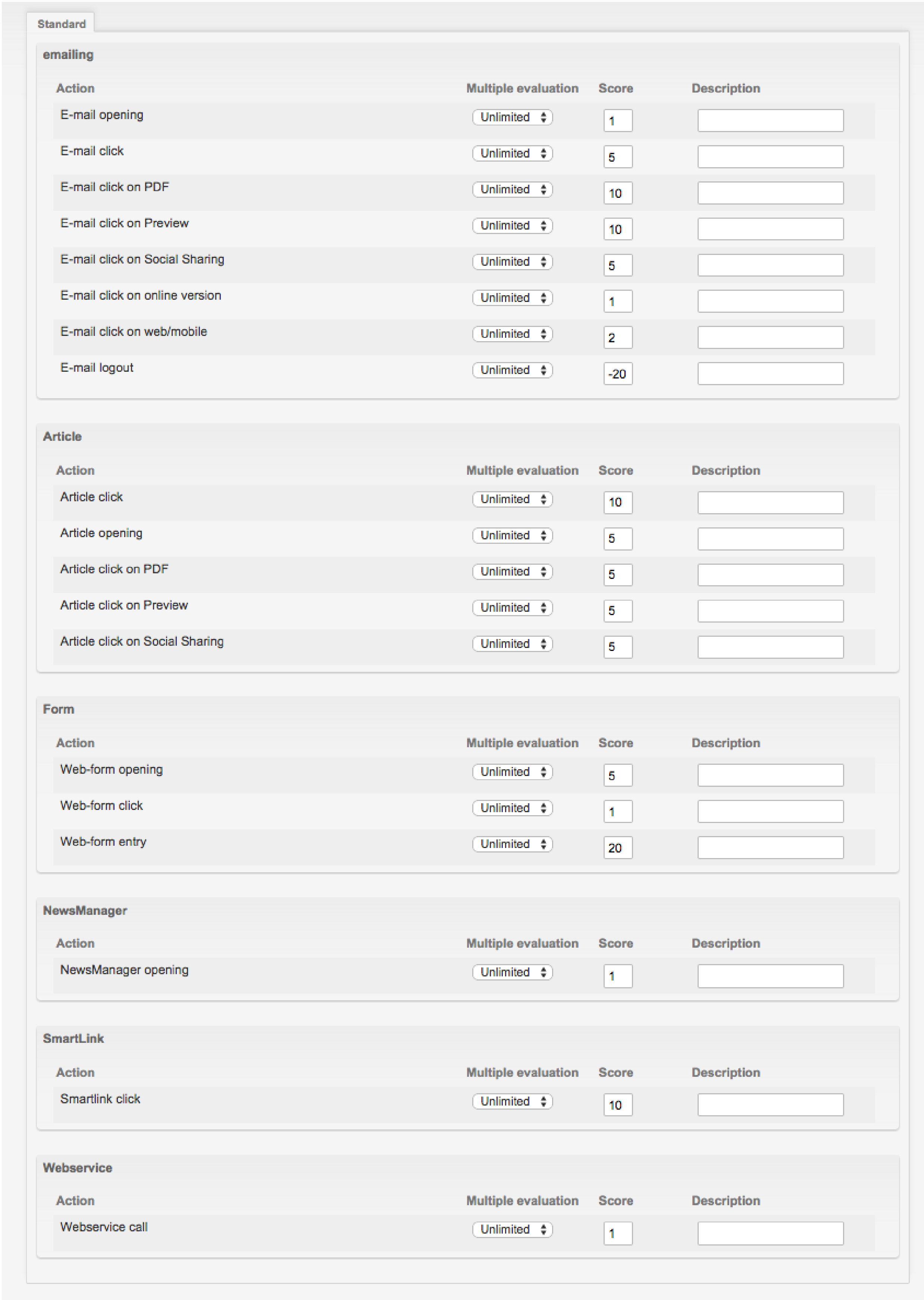
The value which is set here corresponds to the number of points that is evaluated in the subsequent scoring matrix (cumulative). The profile scores are calculated for individual profiles every three hours.

Activity scoring

Activity scoring is based on object-based information carriers (eMailing, articles, forms, LeadPage, SmartLink) as well as information exchange via the web service interface.

The system provides baseline assessments as desired. Whether or not, and to what extent, this type of evaluation can be depicted in the individual customer analyses needs to be determined on a case-by-case basis. Profile scoring definitions need to be individually created in each client.

The rules for setting up, inheritance and individualization are the same as existing familiar functionalities, that is, setups on the global level initially applies to all subordinate clients. The same holds true for inheriting clients on the object level by means of templates.

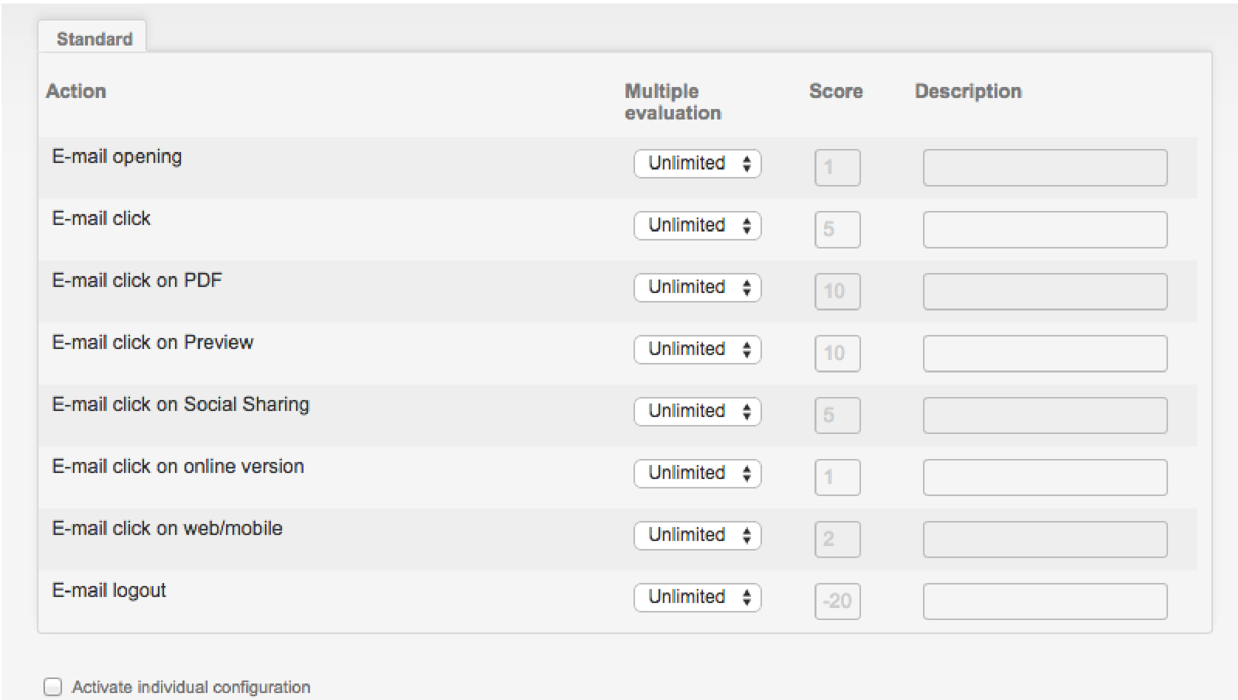

Standard
emailing
Action
Multiple evaluation
Score
Description
E-mail opening
Unlimited
1
E-mail click
Unlimited
5
E-mail click on PDF
Unlimited
10
E-mail click on Preview
Unlimited
10
E-mail click on Social Sharing
Unlimited
5
E-mail click on online version
Unlimited
1
E-mail click on web/mobile
Unlimited
2
E-mail logout
Unlimited
-20
Article
Action
Multiple evaluation
Score
Description
Article click
Unlimited
10
Article opening
Unlimited
5
Article click on PDF
Unlimited
5
Article click on Preview
Unlimited
5
Article click on Social Sharing
Unlimited
5
Form
Action
Multiple evaluation
Score
Description
Web-form opening
Unlimited
5
Web-form click
Unlimited
1
Web-form entry
Unlimited
20
NewsManager
Action
Multiple evaluation
Score
Description
NewsManager opening
Unlimited
1
SmartLink
Action
Multiple evaluation
Score
Description
Smartlink click
Unlimited
10
Webservice
Action
Multiple evaluation
Score
Description
Webservice call
Unlimited
1

Each action can be assigned an individual score which is considered and evaluated in the scoring matrix.

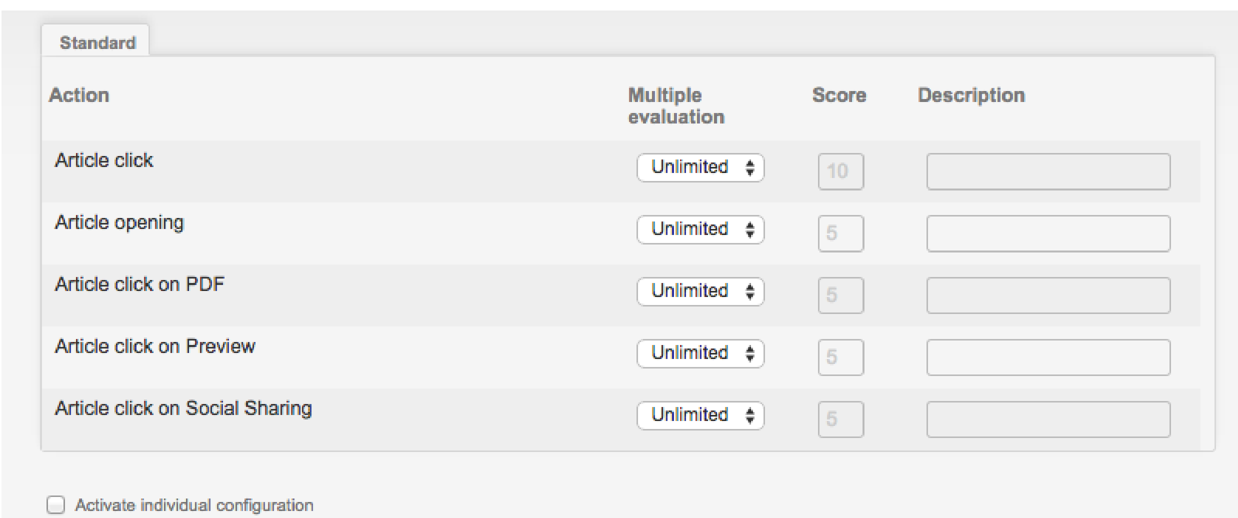
Frei konfigurierbar ist die Bewertung von Mehrfachbewertungen. Hier kann via Dropdown der Zeitraum festgelegt werden innerhalb dessen ein bestimmter Activity-Score nur einmal vergeben werden kann. Wenn nach einer Bewertung der angegebene Zeitraum verstrichen ist, kann ein Profil diese Bewertung erneut erhalten.

Individuelle Scoring Werte können zudem auch auf der jeweiligen Objekt-Ebene vergeben werden.

Mailing:



Article:



It should be remembered that one click yields several scoring assessments due to overlapping. The effect of the click or relative dependencies is defined in the next table.

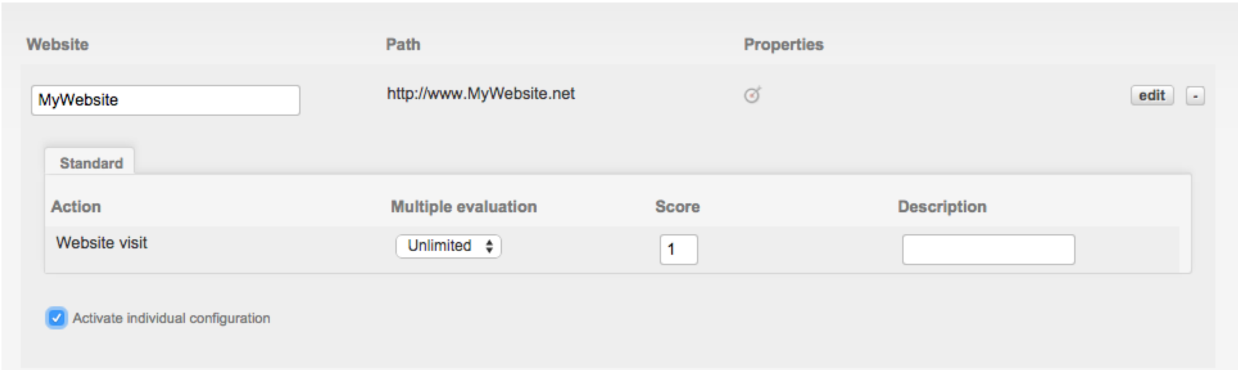
For example, clicking on a landing page within a mailing generates a scoring entry (click) within the mailing, as well as an entry (opening) within the article.

In this way, several scoring entries can be defined with a single action. If this is not desired, 0 can be indicated as the score in the lower-level object (the article in this example).

Action in the eMailing	eMailing	Article	Form	SmartLink
Opening the eMailing	Opening	•	•	•
Click on link in the eMailing	Click	•	•	•
Click on form in the eMailing	Click	•	•	•
Click on SmartLink in the eMailing	Click	•	•	•
Display eMailing as PDF, preview	Click	•	•	•
Display eMailing as social sharing	Click	•	•	•
Display eMailing as online version	Click	•	•	•
Display eMailing as Web/Mobile	Click	•	•	•
Click on log out	Unsubscribe	•	•	•
Click on article landing page (auto) in the eMailing	Click	Opening	•	•
Click on article landing page (ext) in the eMailing	Click	Opening	•	•
Click on article PDF, preview, sharing in the eMailing	Click	Click	•	•
Click on link in the article	Click	Click	•	•
Click on form in the eMailing	Click	•	Opening	•
Click on SmartLink in the eMailing	Click	•	•	Opening

In order to detect or take into account downloads or landing pages by scoring, the corresponding SmartLinks need to be included in the scoring. In general, SmartLinks can be used as a multifaceted instrument, especially in content scoring, since a wide range of targets (including external targets) can be evaluated, and the obtained values can be integrated into the scoring matrix.

WebTouchPoint:

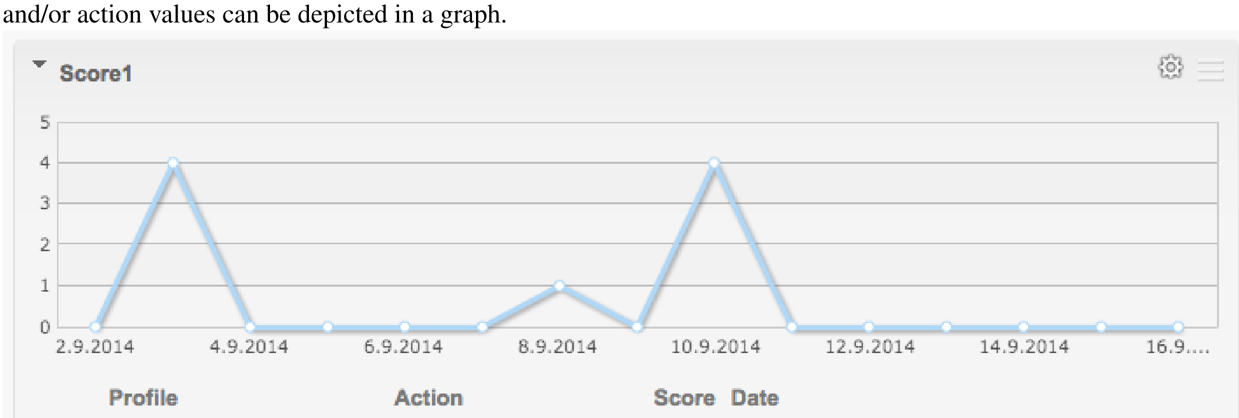


SOAP:

Sie können auch via SOAP API Profilen einen Activity-Scoring-Wert zuweisen. Hierfür steht Ihnen die Methode: `addScore` aus der Profile-WSDL zur Verfügung. Sie benötigen für den Aufruf die Profil-ID, die ID der Scoring Gruppe und den zu vergebenden Wert mit individuellem Namen.

Widget

In the area of scoring, an individually configurable widget can be created on the desktop in which defined profile and/or action values can be depicted in a graph.

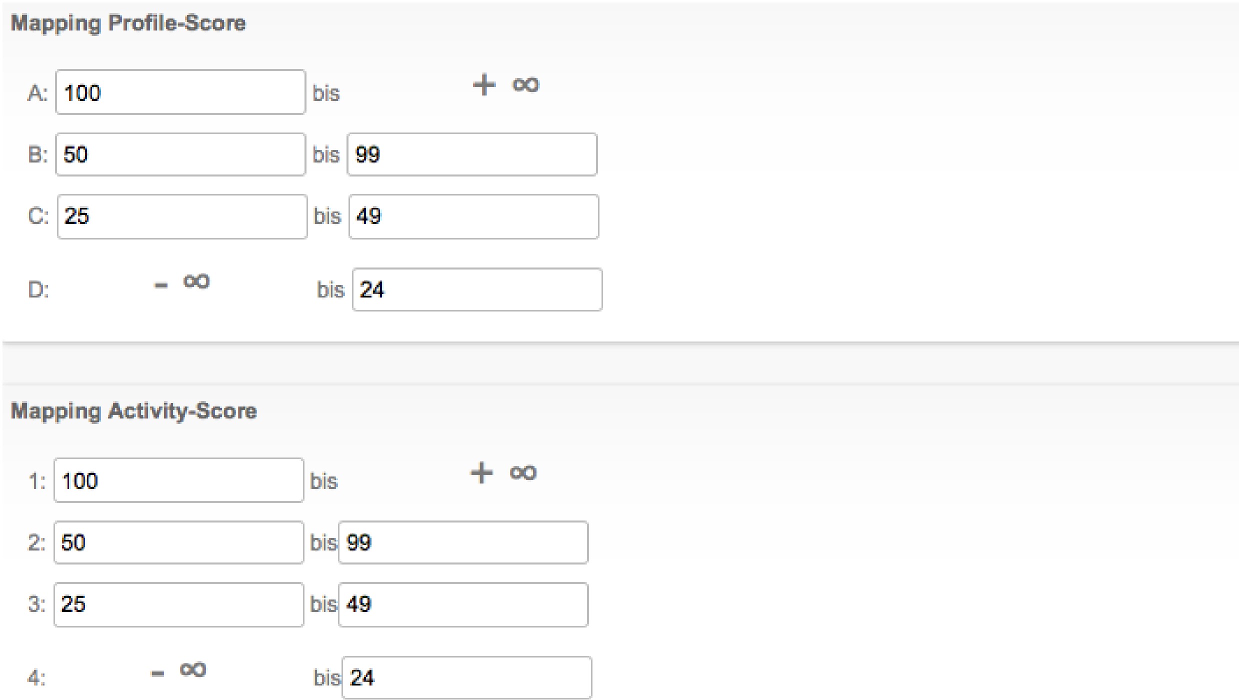


It should be noted that individualized scores are no longer valid after a reset to system defaults

Mapping activity and profile scoring

The final evaluation of the scoring matrix is configured in the area of settings.

Here, the thresholds for the 4 profile scoring columns (A-D) and the 4 activity/interest scoring lines (1-4) are specified.



As a result, a matrix on one or more lead profiles is obtained

For example, a threshold for a lead's readiness to transition can be individually specified; for example, all leads which possess the criteria A1, A2, B1, B2 ... are manually processed and transferred to a CRM, etc.

Whether absolute scores or simplified A/B/C/D and 1/2/3/4 categories, they can be requested by means of target groups and exported via the Web service interface or to the request manager.

Mapping of local and global scoring

To be able to perform scoring evaluations across multiple clients, you can map a global scoring group with one or more scoring groups from any client. The global scoring group now considers the values of all involved (mapped) scoring groups.

The configuration takes place in the global scoring configuration under the tab „Settings“ in the sub-tab „Groups“. Here the mapping configuration can be opened and extended by clicking on „mapping“.

Profile anhand von Score-Werten selektieren

In Zielgruppen können sowohl der Activityscore als auch der Profilescore ausgewertet werden. Bei den Activityscore-Werten muss berücksichtigt werden, dass zwischen Profilen mit Activityscore = 0 und Profilen welche nie irgendeinen Scorewert hatten unterschieden wird. Um alle Profile zu selektieren die weniger als 0 oder exakt 0 oder aber noch nie irgendwelche Scoringpunkte hatten muss die Selektion lauten: “Alle Profile des Pools - und nicht - Activityscore > 0”

3.2.2 Profile Matching and Profile Tagging

Using the Profile Matching function, articles in an e-mail can be sorted automatically according to defined criteria and the recipient profile within a slot. Sorting takes place after article individualization, so that the profile matching configuration of the alternative article, if available, is used for the calculation.

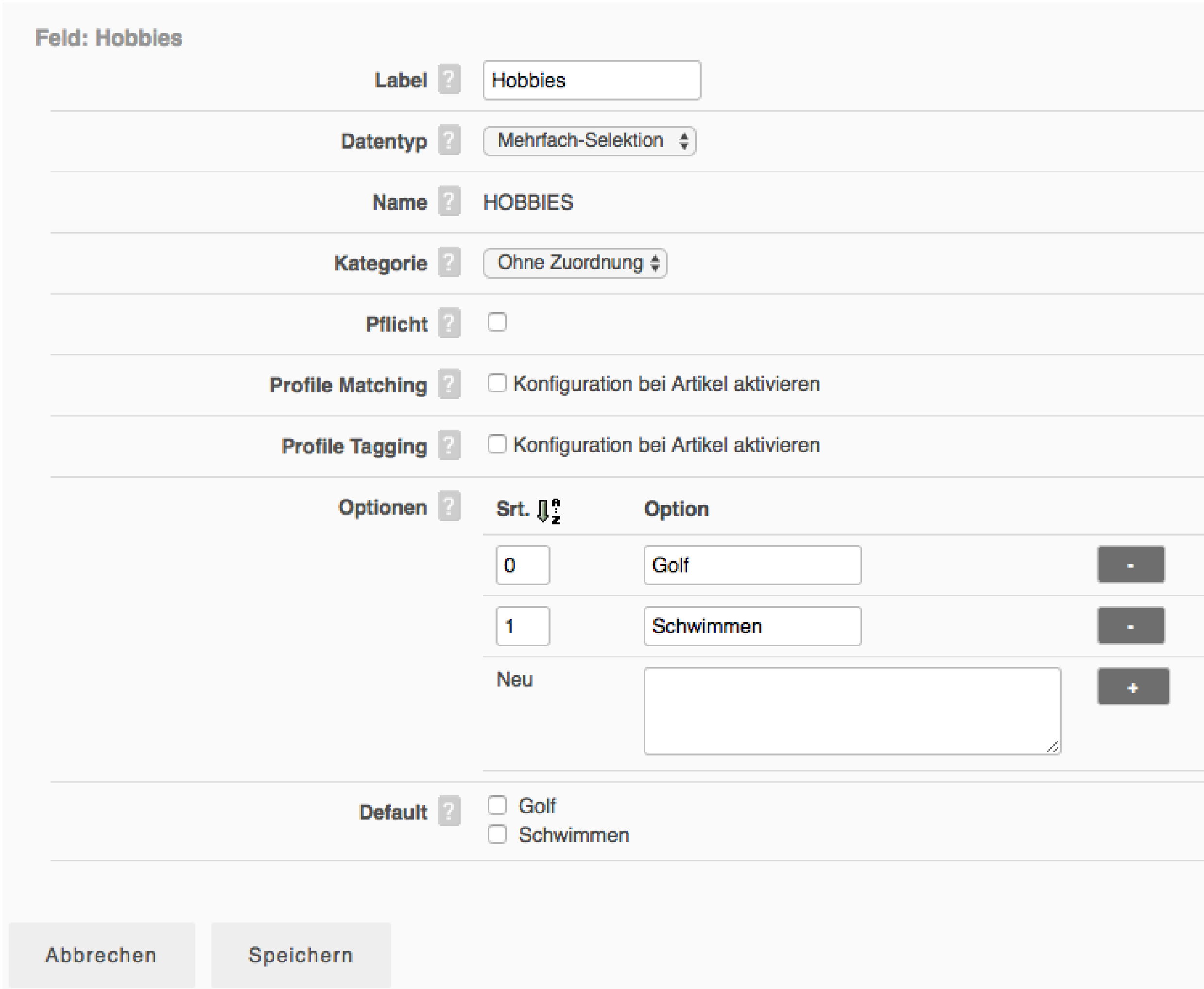
Using the Profile Tagging function, profiles can be manipulated on the basis of their behavior. Viewing the landing page of a football article can lead to an enrichment of the profile with the interest matching the article, in this example „football“. The manipulation takes place the moment the profile calls up the landing page of a selected article. It might initially be useful to create separate pool fields for the Profile Tagging function so that the original profile data can still be viewed separately.

Selecting relevant pool fields

The first step is to select the pool fields used for calculating the sorting and for profile enrichment. This is possible using the option „Profile Matching“ and „Profile Tagging“ for the respective pool field.

The following field field types are supported:

- Salutation
- Country
- Single selection
- Multiple selection
- True/false (Boolean)

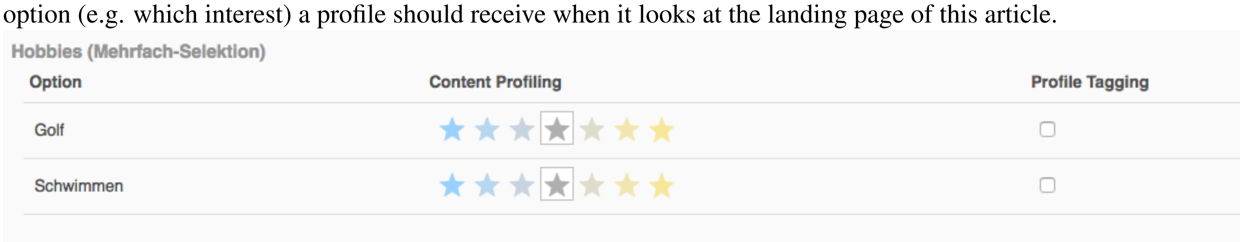


Configuring Profile Matching

The profile criteria can now be configured for the article. Below the "Profiling" tab, a value ranging from "not relevant (-3 points)" up to "very relevant (+3 points)" can be assigned per pool field.

Configure profile packing

Profile tagging can now also be activated in the article under „Profiling" for the selected pool fields. Select which option (e.g. which interest) a profile should receive when it looks at the landing page of this article.



Configuring slot

To use automatic sorting, the slot must still be configured. This is already possible either in the newsletter template in the "Article templates" tab using the cogwheel icon in each slot or in the visual editor. There, you will find the aforementioned cogwheel icon at each slot.

There are four ways to sort slots:

- Manually (articles are sorted as set)
- Automatic (articles are sorted in descending order using the calculated profiling)
- Auto. Top-X (like automatic, except that the number of articles displayed here only corresponds to the number set in the input field that appears)
- Auto. positive (like automatic, but here articles are only displayed for which the relevance calculation resulted in a positive final total)

Calculation of Profile Matching

The recipient profile is used to determine which profiling criteria are relevant for the recipient. All of the set criteria are added up to form a sum which determines the placement of the article. If several articles have a same final sum, the sequence of the configuration is the determining factor in this case.

Example:

The following settings are made for an article:

- Salutation "Mrs" is very relevant
- Country "Germany" is very relevant
- Country "Switzerland" is not relevant

Furthermore, there are three profiles in the pool for which the following attributes are set:

- Profile A
  - Salutation "Mr"
  - Country "Germany"
- Profile B
  - Salutation "Mrs"
  - Country "Switzerland"
- Profile C
  - Salutation "Mrs"
  - Country "Germany"
- Profile D
  - Salutation "Mr"
  - Country "Switzerland"

The following calculation would result in the aforementioned article example:

- Profile A: 3 points (salutation "man" 0 points; country "Germany" +3 points)
- Profile B: 0 points (salutation "woman" +3 points; country "Switzerland" -3 points)
- Profile C: 6 points (title "Ms" +3 points; country "Germany" +3 points)

- Profile D: -3 points (salutation “Mr” 0 points; country “Switzerland” -3 points)

The article would accordingly be placed toward the top for profile C (of course in comparison to other articles in the mailing). For profile B, the article would hence be placed toward the bottom, comparatively speaking.

If the “Auto. positive” sort was set, the article for profile D would not be displayed at all then.

3.3 Generating addresses - via web forms

3.3.1 Creating and Inserting Web Forms

Web forms are particularly relevant to the recording of feedback from recipients when planning permission marketing campaigns. Via these forms one learns whether clients would like to continue to receive data while the campaign is ongoing or not. Or one provides interested parties whose addresses have not yet been input a basis for transmitting their data.

3.3.2 The web form‘s application range

You can use web forms that are created using simple means per editor for a wide range of purposes:

- as newsletter application forms
- as profile editing form in eMailings
- as contact and order forms
- as sweepstakes forms
- for opinion questionnaires and marketing research
- as callback and infoservice web forms
- for booking and catalog queries
- etc.

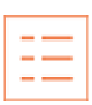
3.3.3 Inserting Web Forms

A form that you have created can be inserted directly in the website or in the eMailing. Several procedures are available for this purpose:

- per popup window on a website or in the eMailing. Web forms in the eMailing can be empty or filled out already with existing profile data of the recipient
- per iFrame directly in the website.

3.3.4 The Form Configurator

The system provides the option of creating customized web forms using an integrated, efficient web form configurator - without programming skills. In doing so, the system transfers the following functions automatically:



- Data maintenance through direct connection to the pool (database)

- Check of the form's contents for plausibility
- Error messages for key errors
- Query Management (automatic routing and processing of form queries)
- Opt-in procedure for permission marketing
- Confirmation page and confirmation email
- Routing mechanisms for linked forms
- Prepopulated form off the eMailing
- Multilingual forms

3.3.5 Creating a web form

Objective: Creation of a web form with direct connection to the pool (database) for generating addresses on websites or in eMailings.

Procedure:

1. Create a new form in the *Web form* folder:

2. Select the pool that you would like to assign the form to
3. Give the form a descriptive name (e.g. "Seminar booking") and enter an optional description text (e.g. the future folder of the form).
4. In the *Configuration* area you will find all necessary settings for the design, language and contents.
5. Under *Fields* you can add or remove form fields. You can use the pool's every field in the form. You must use without fail fields that are defined in the pool as place holders. If you require additional fields in the form, then you will have to have created them in the pool in advance.
6. Finally, integrate the form in the website or eMailing, see *Inserting web forms in the website or in the eMailing*.

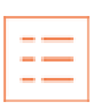
Attention: You can only create a new form in the web forms folder.

3.3.6 Linked web forms

Objective: In some cases it is advisable to distribute information to several forms. Examples: that various follow-up actions are linked with it or that the amount of information is too large for a form. The aim is to be routed to a successive form immediately following the validation of the first web form. Thus one could, for example, enter the minimum data, email addresses and names of the newsletter's interested parties in a first web form and use a second one for the entry of additional optional information - e.g., involving questions about interests and hobbies.

Procedure:

1. Create all required forms, see: `create_a_webformular`

2. In the *Configuration* area below the first web form, set *Routing* to the subsequent form required.

3. The string of routings can continue in any order.

Fields that were already filled out and recur in the follow-up form can be completed automatically in order to avoid a loss of data. If data are no longer displayed in a follow-up form, these data are then protected from overwriting as well.

Tip: In the first web form only the absolutely necessary data query - like e.g. email address and name of newsletter subscribers - and further desired information on following web forms shift.

3.3.7 Prepopulated web forms in the eMailing

Objective: Web forms embedded directly in the mailing - e.g. in contact, callback or order forms - can be displayed prepopulated. With this, one provides eMailing recipients the greatest ease in filling them out. For this purpose, you can use the data stored in the pool.

Procedure:

1. Create a form, see: *Creating a web form*

2. Integrating the web form in the eMailing. See: *Inserting web forms in the website or in the eMailing*
3. The form is prepopulated with the data which are already present.

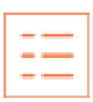
Attention: You can test this function only during actual transmissions (test transmissions).

3.3.8 Customized form template

Objective: Option for customizing the design of web forms with your own HTML template - as an extension of the standard design options.

Procedure:

1. Create a web form, see: *Creating a web form*.

2. In the *Customized template* area, download the sample template by clicking on *Download* in *Current HTML template*.
3. Design the template as you wish in an HTML editor (e.g. with Adobe Dreamweaver or Adobe Golive) copy the HTML code into the source code view (click on *Source code*) or upload the HTML file. You will find the upload area in the bottom left corner. Alternatively, you can also edit the template directly with the WYSIWYG editor or with the HTML-Editor in the system.
4. Activate the template by checkmarking the checkbox *activate HTML template*.
5. Once saved and checked, the new template is enabled automatically.

Important: All expressions in curly brackets must be preserved in the template. If you subsequently change the configuration of the fields, then it too is to be updated in the template. Otherwise data loss can occur. The XHTML code has to be edited accordantly as well. The system automatically attempts to correct faulty code. This can lead to incorrect results under certain circumstances. Therefore, check the representation of your file in the event of any corresponding error messages. Or correct the error in your original and upload the file again.

Tip: Transferred Post/Get variables can be accessed via TAL: REQUEST/variable name

---

3.3.9 Newsletter registration form

Objective: Newsletter registration forms are a special form of web form. After registration a confirmation email is always sent to the interested party including one of the following acknowledgement procedures:

- *Single Opt-In*: the email address of the recipient does not have to be confirmed by the subscriber.
- *Confirmed Opt-In*: after filling out the form, the system sends a confirmation mail to the subscriber with information as to how he can cancel to his subscription.
- *Double Opt-In*: after filling out the form, the system sends a confirmation mail to the subscriber. This contains a link that the subscriber must use to finally confirm his registration.

Procedure:

1. Create a form, see: *Creating a web form*

2. In the "Configuration" tab, activate *Register for Double Opt-In* as *registration procedure*.
3. Create an eMailing that contains the confirmation link `{$GRANT_URL}` This variable is filled automatically with the appropriate confirmation link by the system during transmission.
4. Set this eMailing as *confirmation email* in the "Configuration" tab.

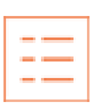
Important: If the registration procedure is activated in the form as *Double Opt-In*, you can first use entries for the transmission after receiving the explicit confirmation by the recipient. On legal grounds you should always use the double opt-in procedure.

3.3.10 Contact form with newsletter option

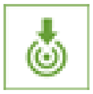
Objective: Contact forms with newsletter option are a special type of web form. The system will only generate a confirmation mail for the interested person if he selected the newspaper option e.g. "I wish to receive a newsletter" prior to the transmission.

Procedure:

1. Create a field for parties interested in the newsletter (e.g. *Newsletter*) in the pool as a Boolean field (true/false).
2. Create a form, see: *Creating a web form*

3. Create a target group with the criterion *Newsletter* is equal *true*. The field for *newsletter* has to be created previously in the pool as well as in the form.



4. On the form below the *Configuration* enable the *Double Opt-In* procedure.
5. Set a confirmation email that contains a confirmation link `{$GRANT_URL}`.
6. Set the new target group of the interested person in the newsletter for the *confirmation email*. Consequently, a confirmation email will only be sent for this target group.
7. Check whether the set confirmation email contains an option for activation.

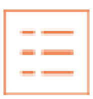
Important: All entries of this form type, which come into play without selecting the newsletter option, remain unconfirmed for an unspecified time and thus are not loaded along with the newsletter. This requires that the newsletter = "Yes" is defined in the target group for the transmission.

3.3.11 Login form for Profile Center

Objective: Login forms are a special type of web form. They only serve to authenticate (validate) and do not make any data changes. With the aid of the login form (validation form), one can allow an authorized person access to one of the following (prepopulated) forms. Every specified field in the login form is matched with the database and only if there is a complete and unequivocal match with a record does routing take place. This mechanism is often applied in the setup of profile centers.

Procedure:

1. Create a form, see: *Creating a web form*. The web form created can contain only the fields that are required for authentication (e.g., for the entry of an email address and password).



2. Under *Routing*, set the actual target form.
3. Under *configuration* in the field *registration procedure*, set *Only Login*.
4. In target form under *Configuration* set the field *uniqueness criterion* to *Profile ID* and select the corresponding login form in *Login form*.

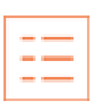
Important: Only after successful authentication does the system route to the target form. This is then prepopulated with the profile data which are already present.

3.3.12 Unsubscription form for the newsletter application

Objective: The unsubscribe form for newsletter cancellation is a special type of web form. After the registration, the system marks the corresponding registration as *unsubscribed* in the pool.

Procedure:

1. Create a form, see: *Creating a web form*



2. In the form below the *configuration*, activate the *registration procedure* as *unsubscribe*.

Important: For security reasons we recommend not to integrate this form directly in the website, rather in an email that sends another form as a confirmation email to the intended recipient. A misuse of this function can thus be prevented.

3.3.13 Query management for web forms

Objective: Automatic delivery of pool entries to specific persons for further processing.

Procedure 1:

1. In the form itself, it is possible under *Configuration* to set request forwarding.
2. To do this, you just need to select a report mail that is then personalized with the data of the profile entered.
3. Finally, specify the recipient email address(es) (multiple addresses must be separated by commas).

Procedure 2:

1. Create a *Query Management*:

2. In the object; configure the *Report type* for *Entries* and define all necessary details for *Fields* and *Report email* etc.
3. By the *target group* and *pool* you determine the entries for which the system should trigger a link.
4. You can configure as many query management objects as you like for different target groups and recipients.

Tip: In forms with high traffic, set the reporting to *daily* instead of *immediately*. Then the system bundles all of a day‘s queries and sends them grouped in the form of a CVS or Excel file.

The *Grouping* function under *Report email* makes it possible to vary the recipients depending on a pool field, such as country.

3.3.14 Inserting web forms in the website or in the eMailing

The system provides the following different options for integrating web forms in the website or in the eMailing. You can also find all example codes directly in the **form** below the **Integration** tab:

Integrating per popup

In this case, a link is inserted in the website or in the eMailing, which opens the form in a window of its own.

```
<a href="[URL AUF FORMULAR]" target="_blank">Link auf Formular</a>
```

Insert as IFRAME

```
< iframe frameborder="0" src="[URL AUF FORMULAR]" width="500" height="800">
     Ihr Browser unterstützt keine iFrames.
     <a href="[URL AUF FORMULAR]" target="_blank">weiter</a>
</iframe>
```

(`URL ON FORM` stands for the URL of the corresponding form. One can find the link to the form in the form‘s detailed view and the integration)

Inserting in the eMailing with prepopulated fields

We make a special personalization variable available for this purpose:

Code for email (text variant):

```
{$FORM:ID}
```

Code for email (HTML variant):

```
<a href="{$FORM:ID}" target="_blank">Link auf Formular</a>
```

> **Attention:** If the pool of the form and the pool of the transferred profile involve different pools, only the default fields predefined by the system are then prefilled

3.4 Tracking, web personalization and statistics

3.4.1 Behavior Targeting with Checkpoint

Objective: You want to log actions of the recipients personally after exiting the eMailing.

With the aid of so-called checkpoints you can record a recipient‘s following actions:

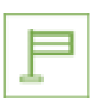
- Recipient enters a special website post-eMailing.

  This site must not connect directly to the eMailing.

- Recipient accesses special URL (link), e.g., to download a document or to click on a link to an external site.

Procedure:

1. Create a checkpoint below the *Tracking* folder:

2. You can create several links within the checkpoint, which you can use for subsequent link tracking.
3. You can integrate the checkpoint at the target location in the following ways:
   - per link (link tracking)
   - per image (image tracking)

   The corresponding code can be found under *Integration* of the checkpoint.
4. The group of recipients that is linked with this object can be determined at a later point. An evaluation of the link is performed via the object‘s statistics.
5. Checkpoints can be summarized in a report for a clearly grouped evaluation.

Caution: In this connection please be aware of the legal situation in your country relevant to data protection dealing with personal data management. Identification and profile detection based on cookies and thus does not return any 100% results.

3.4.2 Conversion tracking from the mailing

In the *Settings* folder under the menu item *Conversion Tracking*, you will find the general settings for the conversion tracking. To set a conversion cookie, and thus ensure that *Conversion Tracking* is useful, the tracking must be enabled here.

After each click within an eMailing, the user is provided with a cookie that indicates the mailing that was last visited per client. If a corresponding tracking pixel is installed in the shop, homepage or on any other pages of the eMailing marketing system, this information, such as the scope of the order, can be correlated with a mailing and evaluated there accordingly.

You can find the URL of the tracking pixel in the *Settings* folder under the menu item *Conversion Tracking* The following optional parameters can be attached to this URL:

`eid=<Zeichenkette>` This value is used for a unique assignment of the call to action. The order number or a separate customer number, for example, are possible as values If this value is transferred, the system will first try to find a data record for the corresponding eMailing. If this is successful, the other data there is overwritten, otherwise a new entry is created.

`i=<Zahl>` The integer transferred is saved as number.

`f=<Zahl>` The floating point number transferred is saved as value. Please note that the number must always be transferred in a machine-readable form. This means that the decimal place is separated by a ..

`s=<Zeichenkette>` Any character string that can contain the content of the shopping basket, for example.

The conversion data collected within an eMailing can be exported directly during the eMailing.

3.4.3 Target-group-specific web contents

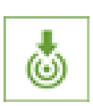
Objective: Use stored profile information to customize individual elements of an external website. For example

- Link customization (e.g., target-group-focused website links)
- Image customization (e.g., target-group-focused images)
- Download Customizations (e.g., download link with target group oriented PDF documents)
- CSS customization (e.g., use appropriate colors and type sizes contingent upon a person‘s age)

In this example, a customized start image is to be displayed on the homepage for the target group *Women* (*salutation* = Mrs).

Procedure:

1. Define and set up the desired target group (in this example, the target group *women*):

2. Create a SmartLink for this under *Web*:

3. Store the target-group-specific images on the respective website‘s web server or in this system.
4. Store an image for the default case. This image is returned if the system cannot detect an assignment to a target group.
5. Add an additional target. Set the *Females* target group and the URL to the image for females here.
6. Replace the URL of the current image on the website in HTML code by the integration URL of the SmartLink.
7. If when a link is called the system can establish a clear target group relationship, then it redirects the recipient to the configured link, otherwise the default link is used.

Important: Identification and profile detection based on cookies and thus does not return any 100% results.

3.4.4 Customized Statistics

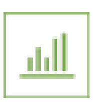
Objective: Create your own customized statistics and display them grouped together.

The system provides you various objects that you can combine in order to generate with their aid statistics as customized as possible. A reporting is a grouping of a wide variety of statistics that the system generates for the individual objects. Examples of such objects are:

- Pool
- Pool Data Miner
- Target group
- eMailing
- Checkpoint

Procedure:

1. Create a new *Report* in the *Reports* folder.



2. Below the *Configuration* tab, add all necessary objects whose statistics you want to include in the report by clicking on +.
3. You can view the evaluation in real time in the Preview.

3.4.5 Creation of sub-statistics using pool data miner

Objective: If a required evaluation is not provided in the standard statistics of the individual objects, you can create your own sub-statistics with the aid of the Data Miner. Using the pool data miner you can combine any pool fields with one another and modify across a target group. With it, the system provides an extensive composite variety of several thousands of statistics.

eMailing evaluations such as:

- Click rates allocated by country
- Opening rates according to gender

- Multiple click rates per fields of interest
- Hardbounces as per countries and branch offices
- etc.

General pool evaluations such as

- Genders allocated according to country narrowed down to client status
- Newsletter login according to interests
- Gender allocation according to purchase interests
- Subscriptions allocated by country
- etc.

Procedure:

1. Create a new Pool Data Miner in the *Reports* folder.

2. Configure an evaluation in the pool data miner.

- Attach + to the fields of the pool that the system should combine together.
- Define an optionally limiting target group.

3. Subsequently assign the pool data miner to a reporting. You can view an initial evaluation under *Preview* in *Statistics*.

3.5 Address management

3.5.1 Automatic data exports

Objective: Automatic and time-controlled forwarding of registration data via the *Query Management*.

Procedure:

1. Create a Query Management:

2. In the object; configure the *Report type* for *All* and specify all necessary details for *Fields* and *Report email* etc.
3. By *target group* and *pool* you determine which entries should be exported automatically. For this purpose create corresponding target groups in advance.
4. Deactivate the option *Report registration changes?*.
5. Select one of the specified *grouped transmissions* as transmission period.

Tip: In forms with high traffic, you should set the reporting to *daily* instead of *immediately*. Then all of a day‘s queries are sent grouped as a CVS or Excel file.

The *Grouping* function under *Report email* makes it possible to vary the distribution to recipients depending on the content of a pool field, such as *country*.

3.5.2 Import Addresses

A profile import can be performed under *Pool* -> *Import* tab. Thus, you can import profiles from a file or prepare existing profiles.

Important: Please note that there are fields that cannot be overwritten by the importer. This may result in profiles not being updated. These fields are: *CREATE_DATE*, *CHANGE_DATE*, *UNSUBSCRIBE_DATE*, *PROFILEID*, *PROFILEID*

Permission

Newly created profiles obtain the “Opt Out (confirmed)” permission by default.

Source selection

In this view, you have to specify which type of import source is to be used for the import. Here, you have a choice of the following import sources:

Excel (xls):

  This is necessary if XLS files (until Excel 2003) are to be imported.

Excel (xlsx):

  XLSX files (from Excel 2007)

CSV file:

  All types of CSV files. This should be the preferred choice for file uploads.

VCard file:

  VCard files. Here, it is possible to upload several VCard files compressed in a ZIP file.

Target group:

  Here, you can specify a previously configured target group. This is advisable if existing files of a target group are to be prepared.

Pool:

  Here, you can specify an existing pool. This is advisable if existing files of a pool are to be prepared.

Hint: All files can also be uploaded as a compressed file - e.g. .zip. When doing so, only one file (except for VCard) should be in the archive. By compressing the file you save valuable time during the import particularly with text formats such as CSV or HTML.

When you have selected the corresponding import source, several input fields appear:

- **Name:** Here you can define the name of the import.
- **Pool:** Here, you can specify the pool where the file is to be imported. This field only appears if more than one pool exists in the system and the importer is started from the pool overview.

- **Configuration:** Here you can specify an import template. An existing import can be marked in the import history as an import template.

Control view

The data to be imported is displayed in the monitoring view. The configuration options vary depending on the import source used.

- **File name:** Filename
- **Number of lines:** Number of lines of the source file
- **File encoding (CSV files):** Here, you select the character encoding of your file. Please check your choice visually to ensure whether all special characters displayed in the file view below are displayed correctly.
- **Separators (CSV files):** Please select here the separator used in your CSV file.
- **Text separator (CSV files):**: Please specify the text separator here. " is used mostly.
- **Preview N line:** Here you can specify the number of preview lines Default value: 20.

  Sonderverhalten bei XLS Dateien If a date format is detected in XLS files, this is converted automatically to the standard format DD.MM.YYYY HH:MM.

Configuration

This is the last and the most extensive configuration page of the import process.

The following basic configurations can be defined:

- **unique assignment by:** This is the primary key of the import, i.e. the data records to be imported are compared, for example, based on the "eMail" field to determine whether they already exist in the pool. If this is the case, the existing data records are updated. If the field is left empty, no check takes place, all profiles are created new - regardless of whether they already exist or not.
- **Duplicate Check by means of:** If a field is selected for the duplicate check, the import source is checked for duplicates by means of the duplicate check field. If duplicates occur, only the first data record of the duplicate is imported - all the rest will be ignored.

  Example (duplicate check by means of email):

  mustermann@example.com;Mr;Max;Sampleman <- is imported

  sampleman@example.com;Mrs;Maxine;Sampleman <- is ignored

  info@example.com;Mr;Max;Sampleman <- is imported
- **Add and update:** Existing data records detected by means of the primary key are updated, added new to the pool.
- **Only add:** New data records only are added to the pool. Existing data records are not updated and therefore ignored.
- **Only update:** Existing data records only are updated in the pool. New data records are ignored.
- **Use pool default values:** If this option is selected, new profiles are created with the defined pool default values (if no other corresponding value is imported).
- **Delete mode:** In the delete mode profiles can be deleted via the importer. There are three different modes for this:
  - **Deactivated**: The deletion mode is deactivated. No profiles are deleted during the import process.

  - **Only delete included profiles**: In this mode, all profiles are deleted that are included in the list as well as in the pool. When this mode is used, a primary key must be specified (clear assignment via)
  - **Remove and import non-included profiles**: In this mode, all profiles are deleted that are not included in the list. All the other data sets are imported.

The column configuration takes place below the basic configurations. In this part, the computer must be informed about which file column matches which pool field.

There are the following configuration options per line:

- **Arrow/warning sign:**
  - **green arrow**: The configuration of this column is OK.
  - **Warning**: Please check the configuration of this column. This does not mean that the configuration has to be incorrect.
  - **X**: This column is not used during the import due to a lack of configuration.
- **Delete Checkbox:** If this checkbox is activated and *Delete* is clicked at the lowest end of the configuration, the column is deleted from the configuration and therefore not used during the import.
- **Column (file) dropdown:** In this dropdown, all columns of your import source are displayed.
- **Field (pool) dropdown:** In this dropdown, all columns of the pool are displayed. If no pool field is selected, the configuration line is invalid and consequently this column is ignored during the import.
- **Advanced settings:** This function will be explained in more detail later.

Function "New line"

Below the column configuration is the *New line* button. This function allows you to generate virtual columns among other things. This is useful if the data records to be imported are to be marked explicitly without changing the import source. If you need to create a virtual column, please proceed as follows:

- Leave the dropdown for the columns of the import file empty,
- select a pool field,
- select the *own value* function in the *Extended settings*,
- select/enter the value to be set for all data records to be imported.

Extended settings

Certain extended settings are available in every configuration line. This may vary depending on the data type.

- **Overwrite:** If this function is activated, then empty fields within this column will also be included from your import source. In the case of multiple selection fields, existing selections of existing profiles are not supplemented but overwritten.

  Example (multiple selection field target group with values A,B,C,D,E):

  - Profile data record prior to import: Max Sampleman, target group: A,B,C
  - Profile data in import source: Max Sampleman, target group: D,E
  - Profile data after import, without overwrite function: Max Sampleman, target group: A,B,C,D,E (data was supplemented)
  - Profile data after import, with overwrite function: Max Sampleman, target group: D,E

- **own value:** If this function is activated, the selected value and not the value of the import source is used for this column. This function can also be used for creating virtual columns in the import source.
- **Remove attribute option:** This function is only displayed with multiple selection fields. This function allows the values for selected profile data to be removed from the selection in the case of multiple selections.

Format assignment within the column configuration

Different type configurations are available in the column configuration depending on the type of pool field selected. There are four different type configurations:

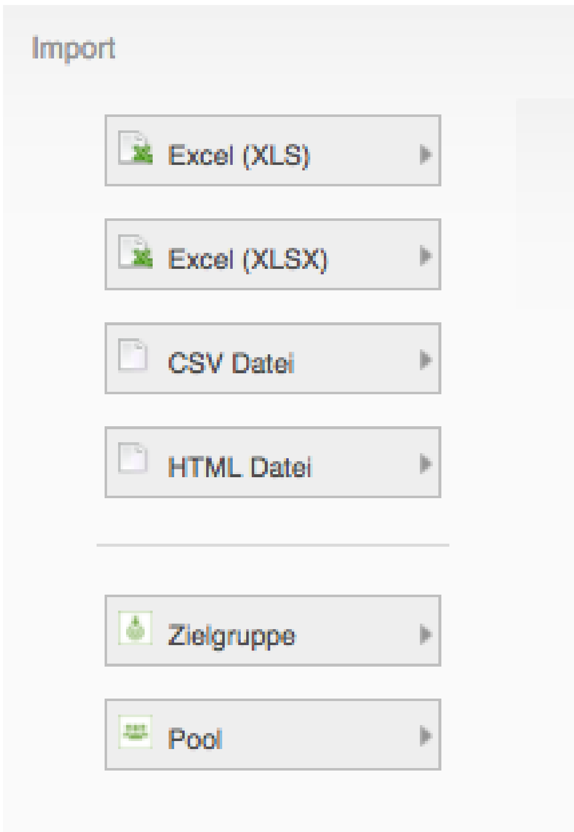
- **No type configuration:** If this is a pool field, then no type configuration is available.
- **If options for a single or multiple selection field are used in an import source that have not yet been configured for this field in the pool, the importer displays an „option mapping". Here you have the possibility to link non-existing options to an existing option or to create them accordingly. This type configuration offers the following configuration options:
  - **Separator: The separator must be specified here if multiple options are specified in a field. Example: a,b,c contains the separator , .
  - **Missing attribute option**: The option that does not yet exist is displayed here.
  - **Available attribute options**: All available attribute options of the pool field are displayed here. If the missing attribute option is to be replaced by an existing attribute option, the correct value must be selected here.
  - **Create attribute option**: If the missing attribute option is not to be replaced with an existing attribute option, the corresponding option can be created by activating this checkbox. This field is not available for system fields.
- **Date configuration:** If the selected pool field is a date field, the following configurations are available:
  - **Format**: Date format of your import source. Attention with Excel files: If a field is recognized as a date field, the date is automatically converted to the format DD.HH.YYYY HH:MM.
  - **Blank format: If there are empty data records within a date column, the format for empty date values must be specified here, e.g. „-„.
  - **Use Regex**: Use a regular expression.
  - **Regex**: If *Use Regex*, the regular expression must be specified here.
  - **Data with time**: If a regular expression is used, it must be specified whether the date values are provided with a time.
  - **Two-digit year**: If you use a regular expression, you must specify whether the year is only two digits in the date values.
- **True / False configuration: If the selected pool field is a *true/false* field, the following configuration is available:
  - **‘‘True‘ value**: Here you have to specify the values for which TRUE is to be set in the corresponding pool field of the profile to be imported. Several attributes can be separated with |. Default value: yes|true|yes|1.
- **Email configuration:** If the selected pool field is an email field, the following configuration is available:
  - **Only valid email addresses**: If this checkbox is activated, only profiles which can show a valid eMail address will be taken into account when importing within this source column.

3.5.3 Importing Addresses (Enhanced)

This How-To provides various practical examples with regard to the profile importer.

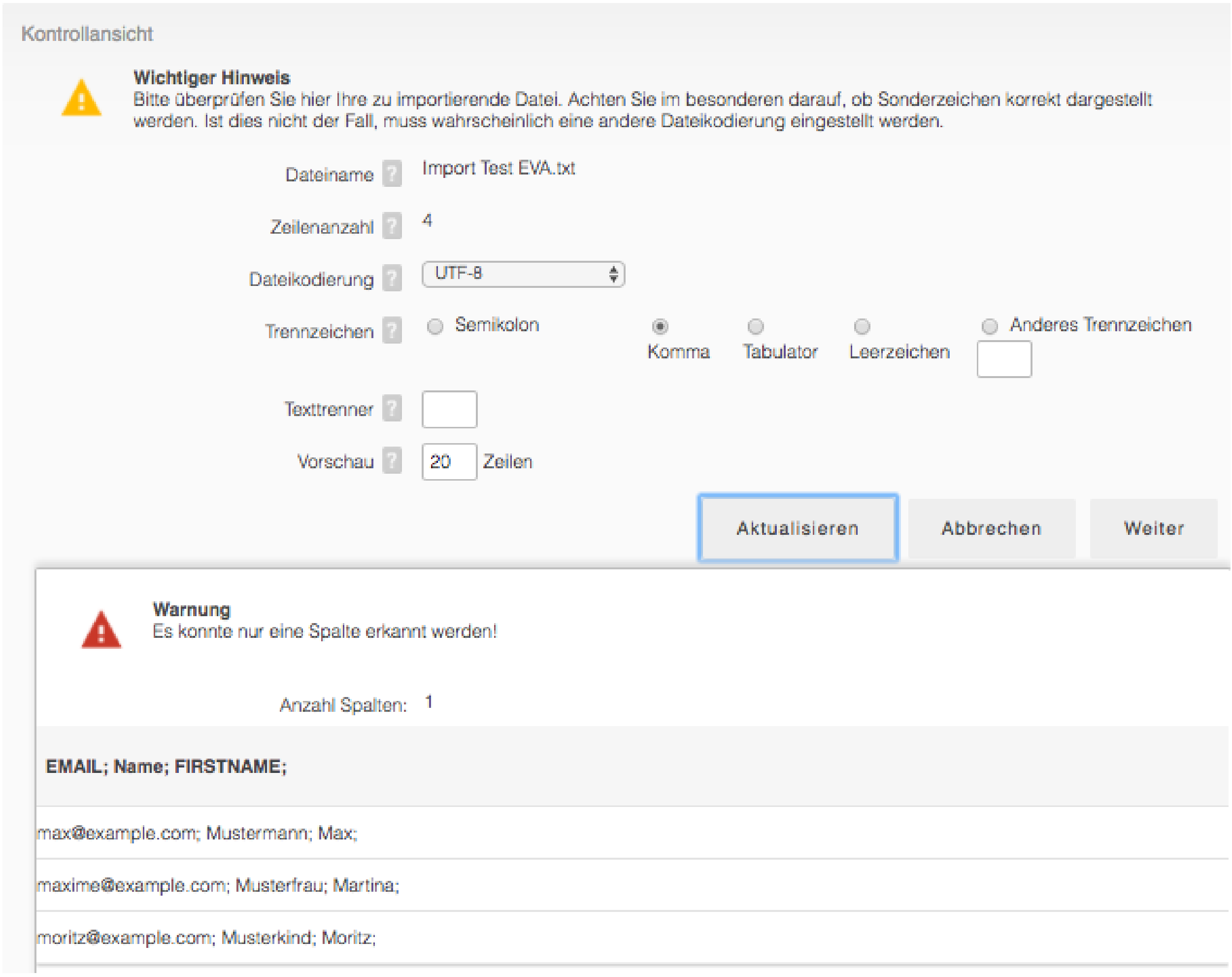
Importing a CSV file

Open the import interface by selecting the *Import* tab within your pool. Now select “File” and your CSV file as import source.



Please configure all the remaining basic settings within this configuration page and click on *Continue* to confirm your entry.

Settings within the control view:



Make all the basic configuration settings here regarding the CSV file. Please refer to the profile import documentation for more details (Category: Control view). Check your settings by clicking on *Update*.

Observe the following checks: *Are all special characters still available? If this is not the case, an incorrect file encoding was probably selected.*

Were all columns recognized correctly, or does the message "Only one column could be recognized! " appear even though the CSV file contains more than one column?

In this example, the control view should look like the following:

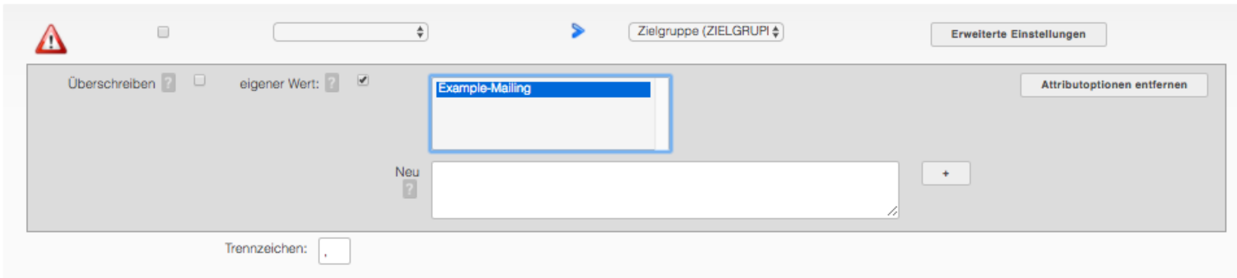
EMAIL	Name	FIRSTNAME
max@example.com	Mustermann	Max
maxime@example.com	Musterfrau	Martina
moritz@example.com	Musterkind	Moritz

You can continue as usual with the import process by clicking on *Continue*.

Selecting the profiles to be imported for a target group

Profiles can be marked explicitly with the profile importer in order to combine them later in a target group. This makes sense if you only want to import profiles for a mailing. For this purpose, a new pool field must be created if it has not already been done. In this example, the pool field “Targetgroup” is used as a multiple selection field with the option”Example-Mailing”.

Start the profile importer and continue until the column configuration view. When you have reached this, create a new column configuration line by clicking on *New line*.



The field “Targetgroup” is now selected as pool field. A file column must not be selected since this not exist in the original file. Now, the *own value* function must be activated in the *Extended settings* and the corresponding options value selected.

The import is started as usual by clicking on *Import* - and all profiles from the import source are marked with the selected option.

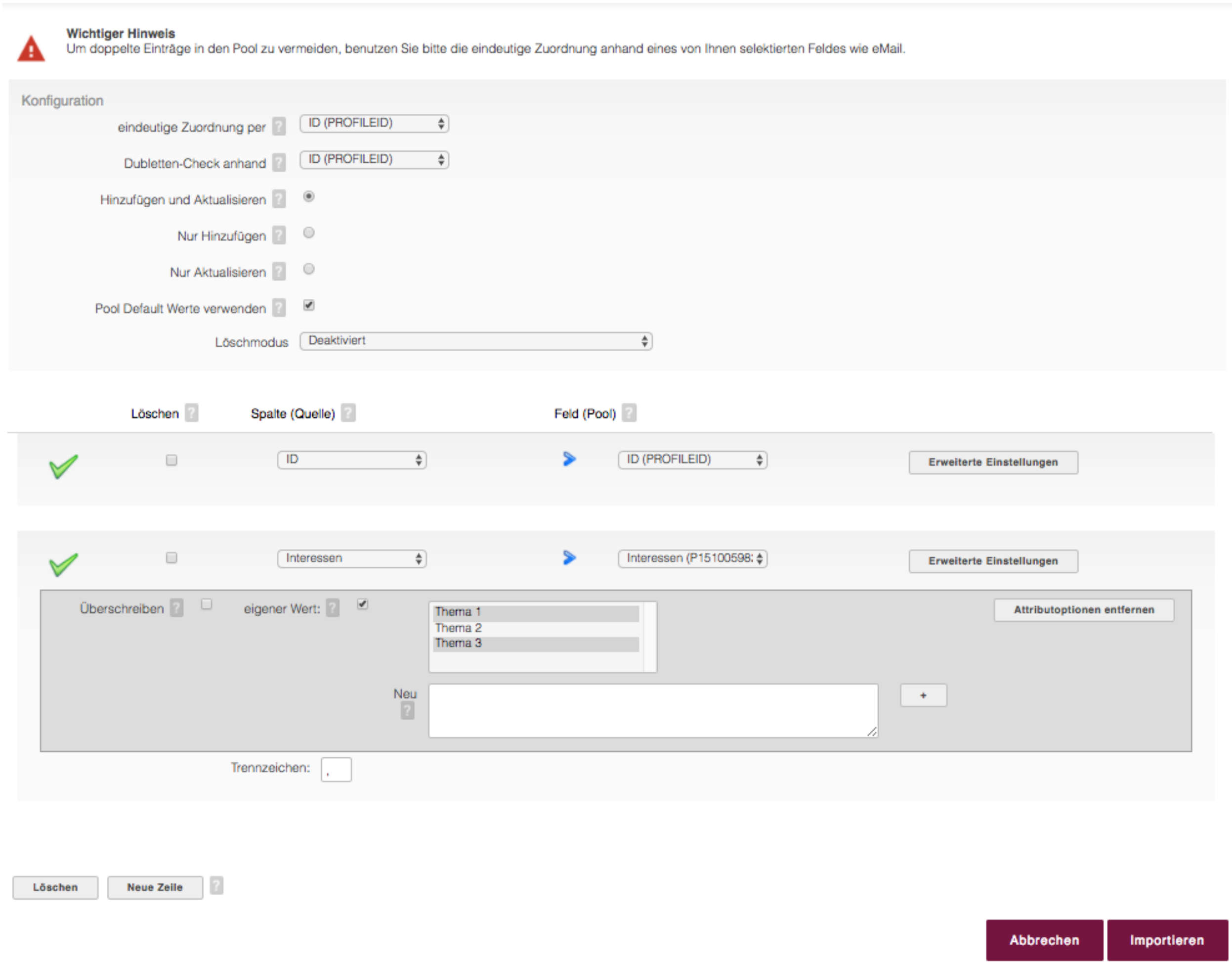
Removal of marked field options in multiple selection fields

Example: There are lots of profiles that have the following options marked in a multiple selection field “Option field”: A,B,C,D. The option B and D are to be deselected from these profiles, however.

The first step is to create a target group containing the profiles to be updated. After that, you can change to the importer interface as usual.

Now select the corresponding target group as the import source, configure the other basic settings at your discretion and continue until the column configuration.

After reaching the column configuration, select the pool field _ID as ‘‘unique assignment by‘‘ and ‘‘duplicate check by means‘‘ and delete all column configurations except for *ID_ as well as the pool field *option field* to be edited in our case.



To deselect the options, click on the *Extended settings* and there on *remove attribute options*. You can deselect all options that should be deselected by simply clicking on the red *X*. Afterwards, the import only needs to be started by clicking on *Import*.

3.5.4 Export Addresses

Each list view of profiles - e.g. in the pool, in a target group, in the eMailing statistics - can be exported as a file. You can find the corresponding function in the relevant list view at the top right. To do this, select the format required in the dropdown menu as well as the language in the info window and then click on the *Export* button.

The following formats are possible:

CSV (comma separated values): List as UTF-8 text with separators

Excel (XLS): Microsoft Excel file

Excel (XLS/HTML): HTML file for using in Excel

Excel XLSX: Office 2010 XML format

HTML: HTML table (can also be used directly with Excel or Open Office)

Google Earth (KML): Geo-coordinates for Google Earth

SQL: SQL insert statements

XML: Simple XML structure with each profile as node and its attributes as elements in the text

The language refers to the standard fields and contents. Your own fields are not translated here and can thus lead to confusion. You generally achieve the best results by using the same language with which you also created on the pool.

In the case of large amounts of data, it is advisable to make the data *to split up*. When doing so, as many files as necessary are created and delivered in the export file so that the set number of profiles is not exceeded per file. This allows you, for example, to split up lists into 65k profiles, which represents the maximum number of lines displayed in some spreadsheet programs.

An export is always sent to the email address belonging to the user account that initiated the export. After a short while, you will receive an email from the system to the email address that was set in the user with a link for downloading the export file. This link only works for a few days and is deleted automatically by the system. Please note that the export might take some time depending on the file format and target group/pool configuration. We generally recommend using the CSV format preferably for the Excel export in order to avoid long waiting times.

Important: Please note that the export of profiles requires corresponding rights. Furthermore, only one export can ever be requested from one object. If an export is already in progress, then try once again at a later time.

3.5.5 Blacklist management

Objective: Blacklisting of email addresses

Procedure:

1. Click on the *New Entry* button and enter the email address. Alternatively, you can also exclude a complete domain from the transmission here (entry: @domain.de). Optionally, you can save a comment to indicate later why this entry was created. After entering the email address, click on *Save*.
2. Der Blacklisteintrag ist ab sofort wirksam - reguläre eMailings werden an diese Adresse(n) nicht mehr verschickt. Es sind jedoch weiterhin Eintragungen in den Pool möglich woraufhin z.B. eine Permissionemail an die Adresse verschickt wird. Des Weiteren kann die Nutzung der Blacklist in jeder Vorlage deaktiviert werden.
3. In the transmission statistics you can check how many entries from your pool were blocked during each transmission (and thus not sent).

You can delete blacklist entries again at any time. After deletion, the subsequent transmissions will be delivered regularly again to the address(es).

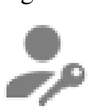
3.6 Administration

3.6.1 Users and rights

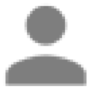
Objective: Creation of a new user with individual rights.

Procedure:

1. In the left folder navigation below the *Settings* folder you can find the items *Users* and *Roles*. Here, you can create new access rights to the system (users) and define which elements of the system they have access to (roles). By clicking on *Roles* you create a new role - e.g. "Editor". You define via roles a group of individual rights from a full list.

2. To assign certain kinds of rights to a role, activate the checkbox at the required rights. In addition, the *Login* right must be activated, since otherwise the owner of this role will not be able to log into the system.
3. To create a new user, enter the access data and assign him one or more of the existing roles.



4. If several roles are assigned to a user, then the sum total of all rights relating to these roles will always apply to him.

Important: Immediately delete users who are no longer authorized. Always assign only as many rights to all authorized users as are absolutely necessary.

3.6.2 Backup

 The Backup Center allows you to recover deleted profiles independently for the most part. The Backup Center can be called up via the menu item *Profile* -> *Backup* Tab.

Which data / actions are backed up automatically?

- **Profile imports**: If a profile import is carried out, a backup of the the entire pool affected is created before the actual import.
- **Profile deletions via target groups**: Before profiles are deleted via a target group, a backup of the affected profiles is created (the entire pool is not backed up here)
- **Recovery of backups**: Before a backup is recovered, all affected profiles are saved again (the entire pool is not backed up here)

Profile deletions via a profile list are not taken into account at present.

Manual backup

It is possible to create a maximum of three manual backups of your pools. Since most deletions / data manipulations are backed up by the Backup Center automatically or there is always a backup of your data from the previous day, creating manual backups regularly is only useful to a limited extent. If the limit of maximum manual backups has already been reached, an old backup must be deleted via the list.

Recovery / Retention period

It is always possible to recover all backups autonomously which were created automatically. Manual backups can only be recovered by our support team. The retention period for automatic backups is 14 days or 90 days for manual backups.

Important: During the recovery of a backup, all profiles that existed in the pool at the time of creating the backup are set to the old status. New profiles that have been imported in the meantime or have been registered via forms are not deleted. If these profiles should be deleted, please first empty the pool completely and then import the backup. Thus, you preserve the pool at the time of the backup.

Deleting backups

Existing backups (whether manual or automatic backups) can be deleted in the Backup Center. If manual backups are deleted, the credit for manual backups will be increased again.

Important: The deletion of a backup is final. The backup cannot be recovered again.

3.6.3 Locking object

An object lock directly influences the usability of the object. The purpose of the object lock is to prevent several users from changing an object at the same time and thus overwriting each others‘ changes. Furthermore, you can protect objects against changes by unauthorized persons by means of a hard lock.

Objects can be locked and unlocked again in the system in two ways - either via the corresponding object function on the object itself or by means of the stylized cogwheel in the list view.

Which lock types are there?

- **Soft block**: A soft block can be created by any user of the system and primarily serves as a hurdle. It is to keep a user from directly editing the object. The block must be removed first before the object can be edited. Every user can remove this block.
- **Hard block**: In order to create/remove hard blocks, the user must have the corresponding right - in this way, other users cannot edit blocked objects.

Furthermore, locked objects can neither be deleted nor changed in any other way. It is also not possible to transmit locked eMailings. None of this applies to the initiator of the lock, however. He can still edit or use the objects without restrictions and regardless of the lock type.

Visualization of the lock

An object lock is visualized by means of a colored lock that is superimposed onto the corresponding object icon in the list view. A yellow lock means that the current user can unlock or edit the object. A red lock, however, means that the current user cannot edit the object.

3.6.4 Locking of external access

Another type of lock is the locking of external access This is a function that an object - such as a form - no longer displays if the URL is activated by a user who is no longer in the system, but merely delivers a white page or routes the caller to the URL set during the lock.

The list view visualizes whether an object is locked for external access by means of a crossed-out stylized globe. The function itself can either by accessed from the object itself of via the stylized cogwheel in the list view.

If the object for which external access is locked is copied, the new object will be locked, too.

3.6.5 Global data

Global data are objects that can be made accessible to all clients via a shared folder at group level. This can be templates, images as well as articles that the corresponding clients can integrate in their articles, eMailings etc. Users created at client level have read only access to global data - the data cannot be changed.

Provision of global data

To make an object available globally, it must be saved in the *Global data* folder at group level (either directly or in a subfolder).

Use of global data

The global data is shown in two places on the mandator level:

- **Navlet**: As a Navlet in the navigation containing a complete tree view of the folder
- **Selection picker**: As a tab in the selection window of the object selection picker

The user must have the corresponding right at client level to be able to see this selection option - users without this right cannot access global data (this is particularly important if this right is taken from the user at any later stage).

Global objects can be copied by the users in their clients and customized at any time.

3.6.6 Client configuration

General

In the *Account* folder below *Settings*, you can view, and in some cases, edit the basic data of the client such as name etc. (provided that you have the corresponding right).

Various system parameters can also be viewed here:

- **Price model**: The price model set for the mandator
- **Industry**: The industry (if any) employed by the client which is used for industry comparison. The selected industry sector is only used for active clients. Demo clients use the industry sector „Other“, regardless of the configuration.
- **Data storage**: Whether anonymous or personal tracking is set for the mandator
- **Size**: The web space used by the mandator in MB

Sample Environment

The current pilot environment can be regenerated here. During this process, all objects and their links in the selected language are created again - data that already exists is not overwritten.

3.6.7 Copying clients

The copy function for clients is an extensive tool for creating a complete image of a client. During the copy process, all dependencies on objects (if these are mapped in the client you are copying) are cancelled and changed over to the new copied objects.

Ranges of applications

The following ranges of applications are possible:

- Creation of your own pilot environment
- Backup of complete clients (without profiles)

- Mapping of fixed initial client structures
- etc.

Supported objects:

The following objects are supported:

- Target groups
- Forms
- eMailings
- Cron eMailings
- Split campaigns
- Images
- QR Codes
- Article templates (all types)
- Emailing templates
- Reports
- Web pages
- Container
- Article
- LeadPage
- Query manager
- Notes
- Files

Functionality

Unlike the normal copying process within the system, all dependencies on the objects are transferred to the new copied objects during copying (if they are copied, too). This means that almost no manual subsequent corrections are necessary.

Activation of a copying process

The function for copying clients can be found in the global client overview. In the middle of the screen are client elements each provided with a cogwheel for the configuration. Click on the cogwheel and then on *Duplicate* to start the configuration interface of the copying process. After configuration and clicking on *Copy client*, a process that copies the complete client starts in the background. After completion, an email with a confirmation is sent to the initiator.

3.6.8 Favorites

You can mark frequently used objects as bookmark. This allows you to access the object directly with one click. To mark the object, open the corresponding object and click on the yellow star on the object functions at the top right. Alternatively, you can - if the object is displayed there - click on the star for the corresponding object in the list of the last objects (*Object history*)

The favorites are saved individually for each user and are preserved after a client change or logout.

Menu item

Favorites are displayed in the *Favorites* menu item and can be re-sorted via drag & drop. By clicking on the favorite the system skips immediately to the selected object.

Favorites in other clients are symbolized by their own star icon.

Deleting favorites

To delete a favorite, just simply click on the yellow star. This is possible within the menu item and in the object itself in the object functions. In doing so, only the link to the object will be deleted, not the object itself.

3.6.9 Object history

Below the menu item *Favorites* the objects last opened are displayed in descending order after the time of opening. You can access this object directly by clicking on the object. Only object from the current client are displayed here. The data is preserved even after a logout.

Objects opened within the last hour are marked in bold

The number of objects that should appear in this list can be predefined in the user settings. This function can be completely deactivated here.

Favorites

Objects from the history can be added to your own favorites by clicking on the star. Alternatively, you can move an object to the favorite area by drag & drop. At the same time, it is marked as favorite automatically and added to the list.

3.6.10 WebDAV

The WebDAV interface allows you to work even more efficiently with certain object types. The tree and object structure are provided as a virtual file system via WebDAV.

Supported Clients

The following clients are supported:

- Adobe Dreamweaver CS5 (Windows, Mac)
- Total Commander ** (with corresponding plug-in) (Windows)
- Cyberduck (Windows, Mac)

- Interarchy ** (Mac)
- Panic Transfer ** (Mac)
- cadaver (Linux)

** The creation of files is very restricted.

Supported object types

The following object types are supported:

Object type	Read / Write	Creation
Images	Yes	Yes
Newsletter template	Yes	No
Files	Yes	Yes
Website	Yes	No
Article template PDF	Yes	No
Article template Text	Yes	No
Article template HTML	Yes	No
Article template Web	Yes	No
Form (individual template)	Yes	No

Configuration of the client

All relevant configuration settings for your WebDAV client can be found under "Settings" -> "WebDAV".

Working with WebDAV

After successful connection, the familiar folder structure of the system is available to you. Here, you can navigate through the objects and edit them as usual.

Neuanlage von Objekten Two object types can be created per WebDAV: Images and files. If an image file is identified, an image object is generated automatically. Alternatively, the file object is selected.

Bearbeiten von Objekten All objects can be edited directly like in a local file system. All changes have a direct effect on the system.

3.6.11 Desk

Simple display

The `Simple display` is a simple interface enabling the user quick access to the most frequently used system folders and for some of these folders also provides the possibility of creating a new object in this folder via a link.

Enhanced display

The `Enhanced display` is a fully configurable desk The contents are organized by means of a widget in columns and can be freely arranged both horizontally and vertically. At the same time, individual widgets can be hidden if required.

System widgets

Widgets such as the system overview or the list views of individual objects (last transmissions etc) are so-called system widgets. These can be hidden but cannot be deleted and they can be configured only to a limited extent. The number of objects displayed by default as well as the number of objects that can be reloaded by means of the `more` button can be preset.

Object widgets

Some system objects can be integrated on the desk as dynamic object widgets. These widgets display core data of the statistics of the corresponding objects (such as the progress of a target group including the number of profiles). These widgets can be feely created or hidden and can also be deleted completely from the desk.

CHAPTER 4

FAQ für Anwender

4.1 eMailing

4.1.1 There are many profiles in my target group. If I want to start a transmission, this is blocked since there are no profiles in the target group!

Only “dispatchable” profiles can be used in a transmission. See **‘Why does the number of dispatchable addresses vary from the number of registrations?‘_**

4.1.2 After I clicked on *Start shipping*, the login window appears and a message that I don’t have the required rights?!

This is because your user account does not have the `eMailing: show` or `eMailing (queue): show` permission. After sending, you will be redirected to an object of the appropriate type. The dispatch was started anyway.

4.1.3 The links in the text eMailing are not tracked!

Tracking of links in text eMailings is not possible at present since perfect detection of the links used is not ensured.

4.1.4 Is there any information on the function of the junk filter in Outlook?

Microsoft has published a list of keywords that are attacked by the built-in junk filter. This list can be found at the following URL https://support.office.com/de-DE/article/Informationen-zum-Schutz-vor-Junk-E-Mail-8d0223ec-1457-46bd-85c3-81e023f52c6d .

4.1.5 How does article sharing work in social networks?

All you need to do is insert the appropriate variables in the article template.

Example: `<a href="{$SOCIAL_TWITTER_URL}">Twitter Share</a>`

You can find the variables in the area `Personalization variables` of the service center in the sub-area `Articles`

4.1.6 When opening the online view of a mailing or landing page, the wrong favicon is displayed!

First upload the file *favicon.ico* as a file (not as an image!) to the system.

Afterwards, enter the line `<link rel="shortcut icon" href="<url_auf_favicon>" type="image/x-icon" />` in the head of the templates email (HTML), landing page and web (if used). Replace `<url_auf_favicon>` simply with the URL, which you can find in the Integration tab of the uploaded file.

4.1.7 When calling the visual editor of a mailing (Contents tab), an error message appears

```
Warning: DOMDocument::loadXML() [function.DOMDocument-loadXML]:
Unescaped '<' not allowed in attributes values in Entity, line: xxx
```

The error occurs if replacement variables (`{$INPUTFIELD_n}`, `{$TEXTAREA_n}` and `{$HTMLAREA_n}`) are noted in HTML attributes. Then, for example, the code `<a href="{$INPUTFIELD_0}">Link</a>` results in the error message mentioned above.

The reason for this is that the aforementioned variables for HTML input elements (<input..., <textarea, etc.) are resolved in the visual editor to enable an input. As a result, the link would then read `<a href="<input type=...">Link</a>`, which represents invalid HTML.

All input fields, text and HTML areas can be output by means of a simple TAL call.

- for input fields this is `${THIS/getInputFields/n}`
- for text areas this is `${THIS/getTextAreas/n}`
- for HTML areas this is `${THIS/getHtmlAreas/n}`

n must be replaced by the number of the field in each case. The example link above would then read as follows:

```
<a href="${THIS/getInputFields/0}">Link</a>
```

Since it is no longer possible to enter the value of the input field in the visual editor now, the template must be supplemented by the following instruction:

```
<div tal:condition="THIS/isAppendMode">Link target:  {$INPUTFIELD_0}</div>
```

4.2 Images

4.2.1 Why is it not possible to upload any other image types except for png, jpeg und gif?

Due to reasons of compatibility for the display in different types of media, we have decided only to support the most common image formats. If you want to upload high resolution images in good quality, then we recommend using the png format.

4.2.2 When I open a picture, the whole display in the browser (Firefox) is "shifted"!

This is a problem of the browser which occurs in conjunction with certain plugins (it is currently known as "Cool Iris"). Please contact the support in this case.

4.3 Profiles

4.3.1 Why do I only have one pool for my addresses?

The segmentation of address databases is organized via target groups since this works more easily and efficiently than via different pools.

Furthermore, problems arise in connection with forms - forms are permanently linked to a pool. A recipient from pool A cannot access a form linked to pool B - this results in an error message for the recipient.

This would have a particularly critical impact on test target groups/transmissions: The members of the test target group are in pool A, the actual recipients in pool B. The form works during the test transmission/among the test recipients, but an error message then appears during the real transmission.

If overlaps of profiles occur within the pools (a recipient exists in several pools), it is neither possible to guarantee the functionality of the bounce management nor of the unsubscriptions. Bounces and unsubscriptions are stored pool-based. If a recipient from pool A withdraws his permission, this information will only be stored in pool A but not in pool B. A transmission to this profile in pool B would still be possible.

A simultaneous transmission to profiles of two pools is also not possible even though target groups can be formed across several pools.

It makes sense to work with several pools if addresses are to be generated via forms (e.g. ongoing competitions, ongoing surveys) whose quality cannot be foreseen. This, for example, can prevent the database of regular newsletter recipients from being impaired.

4.4 Target groups

4.4.1 Why does the number of dispatchable addresses vary from the number of registrations?

Loadable profiles are profiles that fulfill all validity criteria. This means that the profiles have an e-mail address, are not faulty, are not deleted and have a valid permission.

4.4.2 How can I select profiles by zipcode areas?

Select zipcode in the configuration of the ZIPCODE field, select “includes” and enter the zipcode area as selection criterion followed by underscore (_).

An example (with five-digit postcode): If you want to select the postcode area 83, enter the following: 83___

4.5 Form

4.5.1 Many profiles can be registered in a form. In the target group with the setting `form = ...`, these do not appear, however?

An assignment of a profile takes place only once with the first entry. Presumably the profiles were imported and an editing of an existing profile or an entry in another form will not be saved as an entry by the form.

4.6 Inquiry-Manager

4.6.1 My test queries are not registered by the Query Manager!

- Check whether the target group set in the Query Manager displays your profile (Tab *Profile List*)?
- Is the Manager set so that it also registers changes to existing profiles?
- Have you set a valid sender for the email; does the sender have a different domain than the recipients?
- Has the time for the delivery been selected correctly? Only query emails set to `Immediate` are sent within a few minutes after the query. All other queries are only sent once a night.

4.7 Web

4.7.1 How can I upload a PDF file and link it in an article?

To do this, create a File‘ object in the ‘Web‘ folder and upload the PDF. You can then use the ‘‘URL to this resource as a link for your article. Please note that the filename of the PDF (and of every other file) is preserved and is displayed to anyone who downloads the file.

4.8 Templates

4.8.1 After making changes to an eMailing template, these are not included in an existing eMailing!

When creating a new eMailing, the template is copied. To transfer changes in the template to the eMailing, select the *Copy template* entry from the drop-down menu of the corresponding template and click on the eMailing concerned.

4.8.2 Clicks on the table of contents do not jump to the article!

Make sure that the jump target is generated for the article template in the eMailing. Preferably around the headline with the following code:

```
<a name="atruntime" tal:attributes="name THIS/getAnchor">{$HEADLINE}</a>
```

4.8.3 Can individual objects be ignored when reading aloud?

You can ignore HTML tags (and their child tags) by adding the CSS class ISI_IGNORE to the tag. <div class="ISI_IGNORE">I am not read</div>

4.8.4 Is it possible to define the place where the reading should start/finish?

By means of the HTML comments <!--ISI_LISTEN_START--> and <!--ISI_LISTEN_STOP--> you can specify which section you want / do not want to have read aloud.

4.8.5 Chinese/Japanese fonts are not represented in the PDF!

The Sans-Serif font of the templates does not contain these characters. Set the font in the FO template (s. `fo:root`) to `font-family="UKaiCN"`. Also see next question.

4.8.6 Chinese/Japanese text in the PDF is not wrapped. Only the first line of a paragraph appears and this runs off the document.

Extend the template in the `fo:root` tag by the following attributes `hyphenate="true" language="zh"` for Chinese characters, by the following attributes `hyphenate="true" language="yes"` for Japanese characters.

4.9 Geotracking

4.9.1 Why does a new profile not appear on the map?

In order to geocode a profile, it is necessary that the profile has a correct address (postcode and/or city and street) and that this city is located in Europe. If these requirements are met, a profile can be encoded. Depending on the system load, this can take a few minutes.

4.9.2 Why does the number of the profiles on the map vary from the number of profiles in the pool?

Profiles that do not have a usable address or have an address outside Europe cannot be encoded.

4.9.3 All profiles have correct address data but they do not appear on the map even after a long waiting period.

The geocoding might take some time depending on the system load. If the profiles still do not appear even after few hours, please contact the support.

4.9.4 Why does the display of the profiles on the map last very long?

A complex calculation is necessary for displaying the grouped view. The larger the map section and the higher the number of profiles affected, the longer it can last before the map is retrieved. Normally, this should not last longer than 5 seconds, however.

4.10 Misc

4.10.1 After I've edited an article/email, strange code (`function __RP_Callback_Helper`) appears in my text!

This is the *Download & Recording* function of the *Real Player* software. This function, which is nothing more than a browser plug-in, inserts this code into each text input field. This behavior can disabled in the Real Player settings in the menu item *Download & Recording*.

4.10.2 I have been looking for an object with a certain ID/URL but I do not know in which client it is located.

You can search across all clients at global level using the search for ID/URL. In the results, you will then be shown in which clients the searched objects are located.

CHAPTER 5

Developer area

The designer area is intended for experienced users of the system who want to deal directly with the implementation of scenarios themselves. Here, you will find all key information on using the template language with many practical applications, general information for making the designer‘s life easier as well as the detailed list documentation of the web service interface.

5.1 Webservices

5.1.1 APIs

The system offers various interfaces via web service for integration into third-party systems.

The primary interface with read and write access is the SOAP API. You can use it to query and change data object-specifically. The range of functions is based on the functions of the system, so that remote control of the system is possible to a limited extent.

Using the Reporting API, data can be retrieved in its original format, for example for analysis and dashboard tools. Write access is not possible here.

Interface	Version	Status	Description
SOAP V0	0	**Active**	The SOAP API is used to connect to third-party systems such as CRM or CMS systems. The API offers read and write access.
SOAP V1	0	**Active**	Die SOAP v1 API erweitert die standard SOAP API um weitere Funktionen.
Reporting API	1	**Active**	The Reporting API is used to connect to third-party systems that offer extended data analysis, for example. The API provides read-only access.
XML-RPC-API	1	Discontinued (security updates only)	The XML-RPC-API provides read and write access, but will not be further developed. Only bugs will be fixed and security updates will be done.

To avoid compatibility problems with updates to the existing SOAP API, this interface is versioned. A prompt migration to the latest version is recommended.

SOAP V0

Important information

Please note that global users who are intended to use the SOAP-API or are already used for this purpose, should not be used for working in the system interface.

If this is the case, however, you run the risk of making the global user a client user by changing to a client via the system interface, as long as this user changes again to the "Global Overview" via the system interface.

This can have repercussions on all methods that need the parameter `mandator_id`. In this case, this global user that changed to a client would ignore the parameter `mandator_id` and would use the ID of the client that he was changed to.

Restrictions

Please note that an individual request via the API must not be more than 10MB in size. This is a security policy. When transferring larger requests, these must be split up.

Email Validation

When email addresses are entered, a validation takes place that prevents invalid email addresses from being entered into a pool. The following criteria are checked:

- E-mail syntax correct
- E-mail not available in blacklist
- Domain has valid A-Record
- Domain has valid MX record

General use EvalancheHashMaps are used wherever lists of key value pairs are needed; however, the keys can only be first determined at runtime.

Since the property of the individual EvalancheHashMapItem is necessarily of the type `string`, the following rules apply to deviating data types:

- "Multiline entry (formatted)": Valid XHTML with optional root-node or `Foo & Bar` or `<div>Foo & Bar</div>`. **Invalid XHTML can cause the article to no longer be editable via the interface. It might not be possible to send mailings containing such an article anymore!**
- "Multiline entry (HTML-Editor)": Valid XHTML with optional root-node or `Foo & Bar` or `<div>Foo & Bar</div>`. **Invalid XHTML can cause the article to no longer be editable via the interface. It might not be possible to send mailings containing such an article anymore!**
- "Integer": Is sent as a string-representation (in PHP via `strval($integer_value)`
- "True/False": `True` is transmitted as string with the content‘‘1‘‘, `False` is transmitted as blank string
- Single selections: The option idea of the value is transferred as a string representation
- "Date" and "Date with time": Unix timestamp as string representation

Use with articles and containers

- Multiple selections: The list of selected option IDs as a list of IDs separated by "|", e.g. 123|456|789
- "Links (list)": List of objects with the properties `text` and `url` JSON serialized, e.g. [{"text":"DuckDuckGo","url":"https://ddg.gg"},{"text":"Google","url":"http://www.google.com"}]
- "Values (list)": List of objects with the properties `key` and `value` JSON serialized, e.g. [{"key":"foo","value":"bar"},{"key":"herp","value":"derp"}]
- "Image (list)": The list of selected image Ids as JSON serialized list, e.g. ("[1233,1234,1235]")
- "Container (list)":The list of selected container Ids as JSON serialized list, e.g. ("[1233,1234,1235]")

Use in profiles

- Multiple selections: The list of selected option IDs as a string serialized with `|` ("pipe"), e.g. (`"|1233|1234|1235|"`)
- Single selections: The selected option ID is also serialized as a string with `|` ("pipe"), e.g. (`"|1233|"`)
- UID: The encrypted ID of a profile can be queried via the virtual attribute "SID". Methods that return all profile data automatically have the virtual attribute "SID".

WSDL

Information on access points and WSDL files is found in the following table:

Port	WSDL Document/literal	WSDL RPC/encoded
Article	/soap.php/wsdl/article	/soap.php/wsdlcompat/article
Article template	/soap.php/wsdl/articletemplate	/soap.php/wsdlcompat/articletemplate
Articletype	/soap.php/wsdl/articletype	/soap.php/wsdlcompat/articletype
Blacklist	/soap.php/wsdl/blacklist	/soap.php/wsdlcompat/blacklist
Campaign	/soap.php/wsdl/campaign	/soap.php/wsdlcompat/campaign
Category	/soap.php/wsdl/category	/soap.php/wsdlcompat/category
Container	/soap.php/wsdl/container	/soap.php/wsdlcompat/container
Containertype	/soap.php/wsdl/containertype	/soap.php/wsdlcompat/containertype
Document	/soap.php/wsdl/document	/soap.php/wsdlcompat/document
Form	/soap.php/wsdl/form	/soap.php/wsdlcompat/form
Image	/soap.php/wsdl/image	/soap.php/wsdlcompat/image
Emailing	/soap.php/wsdl/mailing	/soap.php/wsdlcompat/mailing
Mailing template	/soap.php/wsdl/mailingtemplate	/soap.php/wsdlcompat/mailingtemplate
Mandator	/soap.php/wsdl/mandator	/soap.php/wsdlcompat/mandator
Pool	/soap.php/wsdl/pool	/soap.php/wsdlcompat/pool
Pool Data Miner	/soap.php/wsdl/pooldataminer	/soap.php/wsdlcompat/pooldataminer
Profiles	/soap.php/wsdl/profile	/soap.php/wsdlcompat/profile
Report	/soap.php/wsdl/report	/soap.php/wsdlcompat/report
Scoring	/soap.php/wsdl/scoring	/soap.php/wsdlcompat/scoring
SmartLink	/soap.php/wsdl/smartlink	/soap.php/wsdlcompat/smartlink
Targetgroup	/soap.php/wsdl/targetgroup	/soap.php/wsdlcompat/targetgroup
User	/soap.php/wsdl/user	/soap.php/wsdlcompat/user

Supported protocols

SOAP is currently supported as protocol. This protocol is independent of language; server and clients can be programmed in any language such as Java, Perl, Python, PHP or Ruby.

For SOAP, libraries are an integral component of many programming languages.

Details on libraries can be found at https://de.wikipedia.org/wiki/SOAP#Implementierungen

error handling

All exceptions returned by the SOAP API indicate either incorrect parameters (e.g. unknown profile or object ID) or illegal operations (missing authorizations, deletion dependencies). In this category of errors, automatic repetition makes no sense.

The SOAP API does not have a maintenance mode. This means that if the email marketing system cannot be reached due to a breakdown or maintenance work, the SOAP API cannot be reached either. In this case the client of the customer will generate an error (Timeout, Connction reset by peer, Connection refused, etc.) and an automated repetition after a few minutes should lead to success again. The same applies to network problems (with the customer, with the email marketing system or in between).

Authentication

The authentication always takes place via *Basic Auth <https://en.wikipedia.org/wiki/Basic_access_authentication>*_. The required steps are generally taken from the libraries used.

The user used to access the API requires the right *Webservices 2.0*.

Access to objects or resources

The following object types can be processed via the interface:

- Article
- Article templates
- Article types
- Container
- Container types
- File
- Form/form alias
- Image
- Emailing
- Mailing templates
- Pool
- Pool Data Miner
- Report
- Targetgroup

The following methods are available for all object types or resource types:

Reading out type ID's belonging to an object class Method:

`EvalancheResourceTypeInformation[] getTypeIds()`

Return value:

`EvalancheResourceTypeInformation[]`: List of EvalancheResourceTypeInformation objects

This method is required for namespaces that administer more than one resource type such as a in `mailing`.

Permissions needed:

None

Reading out the default folder of an object class Method:

`EvalancheCategoryInformation getResourceDefaultCategory(int mandator_id)`

Return value:

`EvalancheCategoryInformation`: Object with the following attributes:

`int id`: ID of the default folder

`string name`: Name of the default folder

Permissions needed:

None

Querying information on an object via object ID Method:

`EvalancheResourceInformation getById(int resource_id)`

Parameter:

`int resource_id`: ID of the resource

Return value:

`EvalancheResourceInformation`: Object with the following attributes:

`int id`: ID of the resource

`string name`: Name of the resource

`string url`: URL of the resource

`int type_id`: Type ID of the resource

`int category_id`: ID of the folder in which the resource is saved

`int customer_id`: ID of the mandator in which the resource is saved

Permissions needed:

None

Reading out information on objects with a particular type ID **Method**:

`EvalancheResourceInformation[] getByTypeId(int type_id, int customer_id)`

Parameter:

`int type_id`: Type ID of the objects that are to be queried

`int customer_id`: ID of the mandator in which objects are to be looked for. If this value is 0, the search takes place on the global level

Return value:

`EvalancheResourceInformation[]`: List of objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Reading out information on objects in a particular folder **Method**:

`EvalancheResourceInformation[] getByCategory(int category_id)`

Parameter:

`int category_id`: ID of the folder in which objects are to be looked for

Return value:

`EvalancheResourceInformation[]`: List of objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Reading out all objects of the namespace **Method**:

`EvalancheResourceInformation[] getAll(int customer_id)`

Parameter:

`int customer_id`: Id of the client in which you want to search for objects. If this value is `0`, the search is carried out on a global level, provided it is a global user on the global level. For client users, the pools of the client are also output with parameter 0.

Return value:

`EvalancheResourceInformation[]`: List of objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Rename an object **Method**:

`EvalancheResourceInformation rename(resource_id, name)`

Parameter

`int resource_id`: ID of a resource that is to be renamed.

`string name`: Name to be assigned to the resource

Return value

`EvalancheResourceInformation`: Objects analogous to the method `getById` (**'Querying information on an object via object ID'_**)

Permissions needed:

Modify resource

Copying an object **Method**:

`EvalancheResourceInformation copy(int resource_id, int category_id)`

Parameter:

`int resource_id`: ID of the resource

`int category_id`: ID of the target folder

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Create resource

Moving an object **Method**:

`EvalancheResourceInformation move(int resource_id, int category_id)`

Parameter:

`int resource_id`: ID of the resource

`int category_id`: ID of the target folder

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Administer resource

Deleting an object **Method**:

`EvalancheResourceInformation delete(int resource_id)`

Parameter:

`int resource_id`: ID of the resource

Return value:

`bool`: `true` if successful

Permissions needed:

Administer resource

Article The following methods are available in addition to the general ones (**'General methods'_**):

Reading out article data **Method**:

`EvalancheHashMap getData(int article_id)`

Parameter:

`int article_id`: ID of the article

Return value:

`EvalancheHashMap`: **'EvalancheHashMap'_** with the data of the article

Permissions needed:

Show article

Create article **Method**:

`EvalancheResourceInformation create (int article_preset_id, string name, EvalancheHashMap data, int category_id)`

Parameter:

`int article_preset_id`: ID of the article type that is to be used

`string name`: Name of the article

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the article

`int category_id`: ID of the folder in which the article is to be created

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Create article

Updating an article **Method**:

`EvalancheResourceInformation update (int article_id, EvalancheHashMap data)`

Parameter:

`int article_id`: ID of the article

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the article

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Change article

Article Type The following methods are available in addition to the general ones (**'General methods'_**):

Folgende Attribute-IDs stehen für die Änderung oder Anlage von Artikel-Typ-Attributen zur Verfügung:

`1` - Einzeilige Eingabe `10` - Mehrzeilige Eingabe (Text) `8` - Mehrzeilige Eingabe (formatiert) `2` - Mehrzeilige Eingabe (HTML-Editor) `6` - Einfachselektion `7` - Mehrfachselektion `13` - Boolean (Wahr/Falsch) `4` - Links (Liste) `3` - Bild `18` - Bild (Liste) `5` - Datum mit Uhrzeit `9` - Dokumentation `11` - Trenner `12` - Farbe `14` - Dokumentation (Html) `15` - Container `20` - Container (Liste) `16` - Url `17` - Code Macro `19` - Werte (Liste)

Create article type **Method**:

`EvalancheResourceInformation create (string name, int category_id)`

Parameter:

`string name`: Name of the article type `int category_id`: Id of the folder, in which the article type should be created

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Create article category

Query article type attributes **Method**:

`EvalancheResourceInformation getAttributes(int resource_id)`

Parameter:

`int resource_id`: Id of the article type

Return value:

EvalancheArticletypeAttributes[]: List of objects with the following attributes: `id`: ID of the attribute `name`: Name of the attribute `label`: Label of the attribute `type_id`: ID of the attribute type `group_id`: ID of the group of the attribute `help_text`: Hiletext of the attribute `input_help_text`: Input help text of the attribute `mandatory`: Mandatory field option of the attribute `visible`: Display option of the attribute `replacement_variable`: Replacement variable of the attribute `allows_options`: Options option creation possibility

Permissions needed:

Display article type

Creating Article Type Attributes **Method**:

`EvalancheResourceInformation addAttribute(int resource_id, string name)`

Parameter:

`int resource_id`: ID of the article type `string name`: Name of the new attribute `string label`: Label of the new attribute `string type_id`: ID of the attribute type of the new attribute `string group_id`: ID of the group of the new attribute

Return value:

EvalancheArticletypeAttributes[]: List of objects with the following attributes: `id`: ID of the new attribute `name`: Name of the new attribute `label`: Label of the new attribute `type_id`: ID of the attribute type of the new attribute `group_id`: ID of the group of the new attribute `help_text`: Hiletext of the new attribute `input_help_text`: Input help text of the new attribute `mandatory`: Mandatory field option of the new attribute `visible`: Display option of the new attribute `replacement_variable`: Replacement variable of the new attribute `allows_options`: Option creation possibility

Permissions needed:

Display article type

Changing Article Type Attributes **Method**:

`EvalancheResourceInformation updateAttribute(int resource_id, int attribute_id, EvalancheHashMap data, string name)`

Parameter:

`int resource_id`: ID of the article type `int attribute_id`: ID of the attribute to be changed `EvalancheHashMap data`: `EvalancheHashMap data`: **‘EvalancheHashMap‘_** with the data of the article

Return value:

EvalancheArticletypeAttributes[]: List of objects with the following attributes: `id`: ID of the changed attribute `name`: Name of the changed attribute `label`: Label of the changed attribute `type_id`: ID of the attribute type of the changed attribute `group_id`: ID of the group of the changed attribute `help_text`: Hiletext of the changed attribute `input_help_text`: Input help text of the changed attribute `mandatory`: Mandatory field option of the changed attribute `visible`: Display option of the changed attribute `replacement_variable`: Replacement variable of the changed attribute `allows_options`: Option creation possibility

Permissions needed:

Change article type

Remove article type attribute **Method**:

`EvalancheResourceInformation removeAttribute (int resource_id, int attribute_id)`

Parameter:

`int resource_id`: ID of the article type from which the attribute is to be removed `int attribute_id`: ID of the attribute to be removed

Return value:

`removeAttributeResult`: True/False if the removal was successful

Permissions needed:

Change article type

Query attribute group **Method**:

`EvalancheResourceInformation getAttributeGroups(int resource_id)`

Parameter:

`int resource_id`: Id of the article type

Return value:

`EvalancheArticletypeGroups[]`: List of objects with the following attributes: `id`: ID of the attribute group `name`: name of the attribute group `sort_order`: position in the order of the attribute groups

Permissions needed:

Display article type

Creating an Attribute Group **Method**:

`EvalancheResourceInformation addAttributeGroup(int resource_id, string name)`

Parameter:

`resource_id`: Id of article type `name`: Name of attribute group

Return value:

`EvalancheArticletypeGroups[]`: List of objects with the following attributes: `id`: ID of the new attribute group `name`: Name of the attribute group `sort_order`: Position in the order of the attribute groups

Permissions needed:

Change article type

Remove Attribute Group **Method**:

`EvalancheResourceInformation removeAttributeGroup(int resource_id, int attribute_group_id)`

Parameter:

`resource_id`: Id of the article type `attribute_group_id`: Name of the attribute group

Return value:

`removeAttributeResult`: True/False if the removal was successful

Permissions needed:

Change article type

Query Attribute Options **Method**:

`EvalancheResourceInformation getAttributeOptions(int resource_id, int attribute_id)`

Parameter:

`int resource_id`: ID of the article type from which the attribute options are to be retrieved `int attribute_id`: ID of the attribute from which the attribute options are to be retrieved

Return value:

`EvalancheArticletypeAttributeOptions[]`: List of objects with the following attributes:

`id`: ID of the attribute option `name`: Name of the attribute option `label`: Label of the attribute option `order`: Position in sequence

Permissions needed:

Display article type

Create Attribute Option **Method**:

`EvalancheResourceInformation createAttributeOption (int resource_id, int attribute_id, string label)`

Parameter:

`int resource_id`: ID of the article type `int attribute_id`: ID of the attribute `string label`: Label of the attribute

Return value:

`id`: ID of the new Attribute option `name`: Name of the new Attribute option `label`: Label of the new Attribute option `order`: Position of the new Attribute option

Permissions needed:

Change article type

Remove Attribute Options **Method**:

`EvalancheResourceInformation removeAttributeOption (int resource_id, int attribute_id, int option_id)`

Parameter:

`int resource_id`: ID of the article type `int attribute_id`: ID of the attribute to be removed `int option_id`: ID of the option to be removed

Return value:

`removeAttributeResult`: True/False if the removal was successful

Permissions needed:

Display article type

Query Attribute Types **Method**:

`EvalancheResourceInformation getTypeIds()`

Parameter:

Return value:

`EvalancheArticletypeAttributeTypes[]`: List of objects with the following attributes:

`id`: ID of the attribute type `description`: Description of the attribute type

Permissions needed:

Display article type

Changing Attribute Types **Method**:

`EvalancheResourceInformation changeAttributeType(int resource_id, int attribute_id, int type_id)`

Parameter:

`int resource_id`: Id of the article type `int attribute_id`: Id of the article type `int type_id`: Id of the article type

Return value:

`changeTypeResult`: True/False if the change was successful

Permissions needed:

Change article type

Query Applicable Article Type Attribute Roles **Method**:

`EvalancheResourceInformation getApplicableRoleTypes(int resource_id, int attribute_id)`

Parameter:

`int resource_id`: Id of the article type `int attribute_id`: Id of the article type attribute

Return value:

`EvalancheApplicableRoleTypes[]`: List of objects with the following attributes: `id`: ID of the role `name`: Name of the role

Permissions needed:

Display article type

Querying Assigned Article Type Attribute Roles **Method**:

`EvalancheResourceInformation getAssignedRoleTypes(int resource_id, int attribute_id)`

Parameter:

`int resource_id`: Id of the article type `int attribute_id`: Id of the article type attribute

Return value:

`EvalancheAssignedRoleTypes[]`: List of objects with the following attributes: `id`: ID of the role `name`: Name of the role

Permissions needed:

Display article type

Assign an Article Type Attribute to a Role **Method**:

`EvalancheResourceInformation assignRoleToAttribute(int resource_id, int attribute_id, int role_type_id)`

Parameter:

`int resource_id`: Id of the article type `int attribute_id`: Id of the article type attribute `int role_type_id`: Id of the article type attribute

Return value:

`changeTypeResult`: True/False if the change was successful

Permissions needed:

Change article type

Artikel-Vorlage

Artikel-Vorlage anlegen **Method**:

`EvalancheResourceInformation create(string title, int type+id, string template, int folder_id)`

Parameter:

`string title`: Name der Artikel-Vorlage `int type_id`: Typ der Artikel-Vorlage (33=Email; 34=Text; 35=PDF; 36=Web/Mobile) `string template`: Inhalt des Templates `int folder_id`: Id des Ordners in dem die Vorlage erstellt werden soll

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Artikel-Vorlage erzeugen

Artikel-Vorlage ändern **Method**:

`EvalancheResourceInformation updateTemplate(int resource_id, int attribute_id, int role_type_id)`

Parameter:

`int template_id`: Id der Artikel-Vorlage `string template`: Inhalt des Templates

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Artikel-Vorlage verändern

Campaign The following methods are available in addition to the general ones (**'General methods'_**):

Read details of a campaign **Method**:

`EvalancheResourceInformation getDetails(int[] profile_id_list, int resource_id)`

Parameter:

`int resource_id`: ID of the campaign whose details are to be queried.

Return value:

`EvalancheCampaignInformation`: Object (SOAP) or array with the following attributes:
`int id`: ID of the queried campaign object `string name`: Name of the campaign object `int date_start`: Timestamp of the start date specified in the campaign object `int date_end`: Timestamp of the start date specified in the campaign object `string external_id`: External ID specified in the campaign object `int category_id`: ID of the folder in which the campaign object is located `string description`: Description text defined in the campaign object `int state`: Status of the campaign; 1=Activated; 2=Paused; `int profile_count`: Number of currently active profiles in the campaign

Move profiles into a campaign Method:

`EvalancheResourceInformation pushProfilesIntoCampaign(int[] profile_id_list, int resource_id)`

Parameter:

`int[] profile_id_list`: List of profile IDs to move into the campaign. `int resource_id`: ID of the campaign into which the profiles should be moved.

Return value:

`EvalancheResourceInformation`: True/False whether the profiles were successfully moved into the campaign.

Kampagne konfiguriert anlegen Method:

`EvalancheResourceInformation createConfigured(string name, int schema_version, string configration, int category_id)`

Parameter:

`string name`: Name der neuen Kampagne `int schema_version`: Version des JSON-Schemas (aktuell: 7) `string configuration`: Kampagnen-Konfiguration im JSON-Schema `int category_id`: Id des Ordners

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Kampagne nach Startdatum suchen Method:

`EvalancheResourceInformation getByStartDateRange(int from, int to)`

Parameter:

`int from`: Timestamp des Beginns des Zeitintervalls `int to`: Timestamp des Endes des Zeitintervalls

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Kampagne nach Enddatum suchen Method:

`EvalancheResourceInformation getByEndDateRange(int from, int to)`

Parameter:

`int from`: Timestamp des Beginns des Zeitintervalls `int to`: Timestamp des Endes des Zeitintervalls

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Kampagne nach externer Id suchen **Method**:

`EvalancheResourceInformation getByEndDateRange(string external_id)`

Parameter:

`string external_id`: External_Id der Kampagne

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Kampagnen-Details auslesen **Method**:

`EvalancheResourceInformation getDetails(int resource_id)`

Parameter:

`int resource_id`: Id der Kampagne

Return value:

`int id`: Id der Kampagne `string name`: Name der Kampagne `int date_start`: Timestamp des Startzeitpunktes der Kampagne `int date_end`: Timestamp des Endzeitpunktes der Kampagne `string external_id`: Externe Id der Kampagne `int category_id`: Id des Ordners, in dem sich die Kampagne befindet `string description`: Beschreibung der Kampagne `int state`: Status der Kampagne (0=pausiert, 1=aktiviert) `int profile_count`: Anzahl der aktiven Profile

Container The following methods are available in addition to the general ones (**'General methods'_**):

Read container **Method**:

`EvalancheHashMap getData(int container_id)`

Parameter:

`int container_id`: Id des Containers

Return value:

`EvalancheHashMap`: **'EvalancheHashMap'_** with a container's data

Permissions needed:

Show container

Edit container **Method**:

`EvalancheResourceInformation create (int container_preset_id, string name, EvalancheHashMap data, int category_id)`

Parameter:

`int container_preset_id`: Id of the container type to be used

`string name`: Name of container

`EvalancheHashMap data`: **'EvalancheHashMap'_** with data of the container

`int category_id`: Id of the folder in which the container is to be created

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Create container

Update container **Method**:

`EvalancheResourceInformation update (int container_id, EvalancheHashMap data)`

Parameter:

`int container_id`: Id des Containers

`EvalancheHashMap data`: **'EvalancheHashMap'_** with data of the container

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Modify container

Container type The following methods are available in addition to the general ones (**'General methods'_**):

Folgende Attribute-IDs stehen für die Änderung oder Anlage von Container-Typ-Attributen zur Verfügung:

`1` - Einzeilige Eingabe `10` - Mehrzeilige Eingabe (Text) `8` - Mehrzeilige Eingabe (formatiert) `2` - Mehrzeilige Eingabe (HTML-Editor) `6` - Einfachselektion `7` - Mehrfachselektion `13` - Boolean (Wahr/Falsch) `4` - Links (Liste) `3` - Bild `18` - Bild (Liste) `5` - Datum mit Uhrzeit `9` - Dokumentation `11` - Trenner `12` - Farbe `14` - Dokumentation (Html) `15` - Container `20` - Container (Liste) `16` - Url `17` - Code Macro `19` - Werte (Liste)

Create container type **Method**:

`EvalancheResourceInformation create (string name, int category_id)`

Parameter:

`string name`: Name of the container type `int category_id`: Id of the folder, in which the container type is to be created

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Create container type

Query container type attributes Method:

`EvalancheResourceInformation getAttributes(int resource_id)`

Parameter:

`int resource_id`: Id of the container type

Return value:

`EvalancheContainertypeAttributes[]`: List of objects with the following attributes: `id`: ID of the attribute `name`: Name of the attribute `label`: Label of the attribute `type_id`: ID of the attribute type `group_id`: ID of the group of the attribute `help_text`: Hiletext of the attribute `input_help_text`: Input help text of the attribute `mandatory`: Mandatory field option of the attribute `visible`: Display option of the attribute `replacement_variable`: Replacement variable of the attribute `allows_options`: Options option creation possibility

Permissions needed:

Display container type

Creating Container Type Attributes Method:

`EvalancheResourceInformation addAttribute(int resource_id, string name)`

Parameter:

`int resource_id`: ID of the container type `string name`: Name of the new attribute `string label`: Label of the new attribute `string type_id`: ID of the attribute type of the new attribute `string group_id`: ID of the group of the new attribute

Return value:

`EvalancheContainertypeAttributes[]`: List of objects with the following attributes: `id`: ID of the new attribute `name`: Name of the new attribute `label`: Label of the new attribute `type_id`: ID of the attribute type of the new attribute `group_id`: ID of the group of the new attribute `help_text`: Hiletext of the new attribute `input_help_text`: Input help text of the new attribute `mandatory`: Mandatory field option of the new attribute `visible`: Display option of the new attribute `replacement_variable`: Replacement variable of the new attribute `allows_options`: Option creation possibility

Permissions needed:

Display container type

Changing Container Type Attributes Method:

`EvalancheResourceInformation updateAttribute(int resource_id, int attribute_id, EvalancheHashMap data, string name)`

Parameter:

`int resource_id`: ID of the container type `int attribute_id`: ID of the attribute to be changed `EvalancheHashMap data`: `EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the container

Return value:

EvalancheContainertypeAttributes[]: List of objects with the following attributes: id: ID of the changed attribute name: Name of the changed attribute label: Label of the changed attribute type_id: ID of the attribute type of the changed attribute group_id: ID of the group of the changed attribute help_text: Hiletext of the changed attribute input_help_text: Input help text of the changed attribute mandatory: Mandatory field option of the changed attribute visible: Display option of the changed attribute replacement_variable: Replacement variable of the changed attribute allows_options: Option creation possibility

Permissions needed:

Change container type

Remove container type attribute **Method**:

`EvalancheResourceInformation removeAttribute (int resource_id, int attribute_id)`

Parameter:

int resource_id: ID of the container type from which the attribute is to be removed int attribute_id: ID of the attribute to be removed

Return value:

removeAttributeResult: True/False if the removal was successful

Permissions needed:

Change container type

Query attribute group **Method**:

`EvalancheResourceInformation getAttributeGroups(int resource_id)`

Parameter:

int resource_id: Id of the container type

Return value:

EvalancheContainertypeGroups[]: List of objects with the following attributes: id: ID of the attribute group name: name of the attribute group sort_order: position in the order of the attribute groups

Permissions needed:

Display container type

Creating an Attribute Group **Method**:

`EvalancheResourceInformation addAttributeGroup(int resource_id, string name)`

Parameter:

resource_id: Id of the container type name: Name of the attribute group

Return value:

`EvalancheContainertypeGroups[]`: List of objects with the following attributes: `id`: ID of the new attribute group `name`: Name of the attribute group `sort_order`: Position in the order of the attribute groups

Permissions needed:

Change container type

Remove Attribute Group **Method**:

`EvalancheResourceInformation removeAttributeGroup(int resource_id, int attribute_group_id)`

Parameter:

`resource_id`: Id of the container type `attribute_group_id`: Name of the attribute group

Return value:

`removeAttributeResult`: True/False if the removal was successful

Permissions needed:

Change container type

Query Attribute Options **Method**:

`EvalancheResourceInformation getAttributeOptions(int resource_id, int attribute_id)`

Parameter:

`int resource_id`: ID of the container type from which the attribute options are to be retrieved `int attribute_id`: ID of the attribute from which the attribute options are to be retrieved

Return value:

`EvalancheContainertypeAttributeOptions[]`: List of objects with the following attributes:

`id`: ID of the attribute option `name`: Name of the attribute option `label`: Label of the attribute option `order`: Position in sequence

Permissions needed:

Display container type

Create Attribute Option **Method**:

`EvalancheResourceInformation createAttributeOption (int resource_id, int attribute_id, string label)`

Parameter:

`int resource_id`: ID of the container type `int attribute_id`: ID of the attribute `string label`: Label of the attribute

Return value:

`id`: ID of the new Attribute option `name`: Name of the new Attribute option `label`: Label of the new Attribute option `order`: Position of the new Attribute option

Permissions needed:

Change container type

Remove Attribute Options **Method**:

`EvalancheResourceInformation removeAttributeOption (int resource_id, int attribute_id, int option_id)`

Parameter:

`int resource_id`: ID of the container type `int attribute_id`: ID of the attribute to be removed `int option_id`: ID of the option to be removed

Return value:

`removeAttributeResult`: True/False if the removal was successful

Permissions needed:

Display container type

Query Attribute Types **Method**:

`EvalancheResourceInformation getTypeIds()`

Parameter:

Return value:

`EvalancheContainertypeAttributeTypes[]`: List of objects with the following attributes:

`id`: ID of the attribute type `description`: Description of the attribute type

Permissions needed:

Display container type

Changing Attribute Types **Method**:

`EvalancheResourceInformation changeAttributeType(int resource_id, int attribute_id, int type_id)`

Parameter:

`int resource_id`: Id of the container type `int attribute_id`: Id of the container type `int type_id`: Id of the container type

Return value:

`changeTypeResult`: True/False if the change was successful

Permissions needed:

Change container type

Blacklist The following methods are available:

Add blacklist entry **Method**:

`bool addToBlacklist(int mandator_id, string entry, string comment = ”)`

Parameter:

`int mandator_id`: ID of the client in which the blacklist entry is to be created. With global users, the ID `0` addresses the global blacklist; with client users, the ID addresses the blacklist assigned to the client.

Please note that global users who have changed to a client via the system interface, will be deemed as client users from this time onwards and will ignore the parameter `mandator_id` until the global user has changed again to the "Global Overview" via the system interface.

`string entry`: E-mail address (such as `mustermann@example.com`) or e-mail domain (such as `@example.com`) that is to be entered in the blacklist

`string comment`: Optional comment

Return value:

`bool`: `true` if the entry is successful, `false` if the entry already exists.

Permissions needed:

Edit blacklist

Form/form alias The following methods are available in addition to the general ones (**'General methods'_**):

Updating an individual template **Method**:

`EvalancheResourceInformation updateTemplate(int form_id, string template)`

Parameter:

`int form_id`: ID of the form

`string template`: Template as a well-designed XML string, UTF-8 coded

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Adding an attribute option **Method**:

`bool addAttributeOption(int form_id, int option_id)`

Parameter:

`int form_id`: ID of the form

`int option_id`: ID of the pool attribute option

Return value:

`bool`: `true` if successful

Permissions needed:

None

Removing an attribute option **Method**:

`bool removeAttributeOption(int form_id, int option_id)`

Parameter:

`int form_id`: ID of the form

`int option_id`: ID of the pool attribute option

Return value:

`bool`: `true` if successful

Permissions needed:

None

Creating a form alias **Method**:

`EvalancheResourceInformation createAlias (int form_id, string name, int category_id)`

Parameter:

`int form_id`: ID of the form

`string name`: Name of the form alias

`int category_id`: ID of the folder in which the alias is to be created

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Reading out form aliases for a particular form **Method**:

`EvalancheResourceInformation[] getAliases (int form_id)`

Parameter:

`int form_id`: ID of the form

Return value:

`EvalancheResourceInformation[]`: List of objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Reading out the form of an alias **Method**:

`EvalancheResourceInformation getFormByAlias(int form_id)`

Parameter:

`int form_id`: ID of the form

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Reading out form statistics **Method**:

`EvalancheFormStatistics[] getStatistics(int form_id, bool with_aliases)`

Parameter:

`int form_id`: ID of the form

`bool with_aliases`: If this parameter is `true`, the statistics of all aliases of the form are returned as well

Return value:

`EvalancheFormStatistics[]`: List of objects with the following attributes:

`int id`: ID of the form or alias form

`string name`: Name of the form or alias form

`bool is_alias`: `true` if it is an alias form

`int impressions`: Number of page impressions or openings

`int succeeded`: Number of successful entries

`int identity_errors`: Number of identity errors (in login forms)

`int duplication_errors`: Number of prevented double entries `int validation_errors`: Number of validation errors

`int mandatory_errors`: Number of mandatory fields violations

Permissions needed:

None

Image The following methods are available in addition to the general ones (**'General methods'_**):

Creating an image **Method**:

`EvalancheResourceInformation create(string image_base64, string name, int category_id )`

Parameter:

`string image_base64`: base64-encrypted image data

`string name`: Name of the image to be created

`int category_id`: ID of the folder in which the image is to be created

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Generate image

Emailing The following methods are available in addition to the general ones (**'General methods'_**):

Querying general recipient data **Methods**:

`getRecipientsProfiles (int mailing_id, string[] attribute_names)`: Provides the list of recipients

`getImpressionProfiles (int mailing_id, string[] attribute_names)`: Provides the list of openers

`getMultipleImpressionProfiles (int mailing_id, string[] attribute_names)`: Provides the list of multiple openers

`getClickProfiles (int mailing_id, string[] attribute_names)`: Provides the list of clickers

`getMultipleClickProfiles (int mailing_id, string[] attribute_names)`: Provides the list of multiple clickers

`getBounceProfiles (int mailing_id, string[] attribute_names)`: Provides the list of profiles that have generated a soft or hard bounce

`getHardbounceProfiles (int mailing_id, string[] attribute_names)`: Provides the list of profiles that have generated a hard bounce `getSoftbounceProfiles (int mailing_id, string[] attribute_names)`: Provides the list of profiles that have generated a soft bounce

`getUnsubscriptionProfiles (int mailing_id, string[] attribute_names)`: Provides the list of profiles that have unsubscribed via the mailing

`getAllArticleImpressionProfiles (int mailing_id, string[] attribute_names)`: Provides the list of profiles that have clicked any article in the mailing

`getAllLinkClickProfiles (int mailing_id, string[] attribute_names)`: Provides the list of profiles that have clicked any link in the mailing

`getArticleImpressionProfiles (int mailing_id, int article_id, string[] attribute_names)`: Provides the list of clickers for a particular article

`getLinkClickProfiles (int mailing_id, int link_id, string[] attribute_names)`: Provides the list of clickers for a particular link

Parameter:

`int mailing_id`: ID of the mailing for which the recipient data is to be requested

`int article_id`: (only with `getArticleImpressionProfiles`): ID of the article for which the clickers are to be returned

`int link_id`: (only with `getLinkClickProfiles`): ID of the link for which the clickers are to be returned

`string[] attribute_names`: List of pool attribute names that are to be returned in the result

Return value:

`EvalancheJobResult`: Information on the background job, analogous to **'Export of all profiles of a pool'_** or under **'Working with background jobs'_**

Permissions needed:

View eMailing and export profiles

Querying details of a transmission **Method**:

`EvalancheMailingInformation getDetails(int mailing_id)`

Parameter:

`int mailing_id`: ID of the transmission

Return value:

`EvalancheMailingInformation`: Object (SOAP) or array with the following attributes:

- `int timestamp`: Time stamp of the mailing
- `int recipients`: Number of recipients
- `bool sent`: `true` when the mailing has been completely transmitted
- `string preview_url`: URL to the preview of the transmission
- `string report_url`: URL to the statistics page of the transmission
- `string admin_url`: URL to the management page of the transmission
- `string subject`: Default subject line of the transmission

`int targetgroup_id`: ID of the target group to which the mailing has been sent

`int send_start_time`: UNIX time stamp of the transmission start time

`int send_end time`: UNIX time stamp of the transmission end time

Permissions needed:

Show eMailing

Query the history of a profile Method:

`EvalancheMailingTrackingHistory[] getTrackingHistory(int profile_id, int from = 0, int to = 0)`

Parameter:

`int profile_id`: Id of the profile

`int from = 0`: UTC Unix Timestamp, starting from which time tracking data should be considered

`int to = 0`: UTC Unix Timestamp, up to and including which time tracking data are to be considered

Return value:

`EvalancheMailingTrackingHistory[]`: List of objects with the following attributes:

`int id`: Definite ID of the tracking data set

`int resource_id`: Unique ID of the tracked object

`string resource_name`: Name of the tracked object

`int resource_type_id`: Unique id of the tracked object type

`int sub_resource_id`: Unique ID of the object involved (e.g. article within an eMailing)

`string sub_resource_name`: Name of the object involved

`int sub_resource_type_id`: Type of object involved

`string sub_url`: Direct URL involved

`int profile_id`: Unique Id of the tracked profile

`int type`: Type of tracking

`int timestamp`: Unix timestamp of tracking

`string referrer_domain`: Involved domain

Querying openings/impressions of a transmission Method:

`EvalancheMailingImpression[] getImpressions(int mailing_id, int start_timestamp = 0, int end_timestamp = 0)`

Parameter:

`int mailing_id`: ID of the transmission

`int start_timestamp = 0`: UTC UNIX Timestamp, from which point in time (inclusive) tracking data is to be considered

`int end_timestamp = 0`: UTC UNIX Timestamp, up to which point in time (inclusive) tracking data is to be considered

Return value:

`EvalancheMailingImpression[]`: List of objects with the following attributes:

`int id`: Definite ID of the tracking data set

`int profile_id`: ID of the user that cas performed the action

`int timestamp`: UTC UNIX timestamp of the action

Permissions needed:

Show eMailing

Querying clicks of a transmission **Method**:

```
EvalancheMailingClick[] getClicks(int mailing_id, int start_timestamp
= 0, int end_timestamp = 0)
```

Parameter:

`int mailing_id`: ID of the transmission

`int start_timestamp = 0`: UTC UNIX Timestamp, from which point in time (inclusive) tracking data is to be considered

`int end_timestamp = 0`: UTC UNIX Timestamp, up to which point in time (inclusive) tracking data is to be considered

Return value:

`EvalancheMailingClick[]`: List of objects with the following attributes:

`int id`: Definite ID of the tracking data set

`int profile_id`: ID of the user that cas performed the action

`int timestamp`: UTC UNIX timestamp of the action

`int link_id`: ID of the link for which a click was recorded. If the value `-1` is returned, the `parent_id` should be considered

`int link_type_id`: Format of the resource in which the user performed the action. The following values are possible:

`0`: Unknown

`1`: Online view of a newsletter

`2`: PDF view of an article or newsletter

`3`: Text view of the newsletter

`4`: Web/Mobile view of the newsletter

`5`: Landing page of an article

`6`: Click on landing page link

`7`: LeadPage Opening

`11`: XSL-FO view of an article or newsletter

`12`: Display the RSS feed of a newsletter

`19`: Open newsletter

`20`: External link

`21`: Display the ATOM feed of a newsletter

`22`: Call preview service for an article or newsletter

`23`: Call the article or newsletter via the preview service

`24`: Call preview service for a newsletter podcast

`25`: Call the podcast data via the preview service

`40`: Article shared in Facebook

`41`: Article shared in Google Bookmarks

`42`: Article shared in MySpace

`43`: Article shared in del.icio.us

`44`: Article shared in Twitter

`45`: Article shared in Xing

`46`: Article shared in LinkedIn

`47`: Article shared in Google+

`50`: Click on article shared in Facebook

`51`: Click on article shared in Google Bookmarks

`52`: Click on article shared in MySpace

`53`: Click on article shared in del.icio.us

`54`: Click on article shared in Twitter

`55`: Click on article shared in Xing

`59`: Click on article shared in LinkedIn

`60`: Click on article shared in Google+

`int parent_id`: ID of the resource in which the link was clicked; either the mailing ID or the ID of the article reference.

Explanation of ‘‘link_id‘‘ and ‘‘parent_id‘‘:

The‘‘link_id‘‘ only then has a value greater or equal to `0` if it involves a “normal” link. Clicks, for example, on landing page links, PDF generation, voice, etc. always have a `link_id` equal to `-1`.

In this case, the `parent_id` must be used for closer consideration:

- If the `parent_id` is the same as the higher-level `mailing_id`, the click occurred in the mailing, or the mailing was depicted in the `link_type_id` format. This can for example be a click to read aloud, or the PDF generation in the mailing itself.
- If the `parent_id` is not the same as the higher-level `mailing_id`, the click referred to the article referenced with the ID equivalent to the `parent_id`, for example by a click on a sharing link of the article.

Permissions needed:

Show eMailing

Querying cumulated statistics of a transmission **Method**:

`EvalancheMailingStatistics getStatistics(int mailing_id)`

Parameter:

`int mailing_id`: ID of the transmission

Return value:

`EvalancheMailingStatistics`: Object (SOAP) or array with the following attributes:

- `int addressees`: Number of addressees
- `int recipients`: Number of recipients
- `int duplicates`: Number of profiles ignored by the duplication check
- `int blacklisted`: Number of profiles ignored by blacklisting
- `int robinsonlisted`: Number of profiles ignored by the Robinson list
- `int hardbounces`: Number of hard bounces
- `int softbounces`: Number of soft bounces
- `int unsubscribes`: Number of unsubscriptions
- `int impressions`: Number of openings
- `int unique_impressions`: Number of openers
- `int clicks`: Number of clicks
- `int unique_clicks`: Number of unique clicks
- `EvalancheMediaStatistics media`: Object (SOAP) or array with the following attributes:
  - `string name`: Name of the medium
  - `int clicks`: Number of clicks
  - `int unique_clicks`: Number of clickers
- `EvalancheArticleStatistics articles`: Object (SOAP) or array with the following attributes:
  - `int id`: ID of the article
  - `string name`: Name of the article
  - `int clicks`: Number of clicks
  - `int unique_clicks`: Number of clickers
- `EvalancheLinkStatistics links`: Object (SOAP) or array with the following attributes:

`int id`: ID of the link

`string name`: URL of the link

`int clicks`: Number of clicks

`int unique_clicks`: Number of clickers

`string url`: URL of the link

Permissions needed:

Show eMailing

Querying detailed article statistics of a transmission **Method**

`ArticleStats[] getArticleStatistics(int mailing_id)`

Parameter

`int mailing_id`: ID of the transmission

Return value

`ArticleStats[]`: List of objects with the following attributes:

`int id`: ID of the article

`int reference_id`: Id of the article reference (link between article and transmission)

`string name`: Name of the article

`ArticleFormatStats overall`: Statistics of the article for all output formats

`ArticleFormatStats landing page`: Statistics of the landing page clicks of the article

`ArticleFormatStats print`: Statistics of the PDF/Print clicks of the article

`ArticleFormatStats voice`: Statistics of the Voice clicks of the article

`ArticleFormatStats social_sharing`: Statistics of the share to social clicks of the article

Objects of type `ArticleFormatStats` have the following attributes:

`int clicks`: Number of clicks on the corresponding format of the article

`int unique_clicks`: Number of unique clicks on the corresponding format of the article

`float clickrate`: Clickrate of the corresponding format of the article

`float clickrate_relative`: Relative clickrate of the corresponding format of the article

`float multiple_clickrate`: Multiple clickrate of the corresponding format of the article

`float multiple_clickrate_relative`: Relative multiple clickrate of the corresponding format of the article

`ArticleLinkStats[] links`: List of objects with the following attributes:

`int id`: Link Id

`string url`: Link URL

`string name`: Name of the link

`int clicks`: Number of clicks on this link

`int unique_clicks`: Number of unique licks on this link

Permissions needed:

None

Reading out the status of the recipient profiles **Method**:

`EvalancheProfileMailingStatus[] getStatus(int mailing_id, int timeframe, string[] profile_attributes)`

Parameter:

`int mailing_id`: ID of the transmission

`int timeframe`: Seconds from now in the past, from which status changes shall be queried.

`string[] profile_attributes`: List of profile attributes that are to be returned in the result. If an empty array is transferred, no profile data is returned

Return value:

`EvalancheProfileMailingStatus[]`: List of objects (SOAP) or arrays with the following attributes:

`int profile_id`: ID of the profile

`int newsletter_id`: ID of the transmission

`int last_status_change`: UNIX time stamp of the time of the last status change of the profile in question

`int status`: Transmission status of the profile in question

`string preview_url`: Personalized mailing preview

`string profile_data`: Array with the attribute names as the key and the attribute value as the value

Permissions needed:

Show eMailing

**The dispatch status means: **

`0` = Queue

`1` = Shipping successful

`2` = Hardbounce

`3` = Softbounce

Reading out the subject lines of a mailing **Method**:

`EvalancheMailingSubject getSubjects(int mailing_id)`

Parameter:

`int mailing_id`: ID of the mailing

Return value:

`EvalancheMailingSubject[]`: List of objects with the following attributes:

`int targetgroup_id`: ID of the target group for the subject line. ID 0 represents the default subject line

`string subjectline`: The subject line

Permissions needed:

Show eMailing

Setting the subject lines of a mailing **Method**:

`bool setSubjects (int mailing_id, EvalancheMailingSubject[] subjects)`

Parameter:

`int mailing_id`: ID of the mailing

`EvalancheMailingSubject[] subjects`: Analogous to the return value under **'Reading out the subject lines of a mailing'_**. A configuration with the target group ID `0` is mandatory.

Return value:

`bool`: `true` if successful

Permissions needed:

Modify eMailing

Read client statistic of an eMailing **Method**:

`bool getClientStatistics (int mailing_id)`

Parameter:

`int mailing_id`: ID of the mailing

Return value:

`ClientStatistics`: Objekt (SOAP)/Array with the following attributes:

`mail_clients`: Array with the following attributes:

`item`: Array with the following attributes:

`string description`: Email client description

`int count`: Number of profiles tracked with this email client

`browsers`: Array with the following attributes:

`item`: Array with the following attributes:

`string description`: Browser description

`int count`: Number of profiles tracked with this browser

`devices`: Array with the following attributes:

`item`: Array with the following attributes:

`string description`: Device description

`int count`: Number of profiles tracked with this device

Permissions needed:

Show eMailing

Reading out the configuration of a mailing **Method**:

`EvalancheMailingConfiguration getConfiguration(int mailing_id)`

Parameter:

`int mailing_id`: ID of the mailing

Return value:

`EvalancheMailingConfiguration` Object with the following attributes:

`string external_trackingcode`: Value of the field `External tracking code`

`string campaign_id`: Value of the field `Campaign ID`

`string externalxml_url`: Value of the field `URL to external XML`

`string salutation_female`: Value of the field `Individualized salutation: Ms.`

`string salutation_male`: Value of the field `Individualized salutation: Mr.`

`string salutation_company`: Value of the field `Individualized salutation: Company`

`string salutation_family`: Value of the field `Individualized salutation: Family`

`string salutation_other`: Value of the field `Individualized salutation: Not specified`

`string sender_email`: Value of the field `Sender address`

`string sender_name`: Value of the field `Sender`

`string reply_email`: Value of the field `Reply address`

`string reply_name`: Value of the field `Reply to`

`string grant_url`: Value of the field `Confirmation URL to subscriptions`

`string revoke_url`: Value of the field `Confirmation URL to unsubscriptions`

`string inputfield_0`: Value of the field `INPUTFIELD 0`

`string inputfield_1`: Value of the field `INPUTFIELD 1` `string inputfield_2`: Value of the field `INPUTFIELD 2`

`string inputfield_3`: Value of the field `INPUTFIELD 3`

`string inputfield_4`: Value of the field `INPUTFIELD 4`

`string inputfield_5`: Value of the field `INPUTFIELD 5`

`string inputfield_6`: Value of the field `INPUTFIELD 6`

`string inputfield_7`: Value of the field `INPUTFIELD 7`

`string inputfield_8`: Value of the field `INPUTFIELD 8`

`string inputfield_9`: Value of the field `INPUTFIELD 9`

`string textarea_0`: Value of the field `TEXTAREA 0`

`string textarea_1`: Value of the field `TEXTAREA 1`

`string textarea_2`: Value of the field `TEXTAREA 2`

`string textarea_3`: Value of the field `TEXTAREA 3`

`string textarea_4`: Value of the field `TEXTAREA 4`

`string textarea_5`: Value of the field `TEXTAREA 5`

`string textarea_6`: Value of the field `TEXTAREA 6`

`string textarea_7`: Value of the field `TEXTAREA 7`

`string textarea_8`: Value of the field `TEXTAREA 8`

`string textarea_9`: Value of the field `TEXTAREA 9`

`string htmlarea_0`: Value of the field `HTMLAREA 0` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_1`: Value of the field `HTMLAREA 1` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_2`: Value of the field `HTMLAREA 2` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_3`: Value of the field `HTMLAREA 3` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_4`: Value of the field `HTMLAREA 4` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_5`: Value of the field `HTMLAREA 5` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_6`: Value of the field `HTMLAREA 6` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_7`: Value of the field `HTMLAREA 7` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_8`: Value of the field `HTMLAREA 8` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

`string htmlarea_9`: Value of the field `HTMLAREA 9` - **The valid XHTML is essential!**: Value of the field - **The valid XHTML is essential!**

Permissions needed:

Show eMailing

Setting the configuration of a mailing **Method**:

`bool setConfiguration(int mailing_id, EvalancheMailingConfiguration configuration)`

Parameter:

`int mailing_id`: ID of the mailing

`EvalancheMailingConfiguration`: Object analogous to the return value under **'Reading out the configuration of a mailing'_**

Return value:

`bool`: `true` if successful

Permissions needed:

Modify eMailing

Reading out mailing drafts ready for transmission **Method**:

`EvalancheResourceInformation[] getSendableDrafts(bool unsent)`

Parameter:

`bool unsent`: If `true` is transmitted, only drafts that have not yet been transmitted will be returned

Return value:

`EvalancheResourceInformation[]`: List of objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Read ready-to-send mailing drafts from a client **Method**:

`EvalancheResourceInformation[] getSendableDraftsByMandatorId(int mandator_id, bool unsent)`

Parameter:

`int mandator_id`: The id of the client from which drafts are to be searched `bool unsent`: If `true` is passed, only drafts which have not yet been sent are returned

Return value:

`EvalancheResourceInformation[]`: List of objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

None

Creating a mailing from a template Method:

`EvalancheResourceInformation createDraft(string name, int template_id, int category_id)`

Parameter:

`string name`: Name of the mailing to be created

`int template_id`: ID of the template to be used

`int category_id`: ID of the folder in which the mailing is to be created

Return value:

`EvalancheResourceInformation`: Object analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Create eMailing

Read out the article of a mailing Method:

`MailingArticle[] getArticles(int mailing_id)`

Parameter:

`int mailing_id`: The ID of the mailing from which the articles are to be read out

Return value:

`MailingArticle[]`: List of objects with the following attributes:

`int id`: ID of the article reference (link between the actual article and mailing)

`int article_id`: ID of the article object

`int targetgroup_id`: ID of the target group exclusively for which the articles are to be displayed.

`int sortpos`: Position of the article within its slot starting at 0

`int slot`: Number of slot in which the articles are located starting at 0

`int html_preset_id`: ID of the assigned article template for the e-mail (HTML) view. The ID `0` means that the corresponding standard article template is used

`int landingpage_preset_id`: ID of the assigned article template for the landing page view. The ID `0` means that the corresponding standard article template is used

`int text_preset_id`: ID of the assigned article template for the text view. The ID `0` means that the corresponding standard article template is used

`int pdf_preset_id`: ID of the assigned article template for the PDF view. The ID `0` means that the corresponding standard article template is used

`int mobile_preset_id`: ID of the assigned article template for the cell phone or web view. The ID `0` means that the corresponding standard article template is used

Permissions needed:

Show eMailing

Remove all articles of a mailing **Method**:

`bool removeAllArticles(int mailing_id)`

Parameter:

`int mailing_id`: The ID of the mailing from which the articles are to be removed. The article objects per se are not changed

Return value:

`bool`: In case of success `true`

Permissions needed:

Modify eMailing

Remove a specific article of a mailing **Method**:

`MailingArticle[] removeArticles(int mailing_id, int[] reference_ids)`

Parameter:

`int mailing_id`: The ID of the mailing from which the articles are to be removed. The article objects per se are not changed

`int[] reference_ids`: List of article reference IDs of the articles to be removed. The IDs can be determined by the `getArticles` method (`MailingArticle:id`)

Return value:

`MailingArticle[]`: List of objects analogous to the return value with **'Read out the article of a mailing'_** - after removing the indicated articles.

Permissions needed:

Modify eMailing

Add articles to a mailing and update the existing articles **Method**:

`MailingArticle[] addArticles(int mailing_id, MailingArticle[] articles)`

Parameter:

`int mailing_id`: The ID of the mailing in which the articles are to be updated or inserted

`MailingArticle[] articles`: List of objects analogous to the return value with **'Read out the article of a mailing'_**. Whether an article is created or updated is controlled by the `MailingArticle:id` attribute of an article. If a valid article reference ID is transferred here, the corresponding article is updated. If `0` is transferred, the article is inserted.

Return value:

List of objects analogous to the return value with **'Read out the article of a mailing'_** - after inserting or updating the articles

Permissions needed:

Modify eMailing

Sending a mailing to a target group **Method**:

`EvalancheMailingInformation sendToTargetgroup(int mailing_id, int targetgroup_id, int send_time, int speed)`

Parameter:

`int mailing_id`: ID of the mailing draft to be transmitted

`int targetgroup_id`: ID of the target group

`int send_time`: UNIX time stamp of the transmission time (UTC). If this is in the past, the transmission will begin immediately

`int speed`: Transmission speed in mails/hour. If this value is 0, no reduction will take place

Return value:

`EvalancheMailingInformation`: Object analogous to the method `getDetails` (**'Querying details of a transmission'_**)

Permissions needed:

Manage eMailing

Transmitting a mailing to individual profiles **Method**:

`int[] sendToProfiles(int mailing_id, int[] profile_ids)`

Parameter:

`int mailing_id`: Id des Mailings `int[] profile_ids`: Liste von Profil Ids

Return value:

`int[]`: List with the ID's of the profiles to which the transmission was successful

Permissions needed:

Manage eMailing

Pool The following methods are available in addition to the general ones (**'General methods'_**):

Reading out the attributes of a pool **Method**:

`EvalanchePoolAttribute[] getAttributes`

Parameter:

`int pool_id`: ID of the pool

Return value:

`EvalanchePoolAttribute[]`: List of objects with the following attributes:

`int id`: ID of the attribute

`string name`: Internal name of the attribute

`string label`: Display name of the attribute

`bool has_options`: `true` when this attribute allows options `bool can_add_options`: `true` when options can be added to this attribute

`EvalanchePoolAttributeOption[] options`: List of objects with the following attributes:

`int id`: ID of the attribute option

`string value`: Display name of the attribute option

`int type_id`: ID of the type of the attribute. The following values are possible:

`1`: SALUTATION

`2`: COUNTRY

`3`: STATE

`4`: MOBILENETWORK

`5`: INPUT (one-line text input)

`6`: MULTILINEINPUT (multi-line input)

`7`: SELECTION (single selection)

`8`: MULTIPLESELECTION (multiple selection)

`9`: DATE

`10`: EMAIL

`11`: DATETIME (date with time)

`12`: LANGUAGE

`14`: DATETIMERO (date with time, read-only)

`15`: PERMISSION

`16`: BOOLEAN (true/false)

`17`: HARDBOUNCES

`18`: USERSTATE (user status, deprecated)

`19`: FORM (origin form)

`20`: INT (whole number)

`21`: ZIPCODE (postal code)

Permissions needed:

View pool

Creating an attribute **Method**:

`EvalanchePoolAttribute addAttribute(int pool_id, string name, string label, int type_id`

Parameter:

`int pool_id`: ID of Pools `string name`: Internal name of the attribute (only capital letters and numbers, first character must be a capital letter)``string label``: Display name of the attribute `int type_id`: Type ID of the field, see return value under **'Reading out the attributes of a pool'_**

Return value:

`EvalanchePoolAttribute`: An object of the type EvalanchePoolAttribute analogous to the return value of the `getAttributes` methods

Permissions needed:

Modify pool

Creating attribute options **Method**:

`EvalanchePoolAttribute addAttributeOptions(int pool_id, int attribute_id, string[] labels`

Parameter:

`int pool_id`: ID of the pool

`int attribute_id`: ID of the attribute

`string[] labels`: List with the display names of the options to be created

Return value:

`EvalanchePoolAttribute`: An object of the type EvalanchePoolAttribute analogous to the return value of the `getAttributes` method

Permissions needed:

None

Deleting attribute options **Method**:

`EvalanchePoolAttribute deleteAttributeOption(int pool_id, int attribute_id, int option_id)`

Parameter:

`int pool_id`: ID of the pool

`int attribute_id`: ID of the attribute

`int option_id`: ID of the option to be deleted

Return value:

`EvalanchePoolAttribute`: An object of the type EvalanchePoolAttribute analogous to the return value of the `getAttributes` method

Permissions needed:

None

Updating attribute options **Method**:

`EvalanchePoolAttribute updateAttributeOption(int pool_id, int attribute_id, int option_id, string label)`

Parameter:

`int pool_id`: ID of the pool

`int attribute_id`: ID of the attribute

`int option_id`: ID of the option to be deleted

`string label`: New display name for the option

Return value:

`EvalanchePoolAttribute`: An object of the type EvalanchePoolAttribute analogous to the return value of the `getAttributes` method

Permissions needed:

None

Report The following methods are available in addition to the general ones (**'General methods'_**):

Creation of a report **Method**:

`EvalancheResourceInformation create(string name, int category_id)`

Parameter:

`string name`: Internal name of the report

`int category_id`: ID of the folder in which the report is to be created

Return value:

`EvalancheResourceInformation`: Objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Create report

Inserting an Object in a Report **Method**:

`EvalancheResourceInformation addResourceToReport(int resource_id, int report_id)`

Parameter:

`int resource_id`: Id of the object to be inserted into the report.

`int report_id`: Id of the report into which the object is to be inserted.

Return value:

`bool`: In case of success `true`

Permissions needed:

Change report

Targetgroup The following methods are available in addition to the general ones (**'General methods'_**):

Creation of a target group using an attribute option **Method**:

`EvalancheResourceInformation createByOption (int pool_id, int attribute_id, int option_id, int category_id, string name)`

Parameter:

`string pool_id`: ID of the pool in which the attribute option is located

`int attribute_id`: ID of the attribute in which the option is located

`int option_id`: ID of the attribute option which is to be used to create the target group

`int category_id`: ID of the folder in which the target group is to be created

`string name`: Name of the target group to be created

Return value:

`EvalancheResourceInformation`: Objects analogous to the `getById` method (**'Querying information on an object via object ID'_**)

Permissions needed:

Zielgruppe erzeugen

Reading out information on a target group **Method**:

`EvalancheTargetgroupInformation getInformation(int targetgroup_id)`

Parameter:

`int targetgroup_id`: ID of the target group

Return value:

`EvalancheTargetgroupInformation`: Object with the following attributes:

`string name`: Internal name of the target group

`int profile_count`: Number of profiles that are currently present in the target group

Permissions needed:

View target group

SmartLink The following methods are available in addition to the general ones (**'General methods'_**):

New creation of a SmartLink Method:

`EvalancheResourceInformation create (string name, int category_id,)`

Parameter:

`string name`: Name of the SmartLink

`int category_id`: Id of the folder in which the SmartLink was created

Return value:

`string url`: External URL of the SmartLink

`int type_id`: Type_id of the SmartLink

`int category_id`: Id of the folder in which the SmartLink was created

`int customer_id`: Id of the client in which the SmartLink was created

`int id`: Id of the SmartLink

`string name`: Name of the SmartLink

Permissions needed:

Create SmartLink

New creation of a SmartLink link Method:

`EvalancheResourceInformation createLink (int smartlink_id, string link_name, string link_url,)`

Parameter:

`int smartlink_id`: Id of the SmartLink object in which the new SmartLink link is to be created.

`string link_name`: Name of the new SmartLink link.

`string link_url`: Url of the new SmartLink link.

Return value:

`string url`: Integration url of the new SmartLink link.

Permissions needed:

Change SmartLink

Reading the integration URLs of a SmartLink Method:

`EvalancheResourceInformation getTrackingUrls (int smartlink_id)`

Parameter:

`int smartlink_id`: Id of the SmartLink object whose integration URLs are to be read.

Return value:

`int id`: ID of the SmartLink link. `string name`: Name of the SmartLink link. `string url`: Integration url of the SmartLink link.

Permissions needed:

Display SmartLink

Profiles

Creation of a new profile **Method**:

`int create(int pool_id, EvalancheHashMap data)`

Parameter:

`int pool_id`: The ID of the pool in which the profile is to be created

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the profile

Return value:

`int profile_id`: The ID of the newly created profile

Permissions needed:

View profile list

Deleting profiles **Method**:

`bool delete(int[] profile_id)`

Parameter:

`int[] profile_id`: A list of the ID of the profiles to be deleted

Return value:

`bool`: `true` if successful

Permissions needed:

Delete profiles

Revoking the permission/unsubscription of a profile **Method**:

`bool revokePermission(int profile_id)`

Parameter:

`int profile_id`: ID of the profile to be unsubscribed

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Revoke the tracking approval of a profile **Method**:

```
bool revokeTracking(int profile_id)
```

Parameter:

`int profile_id`: Id of the profile that should no longer be tracked

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Confirmation of a profile **Method**:

```
bool grantPermission(int profile_id)
```

Parameter:

`int profile_id`: ID of the profile to be confirmed

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Looking for a profile according to the profile ID **Method**:

```
EvalancheHashMap getById(string profile_id)
```

Parameter:

`string profile_id`: ID or UID of the profile to be searched

Return value:

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the profile

Permissions needed:

Export profiles

Search for profiles with a particular attribute value **Method**:

```
EvalancheHashMap[] getByKey(int pool_id, string key, string value,
string[] pool_attribute_list)
```

Parameter:

`int pool_id`: ID of the pool in which to search

`string key`: Name of the pool attribute that is they key for the search, e.g. `EMAIL`

`string value`: Value to be searched for, e.g. `mustermann@example.com`

`string[] pool_attribute_list`. List with names of the pool attributes that are to be returned in the result

Return value:

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the profile

Permissions needed:

Export profiles

Export of all hard bounces of a pool **Method**:

`ProfileBounceStatus[] getBounces(int pool_id, int start_time, int end_time, string[] pool_attribute_list)`

Parameter:

`int pool_id`: ID of the pool to be requested

`int start_time`: UTC Unix timestamp of the time from which bounces are to be returned

`int end_time`: UTC Unix timestamp of the time up to (and including) which bounces are to be returned . If `0` is transferred here, the current time is used

`string[] pool_attribute_list`: Optional array of strings with names of pool attributes that are to be returned in the result

Return value:

`ProfileBounceStatus[]`: List of objects with the following attributes:

`int profile_id`: ID of the relevant profile

`int mailing_id`: ID of the mailing in which the bounce occurred

`int status`: Bounce status. Possible values are:

`1`: Softbounce

`2`: Hardbounce

`int timestamp`: UTC Unix timestamp of the time at which the bounce occurred

`EvalancheHashMap profile_data`: **'EvalancheHashMap'_** with the data of their own profile

Permissions needed:

Export profiles

Export of all changed profiles of a pool **Method**:

`EvalancheHashMap[] getModifiedProfiles(int pool_id, string[] attributes, int start_time, end end_time = 0)`

Parameter:

`int pool_id`: ID of the pool to be requested

`string[] attributes`: List of the pool attribute names whose values are to be returned for found profiles

`int start_time`: UTC Unix timestamp of the time from which changed profiles are to be returned

`int end_time`: UTC Unix timestamp of the time up to which changed profiles are to be returned

Return value:

`EvalancheHashMap[]`: List of **'EvalancheHashMap'_** with the data of the relevant profiles

Permissions needed:

Export profiles

Export of all unsubscriptions of a pool **Method**:

`EvalancheHashMap[] getUnsubscriptions(int pool_id, int start_time, int end_time, string[] pool_attribute_list)`

Parameter:

`int pool_id`: ID of the pool to be requested

`int start_time`: UTC Unix timestamp of the time from which unsubscriptions are to be returned

`int end_time`: UTC Unix timestamp of the time up to which unsubscriptions are to be returned. If `0` is transferred here, the current time is used

`string[] pool_attribute_list`: Optional array of strings with names of pool attributes that are to be returned in the result

Return value:

`EvalancheHashMap[]`: List of **'EvalancheHashMap'_** with the data of the relevant profiles

Permissions needed:

Export profiles

Export of all double opt-in confirmed profiles of a pool

Method: `EvalancheHashMap[] getGrantedPermissions(int pool_id, string[] attributes, int start_time, end end_time = 0)`

Parameter:

`int pool_id`: ID of the pool to be requested

`string[] attributes`: List of the pool attribute names whose values are to be returned for found profiles `int start_time`: UTC Unix timestamp of the time from which double opt-in confirmations are to be returned

`int end_time`: UTC Unix timestamp of the time up to which double opt-in confirmations are to be returned

Return value:

`EvalancheHashMap[]`: List of **'EvalancheHashMap'_** with the data of the relevant profiles

Permissions needed:

Export profiles

Export of all profiles of a target group **Method**:

`EvalancheJobHandle getByTargetGroup(int targetgroup_id, string[] pool_attribute_list)`

Parameter:

`int targetgroup_id`: ID of the target group to be exported

`string[] pool_attribute_list`: List of the attribute names that are to be returned in the result

Return value:

`EvalancheJobHandle`: An object of the type EvalancheJobHandle with the following attributes:

`string id`: The ID of the background job

`int status`: The status of the background job with the following values:

0: Queue

1: In processing

2: Finished

3: Temporary error (processing is repeated)

4: Error

`string status_description`: Description of the status

`namespace`: Namespace under which the background job has been started, here always `profile`

`string method`: Name of the method that has started the background job, here always `getByTargetGroup`

`int resource_id`: ID of the resource for which the background job was started, here always `targetgroup_id`

`int result_chunks`: Number of result blocks found. This value is not equal to 0 only in status 2

Permissions needed:

View target group and export profiles

Example:

'Working with background jobs'_

Export of all profiles of a pool **Method**:

`EvalancheJobHandle getByPool(int pool_id, string[] pool_attribute_list)`

Parameter:

`int targetgroup_id`: ID of the pool to be exported

`string[] pool_attribute_list`: List of the attribute names that are to be returned in the result

Return value:

`EvalancheJobHandle`: An object of the type EvalancheJobHandle with the following attributes:

`string id`: The ID of the background job

`int status`: The status of the background job with the following values:

0: Queue

1: In processing

2: Finished

3: Temporary error (processing is repeated)

4: Error

`string status_description`: Description of the status

`namespace`: Namespace under which the background job has been started, here always `profile`

`string method`: Name of the method that started the background job, here always `getByPool`

`int resource_id`: ID of the resource for which the background job was started, here always `pool_id`

`int result_chunks`: Number of result blocks found. This value is not equal to 0 only in status 2

Permissions needed:

View pool and export profiles

Example:

'Working with background jobs'_

Querying the status of a background job **Method**:

`EvalancheJobHandle getJobInformation(string job_id)`

Parameter:

`string job_id`: The ID of a background job, as returned e.g. by a previous `getByPool`

Return value:

`EvalancheJobHandle`: An object of the type EvalancheJobHandle analogous to the methods `getByPool` and `getByTargetGroup`

Permissions needed:

None

Retrieving the results of a background job Background jobs can be queried for 28 days.

Method:

`EvalancheJobResult getResults(string job_id)`

Parameter:

`string job_id`: The ID of a background job, as returned e.g. by a previous `getByPool`

Return value:

`EvalancheJobResult`: An object of the type EvalancheJobResult with the following attributes:

`string id`: The ID of the background job

`int status`: The status of the background job with the following values:

0: Queue

1: In processing

2: Finished

3: Temporary error (processing is repeated)

4: Error

`string status_description`: Description of the status

`namespace`: Namespace under which the background job has been started, here always `profile`

`string method`: Name of the method that started the background job, here always `getByPool`

`int resource_id`: ID of the resource for which the background job was started, here always `pool_id`

`int result_chunks`: Number of result blocks found. This value is not equal to 0 only in status 2

`chunks_left`: Number of result blocks that are still ready for retrieval

`result_size`: Number of profiles found

Permissions needed:

None

Example:

'Working with background jobs'_

Querying the cursor position of a background-job result **Method**:

`int getResultCursor(string job_id)`

Parameter:

`string job_id`: The ID of a background job, as returned e.g. by a previous `getByPool`

Return value:

`int`: Value of cursor starting with 0

Permissions needed:

None

Setting the cursor position of a background-job result **Method**:

`bool setResultCursor(string job_id, int cursor)`

Parameter:

`string job_id`: The ID of a background job, as returned e.g. by a previous `getByPool`

`int cursor`: Value of cursor starting with 0

Return value:

`bool`: `true` if cursor was set successfully, `false` if not successful (e.g. in case of an invalid value)

Permissions needed:

None

Checking whether a profile is in particular target groups **Method**:

`EvalancheTargetgroupMembership[] isInTargetgroups(int profile_id, int[] targetgroup_ids)`

Parameter:

`int profile_id`: ID of the profile that is to be checked

`int[] targetgroup_ids`: List of target group ID's that are to be checked

Return value:

`EvalancheTargetgroupMembership[]`: List of EvalancheTargetgroupMembership objects with the following attributes:

- `int profile_id`: ID of the checked profile
- `int targetgroup_id`: ID of the target group in which the profile is located

Permissions needed:

View profile list

Tagging profiles with an attribute option tagWithOption:

`bool tagWithOption(int pool_attribute_option_id, string[] values, string key, bool set_profile_edit_date)`

Parameter:

`int pool_attribute_option_id`: ID of the attribute option that is to be set

`string[] values`: List of strings with values that are to be looked for

`string key`: Name of the pool attribute in which `values` are to be looked for

`bool set_profile_edit_date`: If `true` is indicated here, the change date for all relevant profiles is set to the present

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Deleting an attribute option for profiles (untagging) Method:

`bool untagWithOption(int pool_attribute_option_id, string[] values, string key, bool set_profile_edit_date)`

Parameter:

`int pool_attribute_option_id`: ID of the attribute option that is to be deleted

`string[] values`: List of strings with values that are to be looked for

`string key`: Name of the pool attribute in which `values` are to be looked for

`bool set_profile_edit_date`: If `true` is indicated here, the change date for all relevant profiles is set to the present

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Updating a profile using the profile ID Method:

`bool updateById(int profile_id, EvalancheHashMap data)`: Multiple selections are set precisely as indicated, and existing values in the profile are overwritten

`bool mergeById(int profile_id, EvalancheHashMap data)`: Multiple selections are combined, and transferred options are combined with existing options.

Parameter:

`int profile_id`: ID of the profile that is to be updated

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the profile

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Updating profiles on the basis of an attribute **Method**:

`bool updateByKey(int pool_id, string key, string value, EvalancheHashMap data)`: Multiple selections are set precisely as indicated, and existing values in the profile are overwritten

`bool mergeByKey(int pool_id, string key, string value, EvalancheHashMap data)`: Multiple selections are combined, and transferred options are combined with existing options.

Parameter:

`int pool_id`: ID of the pool in which profiles are to be updated

`string key`: Name of the pool attribute in which to look

'string value': Value that is to be looked for in `key`

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the profile

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Updating the profiles of a target group **Method**:

`bool updateByTargetGroup(int target_group_id, EvalancheHashMap data)`: Multiple selections are set precisely as indicated, and existing values in the profile are overwritten

`bool mergeByTargetGroup(int target_group_id, EvalancheHashMap data)`: Multiple selections are combined, and transferred options are combined with existing options.

Parameter:

`int target_group_id`: ID of the target group in which profiles are to be updated

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the profile

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Updating the profiles of a pool **Method**:

`bool updateByPool(int pool_id, EvalancheHashMap data)`: Multiple selections are set precisely as indicated, and existing values in the profile are overwritten

`bool mergeByPool(int pool_id, EvalancheHashMap data)`: Multiple selections are combined, and transferred options are combined with existing options.

Parameter:

`int pool_id`: ID of the pool in which profiles are to be updated

`EvalancheHashMap data`: **'EvalancheHashMap'_** with the data of the profile

Return value:

`bool`: `true` if successful

Permissions needed:

View profile list

Mass import of profiles **Method**:

`MassUpdateResult massUpdate(int pool_id, string key_attribute_name, string[] attributes, string[][] data, bool merge, bool ignore_missing)`

Parameter:

`int pool_id`: The ID of the pool in which the profiles are to be updated or created

`string key_attribute_name`: Name of the pool attribute which is used to check whether a profile already exists. If an empty string is transferred, new profiles are created.

`string[] attributes`: List of pool attribute names that are to be used for importing or updating. If `key_attribute_name` is not empty, it must also be listed here

`string[][] data`: List of the lists of profile data. Each entry of the first dimension of the list represents a profile to be created or updated. The values in the list of the second dimension must be transferred exactly in the sequence in which the names of the attributes are indicated under `attributes`

`bool merge`: Establishes whether values of multiple selection fields should be combined or overwritten by data from the request

`bool ignore_missing`: Establishes whether profiles that do not yet exist in the pool should be ignored or created

Return value:

`MassUpdateResult`: Object with the following attributes:

- `EvalancheHashMap updated`: List of EvalancheHashMapItems where the attribute `key` contains the profile idea of the updated profile, and the attribute `value` contains the value of the field from `key_attribute_name`

- `EvalancheHashMap created`: List of EvalancheHashMapItems where the attribute `key` contains the profile idea of the created profile, and the attribute `value` contains the value of the field from `key_attribute_name` `string[] ignored`: List of values of `key_attribute_name` where the corresponding profiles were not updated. This case can occur in an update of a profile with unchanged data, or when `ignore_missing == true` was found and not a profile of `key_attribute_name`

`int[] error`: List of record numbers from `data` starting with `0` where an error occurred

Example of a request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:prof="https://
   <soapenv:Header/>
   <soapenv:Body>
      <prof:massUpdate>
         <pool_id>000000</pool_id>
         <key_attribute_name>EXTERNALID</key_attribute_name>
         <attributes>
            <!--Zero or more repetitions:-->
            <item>EXTERNALID</item>
            <item>EMAIL</item>
            <item>FIRSTNAME</item>
            <item>NAME</item>
            <item>SALUTATION</item>
         </attributes>
         <data>
            <!--Zero or more repetitions:-->
            <item>
               <!--Zero or more repetitions:-->
               <item>47-11</item>
               <item>julien.bashir@example.com</item>
               <item>Julien</item>
               <item>Bashir</item>
               <item>1</item>
            </item>
            <item>
               <!--Zero or more repetitions:-->
               <item>47-12</item>
               <item>william.picard@example.com</item>
               <item>William</item>
               <item>Picard</item>
               <item>2</item>
            </item>
            <item>
               <!--Zero or more repetitions:-->
               <item>48-11</item>
               <item>benjamin.sisko@example.com</item>
               <item>Benjamin</item>
               <item>Sisko</item>
               <item>2</item>
            </item>
            <item>
               <!--Zero or more repetitions:-->
               <item>48-11</item>
               <item>benjamin.sisko@example.com</item>
               <item>Benjamin</item>
               <item>Sisko</item>
            </item>
         </data>
         <merge>true</merge>
         <ignore_missing>false</ignore_missing>
```

```
        </prof:massUpdate>
    </soapenv:Body>
</soapenv:Envelope>
```

Example of a response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="https:/
    <SOAP-ENV:Body>
        <ns1:massUpdateResponse>
            <massUpdateResult>
                <updated>
                    <items>
                        <item>
                            <key>5718855</key>
                            <value>47-11</value>
                        </item>
                    </items>
                </updated>
                <created>
                    <items>
                        <item>
                            <key>5718856</key>
                            <value>47-13</value>
                        </item>
                    </items>
                </created>
                <ignored>
                    <item>47-12</item>
                </ignored>
                <error>
                     <item>3</item>
                </error>
            </massUpdateResult>
        </ns1:massUpdateResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Permissions needed:

Import profiles

Querying the mailing status for a profile **Method**:

`EvalancheProfileMailingStatus2[] getMailingStatus(int profile_id, int mailing_id)`

Parameter:

`int profile_id`: ID of the profile for which the status is to be queried

`int mailing_id`: ID of the mailing for which the status is to be queried. If `0` is transferred as the ID, all mailings that have been sent to the profile are checked

Return value:

`EvalancheProfileMailingStatus2`: An object of the type EvalancheJobResult with the following attributes:

`int profile_id`: ID of the requested profile

`int newsletter_id`: ID of the requested mailing

`int last_status_change`: Unix timestamp (UTF-8) of the last status change of the profile in reference to the mailing

`int status`: Current status of the profile in reference to the mailing

`string preview_url`: Personalized preview URL of the mailing

`EvalancheHashMap profile_data`: Always empty in this method

Permissions needed:

View profile list

Request grouped scoring values of a profile **Method**:

`ProfileGroupScore[] getScores(int profile_id)`

Parameter:

`int profile_id`: ID of the profile for which the scores are to be requested

Return value:

`ProfileGroupScore[]`: List of objects with the following attributes

`int profile_id`: ID of the profile

`int group_id`: ID of the scoring group

`int group_name`: Name of the scoring group

`int activity_score`: Sum of all activity scores of the profile referring to the aforementioned scoring group

`int profile_score`: Sum of all profile scores of the profile referring to the aforementioned scoring group

Permissions needed:

None

Request detailed scoring values of a profile **Method**:

`ProfileActivityScore[] getActivityScoringHistory(int profile_id, int scoring_group_id = 0, int scoring_type_id = 0, int resource_id = 0, int start_timestamp = 0, int end_timestamp = 0)`

Parameter:

`int profile_id`: ID of the profile for which the scoring history is to be requested

`int scoring_group_id = 0`: ID of the scoring group for which the scoring history for the profile is to be requested. If `0` is transferred here, there is no restriction

`int scoring_type_id = 0`: ID of the scoring type for which the scoring history for the profile is to be requested. If `0` is transferred here, there is no restriction `int resource_id = 0`: ID of the resource for which the scoring history for the profile is to be requested. If `0` is transferred here, there is no restriction

`int start_timestamp = 0`: Unix timestamp (UTC) of the time starting at which the scoring history is to be returned for the profile.

`int end_timestamp = 0`: Unix timestamp (UTC) of the time up to which the scoring history is to be returned for the profile. If `0` is transferred here, the current time is used

Return value:

`ProfileActivityScore[]`: List of objects with the following attributes:

`int id`: ID of the scoring entry

`int scoring_group_id`: ID of the scoring group

`int scoring_type_id`: ID of the scoring type

`int timestamp`: Unix timestamp (UTC) of the entry

`int value`: Value (score) of the entry

`int resource_id`: ID of the resource for which the entry was generated

Permissions needed:

Evaluate scoring

Values of scoring_type_id

ID	Description
1	eMailing opening
2	eMailing Click
3	Form opening
4	Registration form
5	Form Click
6	eMailing Click on Social Sharing
7	eMailing Click on Preview
8	eMailing Click on PDF
9	LeadPage Öffnung
11	SmartLink Click
12	eMailing Unsubscribe
13	Article Opening
14	Article Click
15	Article Click on Social Sharing
16	Article Click on Preview
17	Article Click on PDF
18	eMailing Click on online version
19	eMailing Click on Web/Mobile Version
20	web service call
21	WebTouchPoint Call

Write a score for a profile **Method**:

```
bool addScore(int profile_id, int scoring_group_id, int score = 0,
string name = ”)
```

Parameter:

`int profile_id`: ID of the profile for which the score is to be written

`int scoring_group_id`: ID of the scoring group to which the score is to be assigned

`int score`: Positive or negative score. If `0` is transferred, the configured default score is used

`string name`: Optional description of the score. If an empty string is transferred here, the configured default name is used

Return value:

`bool`: `true` if successful

Permissions needed:

Configure scoring

Working with background jobs Due to high runtime and data amount exports of target groups and pool will be executed as asynchronous actions. The result is being delivered as chunks with 1.000 profiles each.

The result is available for 28 days.

An example of a script for exporting all profiles of a pool in PHP could look like the following:

```
<?php

        require_once 'Zend/Soap/Client.php';

        $username = 'username';
        $password = 'password';
        $pool_id = 123456;

        $client = new Zend_Soap_Client('http://scnem.com/soap.php/wsdlcompat/profile', array('lo

        $job_info = $client->getByPool($pool_id, array('ID', 'FIRSTNAME', 'NAME', 'EMAIL', 'P134
        var_dump($job_info);

        while($job_info->status < 2 || $job_info->status == 3) { // Status prüfen, solange der J
                $job_info = $client->getJobInformation($job_info->id);
                var_dump($job_info);
                sleep(5);
        }

        if($job_info->status == 2) { // Job ist erledigt
                $result = $client->getResults($job_info->id); // erstes Resultset abholen
                var_dump($result);

                while($result->chunks_left > 0) { // weitere Resultsets abholen, falls vorhanden
                        $result = $client->getResults($job_info->id);
                        var_dump($result);
                }
        } else {
                // Der Job konnte nicht erfolgreich ausgeführt werden
        }
```

Further methods

Reading out information on the basis of the mandator ID **Method**:

`EvalancheMandator getById(int mandator_id)`

Parameter:

`int mandator_id`: ID of the mandator

Return value:

`EvalancheMandator`: Object with the following attributes:

`id`: ID of the mandator

`name`: Name of the mandator

`domain`: Domain of the mandator

Permissions needed:

None

Reading out a list of all mandators **Method**:

`EvalancheMandator[] getList()`

Return value:

`EvalancheMandator`: List of objects analogous to **'Reading out information on the basis of the mandator ID'_**

Permissions needed:

None

Users

Finding a user on the basis of the user name **Method**:

`EvalancheUser getByUsername(string username)`

Parameter:

`string username`: User name of the user being looked for

Return value:

`EvalancheUser`: Object with the following attributes:

`int id`: ID of the user

`int mandator_id`: ID of the mandator . Value `0` identifies a global user

Please note that global users who have changed to a client via the system interface, will be deemed as client users from this time onwards and will ignore the parameter `mandator_id` until the global user has changed again to the "Global Overview" via the system interface.

`string username`: User name of the user

`string email`: E-mail of the user

`int salutation`: Salutation of the user (0 = blank, 1 = Ms., 2 = Mr.)

`string firstname`: First name of the user

`string firstname`: Last name of the user

`string description`: Description of the user

`int security_guideline_id`: ID of the security guideline allocated to the user

`int[] role_ids`: List of IDs of rolls that the user assumes

`bool disabled`: `true` if the user is deactivated

`password`: Password of the user, here always value `*SCRAMBLED*`

Permissions needed:

Administer user

Reading out a list of all users **Method**:

`EvalancheUser[] getAll(int mandator_id)`

Parameter:

`int mandator_id`: ID of the mandator whose users are to be returned. Value `0` returns global users

Please note that global users who have changed to a client via the system interface, will be deemed as client users from this time onwards and will ignore the parameter `mandator_id` until the global user has changed again to the "Global Overview" via the system interface.

Return value:

`EvalancheUser[]`: List of objects analogous to **'Finding a user on the basis of the user name'_**

Permissions needed:

Administer user

Updating user data and creating users **Method**:

`EvalancheUser update(EvalancheUser user)`

Parameter:

`EvalancheUser`: User data analogous to the return value under **'Finding a user on the basis of the user name'_**. If the attribute `id` is equal to `0`, a new user is created.

Return value:

`EvalancheUser`: Object analogous to **'Finding a user on the basis of the user name'_**

Permissions needed:

Administer user

Scoring

Request scoring group **Method**:

`ScoringGroupInformation[] getGroups(int mandator_id = 0)`

Parameter:

`int mandator_id`: ID of the client for which the groups are to be requested. If *0* is transferred, the global scoring groups are returned.

Return value:

`ScoringGroupInformation[]`: List of objects with the following attributes:

`int id`: ID of the scoring group

`string name`: Name of the scoring group

`int mandator_id`: ID of the client in which the scoring group exists

Please note that global users who have changed to a client via the system interface, will be deemed as client users from this time onwards and will ignore the parameter `mandator_id` until the global user has changed again to the "Global Overview" via the system interface.

Permissions needed:

Evaluate scoring

file

Query subfolders **Method**:

`EvalancheCategoryInformation[] getSubCategories(int category_id)`

Parameter:

`int category_id`: Id of the start folder

Return value:

`EvalancheCategoryInformation[]`\`

`id`: Id of the subfolder

`name`: Name of the subfolder

Data types requiring explanation

This data type is an object with the properties `key` and `value` that are of the type `string`.

This object only contains the property `items`, an array of `EvalancheHashMapItem`. The structure of disarray can for example be read as an array in PHP:

```
<?php

array('items' =>
    array('item' =>
        array(
            array('key' => 'EMAIL', 'value' => 'max.mustermann@example.com'),
            array('key' => 'NAME', 'value' => 'Mustermann'),
```

```
                array('key' => 'SALUTATION', 'value' => '2'),
            )
        )
    );
```

In order to easily generate this structure from an associative array, the following PHP function could be used:

```
<?php

function hashmapFromArray($data) {
    foreach ($data as $key => $value) {
        $items[] = array(
                'key' => strval($key),
                'value' => strval($value)
        );
    }

    return array('items' => $items);
}
```

SOAP V1

Important information

Please note that global users who are intended to use the SOAP-API or are already used for this purpose, should not be used for working in the system interface.

If this is the case, however, you run the risk of making the global user a client user by changing to a client via the system interface, as long as this user changes again to the "Global Overview" via the system interface.

This can have repercussions on all methods that need the parameter `mandator_id`. In this case, this global user that changed to a client would ignore the parameter `mandator_id` and would use the ID of the client that he was changed to.

Restrictions

Please note that an individual request via the API must not be more than 10MB in size. This is a security policy. When transferring larger requests, these must be split up.

WSDL

Information on access points and WSDL files is found in the following table:

Port	WSDL Document/literal	WSDL RPC/encoded
Accounting	/api/soap/v1/accounting/wsdl	/api/soap/v1/accounting/wsdl
Blacklist	/api/soap/v1/blacklist/wsdl	/api/soap/v1/blacklist/wsdl

Supported protocols

SOAP is currently supported as protocol. This protocol is independent of language; server and clients can be programmed in any language such as Java, Perl, Python, PHP or Ruby.

For SOAP, libraries are an integral component of many programming languages.

Details on libraries can be found at https://de.wikipedia.org/wiki/SOAP#Implementierungen

Authentication

The authentication always takes place via *Basic Auth <https://en.wikipedia.org/wiki/Basic_access_authentication>*_. The required steps are generally taken from the libraries used.

The user used to access the API requires the right *Webservices 2.0*.

Access to objects or resources

The following object types can be processed via the interface:

- Account
- Blacklist

Account The following methods are available:

Letzte Account Abrechnung anzeigen **Method**:

```
EvalancheHashMap getAccount(int customerId)
```

Parameter:

`int customerId`: Id des Mandanten. 0 führt bei globalem API-User zu der Gesamtabrechnung des Accounts.

Return value:

`EvalancheAccount`: Abrechnung nach Customer_id (ID eines Mandanten) und Type_id (ID des Abrechnungs-Typs) geordnet.

Permissions needed:

Globale Abrechnung

Folgende TypeIds werden aktuell verwendet:

ID	Description
1	eMailing versendet
2	Profilanzahl
3	Mandanten existierend
4	Mandanten erzeugt
5	Gruppen existierend
6	Webservice (XML-RPC API)
7	PDFs generiert
8	Vorlesedienst
9	Kartenaufrufe
10	Geokodierungen
11	Client-Checks
12	Eyetracking-Checks
13	Intern

Continued on next page

Table 5.1 – continued from previous page

ID	Description
14	Intern
15	Intern
16	Intern
17	Aufrufe aus Sozialen Netzwerken
18	Aufrufe von Rss/Atom Feeds
19	Intern
20	Benutzeraccounts
21	Intern
22	Webspace
23	Textanalyse
24	Mailgrößenüberschreitung
25	Profil Backups
26	Security Keys
27	Intern
28	Mandanten kopiert
29	Aufrufe von dynamischen Bildern
30	Webservice (Reporting Api)
31	Generierte PrintMailings
32	Webservice (SOAP API)
33	Postadress-Validierung

Blacklist The following methods are available:

Adresse auf Blackliste setzen **Method**:

```
EvalancheHashMap add(int customerId, string email, string
description)
```

Parameter:

`int customerId`: Id des Mandanten. 0 führt bei globalem API-User zur globalen Blackliste. `string email`: eMail, welche gelistet werden soll. `string description`: Optional - Hinweis auf den Grund der Listung.

Return value:

`nothing`: keine Response nach Eintragung, alternativ fehlermeldung (falls bereits vorhanden)

Permissions needed:

Edit blacklist

Adressen der Blackliste anzeigen **Method**:

```
EvalancheHashMap get(int customerId)
```

Parameter:

`int customerId`: Id des Mandanten. 0 führt bei globalem API-User zur globalen Blackliste.

Return value:

`int customerId`: Id des Mandanten. `int item`: Liste mit folgenden Werten: `int email`: gelistete Adresse. `int description`: Grund der Listung, sofern vorhanden.

Permissions needed:

Edit blacklist

Adressen von der Blackliste entfernen **Method**:

`EvalancheHashMap remove(int customerId, string email)`

Parameter:

`int customerId`: Id des Mandanten. 0 führt bei globalem API-User zur globalen Blackliste. `int email`: gelistete Adresse.

Return value:

`nothing`: keine Response nach Eintragung, alternativ fehlermeldung (falls bereits vorhanden)

Permissions needed:

Edit blacklist

Reporting API

In addition to the already existing SOAP and XML-RPC interface, this API offers the possibility to obtain statistical, tracking and scoring data. These can then be integrated, evaluated and visualized in business intelligence solutions and dashboards, for example. For example, the tracking history from 00:00:00 to 23:59:59 of the previous day can be viewed with any HTTP client with the URL:

https://LOGIN_DOMAIN/report.php?table=trackinghistory&format=xml&from=yesterday&to=today-1second To ensure that the correct user is used, we recommend prefixing the user: https://USERNAME@LOGIN_DOMAIN/report.php?table=trackinghistory&format=xml&from=yesterday&to=today-1second

abgerufen werden

Authentication

Authentication is carried out via HTTP Basic Auth

Security

Access to the API is only possible via HTTPS and requires the `Reporting API` permission. In addition, table-specific rights are required, which are described in the tables.

Tables

Use the parameter `table` to determine which data are to be retrieved.

Resource types

This table provides information on the importance of the values in the column `resource_type_id` of the remaining tables. No additional authorizations are required.

table=resourcetypes

Column	Type	Description
id	int	Id of the resource type
name	string	Name of the resource type; influenced by the parameter `lang`

Tracking types

This table provides information on the importance of the values in the column `type` of the `tracking history` table. No additional authorizations are required.

table=trackingtypes

Column	Type	Description
id	int	Id of the link type
name	string	Name of the tracking type, is influenced by the parameter `long`

Folgende Tabelle zeigt die aktuellen Tracking-Types welche als "Klick" interpretiert werden. Dabei entspricht die Zuordnung derjenige, welche auch für die SOAP-Methoden getClicks verwendet werden.

id	description	metric
1	Online version of an eMailing	click
2	PDF version of an article or eMailing	click
4	Web/Mobile version of an eMailing	click
5	Landingpage of an article	click
6	Landingpage Link click	click
12	View of the RSS feed on an eMailing	click
19	Emailing impression	impression
20	External link	click
22	Utilisation of the read-aloud service for an article or eMailing	click
24	Utilisation of the read-aloud service via podcast within an eMailing	click
40	Article shared via facebook	click
41	Article shared via Google Bookmarks	click
42	Article shared via MySpace	click
44	Article shared via twitter	click
45	Article shared via Xing	click
46	Article shared via Linkedin	click
47	Article shared via Google+	click
50	Click on a shared article in Facebook	click
51	Click on a shared article in Google Bookmarks	click
52	Click on a shared article in MySpace	click
54	Click on a shared article in Twitter	click
55	Click on a shared article in Xing	click
59	Click on a shared article in Linkedin	click
60	Click on a shared article in Google+	click
1100	Unsubscription within eMailing	unsubscribe
1101	Double opt-in confirmation via newsletter	click

Mandators

This table provides a list of all clients including information as to whether or not the client can be accessed.

table=customers

Column	Type	Description
id	int	Id of the link type
name	string	Name of the tracking type, is influenced by the parameter `long`
accessible	bool	`true` if the client can be accessed with this API user. Otherwise `false`

Configured scoring groups of the current clients

This table returns the configured scoring groups of the client in which the user is currently located.

The authorization `Scoring` is required.

table=scoringgroups

Column	Type	Description
id	int	ID of the scoring group
name	string	Name of the scoring group

Pools of the current client

This table returns the list of the pools of the client in which the user is currently located. The authorization `Pool anzeigen` is required.

table=pools

Column	Type	Description
id	int	ID of pool
name	string	Name of the pool
profile_count	int	Number of the profiles in the pool
unconfirmed_profiles	int	Number of unconfirmed profiles in the pool
unsubscribed_profiles	int	Number of unsubscribed profiles in the pool
erroneous_profiles	int	Number of incorrect profiles in the pool

Forms and alias forms of the current client

This table returns the list of the forms and alias forms of the client in which the user is currently located. The authorizations `Formular anzeigen` and `Formular-Alias` are required.

table=forms

Column	Type	Description
id	int	ID of the form or alias form
name	string	Name of the form or alias form
parent_id	int/null	With the form `null` and the alias form, the ID of the higher-level form
impressions	int	Number of impressions of the form or alias form
submissions	int	Number of successful entries in the form or alias form
pool_id	int	The ID of the pool assigned to the form or alias form
is_alias	bool	`false` in the form, `true` in the alias form

Checkpoint statistics

This table returns the list of checkpoint statistics for the client in which the user is currently located. The right `Display form` and `Display form alias` is required. The data is commutated so that only the last tracking of a profile by a particular link is displayed.

table=checkpointstatistics

Column	Type	Description
id	int	Entry ID
checkpoint_id	int	Checkpoint ID
link_id	int	ID of checkpoint link
profile_id	int	ID of tracked profile
timestamp	datetime	Time at which a profile was last tracked for a checkpoint link
count	int	Number of trackings performed for a profile using a checkpoint link

Current mandator's checkpoints

This table returns the list of checkpoints of the user customer or client with the ID `customer_id`. The right `Display checkpoint` is required.

table=checkpoints

Column	Type	Description
id	int	Checkpoint ID
name	string	Checkpoint name

Current mandator's LeadPage

This table returns the list of LeadPages of the user customer or client with the ID `customer_id`. The right `Display LeadPage` is required.

table=leadpages

Column	Type	Description
id	int	LeadPage ID
name	string	LeadPage name
resource_type_id	int	LeadPage resource type ID

Profiles of a pool

This table returns the profile data of a pool that is defined by means of the parameter **pool_id**. The authorization `Profile exportieren` is required.

table=profiles

The returned columns correspond to the pool configuration

Mailings of the current client

This table returns the list of the mailings of the client in which the user is currently located. The authorization `eMailing (Versand) anzeigen` is required.

table=mailings

Column	Type	Description
id	int	ID of the transmission
name	string	The name of the mailing
addressees	int	Number of recipients
recipients	int	Number of recipients
conversions	int	Number of conversions
conversion_sum	int	Total `int` values of all conversions
conversion_sum_float	float	Total `float` values of all conversions
clicks	int	Number of kicks
click_rate	float	Click rate
click_rate_relative	float	Relative click rate
click_rate_multiple_relative	float	Relative multiple click rate
clicks_unique	int	Unique clicks
opening_rate	float	Opening rate
opening_rate_por	float	POR opening rate
opening_rate_multiple	float	Multiple opening rate
bounce_rate	float	Bounce rate
campaign_id	string	Campaign ID of the mailing
sendout_starttime	datetime	The start time of the mailing
sendout_endtime	datetime	End time of the mailing
sendout_triggertime	datetime	Time at which the mailing was created
unsubscriptions	int	Number of unsubscriptions from the mailing
unsubscription_rate	float	Unsubscription rate of the mailing
targetgroup_id	int	ID of the transmission target group used
targetgroup_name	string	Name of the transmission target group used

shipping history

This table returns the list of all shipments and recipients. The right `Display eMailing` is required.

table=newslettersendlogs

Column	Type	Description
resource_id	int	Id of dispatch / mailing
profile_id	int	Id of the sent profile
sent_date	datetime	Date of delivery
mail_size	int	Size of the dispatch / mailing in bytes
state	int	Shipping status

state

0 = Queue 1 = Successfully delivered 2 = HB 3 = SB 10 = First Retry 11 = Second Retry 12 = Error 20 = Successful print mail dispatch 21 = Incorrect print mail dispatch

Activities scoring history of the current client

This table returns the scoring entries of all profiles of the client in which the user is currently located. The authorization `Scoring` is required.

table=scoringhistory

Column	Type	Description
id	int	ID of the scoring entry
timestamp	datetime	Time at which the entry was created
resource_id	int	The ID of the resource that instigated the entry
resource_name	string	The name of the resource that instigated the entry
resource_type_id	int	The type of the resource that instigated the entry
scoringgroup_id	int	The ID of the scoring group to which the entry is assigned
profile_id	int	The ID of the profile to which the entry is assigned
scoring_value	int	Scoring value
scoring_type_id	int	Id of the scoring type
tracking_id	int	Id of the tracking
description	string	Description text of the score

Scoring of all profiles of all scoring groups of the current client

This table returns the current scoring values of all profiles of all scoring groups of the client in which the user is currently located.

The authorization `Scoring` is required.

table=profilescores

Column	Type	Description
id	int	Pseudo-ID of the entry in the form `<profile_id>_<scoring_group_id>`
profile_id	int	The ID of the profile to which the entry is assigned
scoring_group_id	int	The ID of the scoring group to which the entry is assigned
profile_score	int	Current profile score of the profile in the scoring group
activity_score	int	Current activity score of the profile in the scoring group

Tracking history of all profiles of the current client

This table provides the detailed tracking history for all profiles of the client in which the user is currently located. No special authorization is required.

table=trackinghistory

Column	Type	Description
id	int	ID of the tracking entry
resource_id	int	The ID of the resource that instigated the entry
resource_name	string	The name of the resource that instigated the entry
resource_type_id	int	The type of the resource that instigated the entry
sub_resource_id	int	The ID of the sub-resource that instigated the entry (e.g. article)
sub_resource_name	string	The name of the sub-resource that instigated the entry (e.g. article)
sub_resource_type_id	int	The type of sub-resource that instigated the entry
sub_url	string	URL of the link, if available
profile_id	int	The ID of the profile to which the entry is assigned
type	int	The type ID of the entry
timestamp	datetime	Time of the entry
referrer_domain	string	Referrer, if available

Profile Changelog

This table provides the profile change log in which permission changes and form activities are stored. A **pool_id** is required. No special authorizations are required.

table=profilechangelogs

Column	Type	Description
form_id	int	Id of the form involved
profile_id	int	Id of the profile involved
permission	int	The permission that was set during the action.
update_mode	int	Type of update
ip	string	IP address used to initiate the activity
date	int	Timestamp of the change
email	string	E-mail address of the participating profile

Update Modes

0 = Registration, 3 = Unsubscribe, 1 = Update, 2 = Update (Unique Criterion), 4 = Tracking Withdrawal

Permission Modes

1 = Enter for Double Opt-In, 2 = Unsubscribe, 3 = Enter as Opt-Out, 4 = Enter as Opt-In, 5 = Confirm as Double Opt-In, 6 = Login Only

Mandators

The parameter `customer_id` defines the client for which data is to be requested. If a local user is used within a client, the parameter is not necessary.

customer_id=<customer_id>

Formats

The parameter `format` defines the desired output format of the requested data.

format=<format> (mandatory)

Possible formats are

Parameter value	Description	MIME Type
xml	XML representation of the data	application/xml
`json` oder `jsonarray`	Representation of the data as a list of json	application/json
jsonobject	Representation of the data as a single json object	application/json
csv parameter	CSV representation of the data	text/csv

The behavior of the API can be influenced by additional parameters.

Output language

The output language affects the column `name` of the table `resourcetypes` and the table `profiles` (such as the values of the columns, `SALUTATION, COUNTRY`, etc.).

lang=<lang>

Possible languages are:

Parameter value	Description
en	Englisch (Standard)
de	German
it	Italian
fr	French

Output format of the date and time

The date and time are normally output in the format `iso8501` (such as *2014-08-15T15:52:01+0000*). This format can be influenced by the parameter `time_format`.

time_format=<time_format>

Possible formats are:

Parameter value	Example	Description
iso8601	2014-08-15T15:52:01+0000	Standard format
unix	1124121121	Unix timestamp
rfc822	Tue, 15 Aug 14 15:52:01 +0000	
rfc850	Tuesday, 15-Aug-14 15:52:01 UTC	
rfc1036	Tue, 15 Aug 14 15:52:01 +0000	
rfc1123	Tue, 15 Aug 2014 15:52:01 +0000	
rfc2822	Tue, 15 Aug 2014 15:52:01: +0000	
rfc3339	2014-08-15T15:52:01+00:00	
w3c	2014-08-15T15:52:01+00:00	

See also http://php.net/manual/en/class.datetime.php#datetime.constants.types

Time restriction of queries

The parameters `from` and `to` serve to restrict the time of query results.

`from=<date string>` (the present default is -24 hours) `to=<date_string>` (default = time of the query). The absolute date and time (such as `23.06.2014 00:00`) as well as relative information (such as `yesterday, last monday, now-24hours`, etc.) can be used as values.

Additional information:

- Note: Always bear in mind that all parameters **URL-coded** must be used!
- http://php.net/manual/en/datetime.formats.relative.php

Possible for:

- `table=profiles` (with reference to the entry/change/double opt-in date)
- `table=newslettersendlogs` (Bezogen auf den Zeitpunkt der abgeschlossenen Versendung)
- `table=scoringhistory` (with reference to the time at which the scoring entries occurred)
- `table=trackinghistory` (with reference to the time at which the tracking entries occurred)

Pool ID

pool_id=<pool_id> This parameter is mandatory for the table `profiles`.

Needed permissions for accessing reporting API

The following permissions are required to access the Reporting API:

Basically:

Reporting API

Table-Specific:

Table	Permission
articlereferences	Permission to read article
checkpoints	Permission to read checkpoints
checkpointstatistics	Permission to read checkpoints
customers	No special permissions required.
forms	Permission to read forms, permission to read form aliases
geocoordinates	GeoDB Services Permission
leadPages	Permission to read lead pages
mailings	Permission to read dispatches, split campaigns, event mailings and trigger mailings
pools	Permission to read pools
profiles	Permission to export profiles
profilescores	Permission to evaluate scoring
resourcetypes	No special permissions required.
scoringcluster	Permission to evaluate scoring
scoringgroups	Permission to evaluate scoring
scoringtypes	Permission to evaluate scoring
trackinghistory	No special permissions required.
trackingtypes	No special permissions required.

XML-RPC-API

What is XML-RPC

XML-RPC enables deleted function calls via the HTTP protocol. Remote Procedure Call or for short RPC is the designation for deleted function calls. XML serves to encode the parameter values onto the deleted procedures. Hence the name XML-RPC.

XML-RPC is language-independent, server and clients can be implemented in any languages such as Java, Perl, Python, PHP or Ruby

Details on libraries (most are now integral component of many programming languages) can be found at http://directory.xmlrpc.com/implementations

Remember that you are XML-coding XML-specific special characters:

> **<** <
>
> **>** >
>
> **&** &
>
> **“** "
>
> **‘** '

Restrictions

Please note that an individual request via the API must not be more than 10MB in size. This is a security policy. When transferring larger requests, these must be split up.

Call

The call always takes place against the URL `https://IhreDomain/xmlrpc.php`; preferred on your own domain using SSL. The encoding for all calls and responses is `UTF-8`

Authentication

The right for `XML-RPC` is necessary for accessing the XML-RPC interface. For this it is best to create a role oneself, which appears sufficiently adequate to the tasks of the external calls. Then create a user to that this role is assigned to. Use the username and password of this user for the `struct auth` during authentication of your XML-RPC client. The `struct auth` consists of the following attributes:

- `string username`
- `string password`

Error handling

Error handling is performed in the standard way for XML-RPC; the IDs for error groups are unique and the report contains more detailed information about the error.

The following classes of errors are used:

801 Authentication is not provided

802 Authentication failed

803 Invalid parameter. This mostly involves a faulty parameter or a parameter with a false value.

804 Resource not found

Example

Generic examples of programming with XML-RPC can be found at http://directory.xmlrpc.com/tutorialspress.

Sample call in Python:

```python
import xmlrpclib

server = xmlrpclib.Server('https://example.com/xmlrpc.php', encoding='UTF-8')
print server.doProfileUpdate(
            dict(username='user', password='pw'),
            12345,
            6789,
            dict(
                 EMAIL='test@example.com',
                 COMPANY='Example Inc.',
                 P3=1168965880,
                 P5=(101,102,104),
```

```
            )
    )
```

Every example shown in the functions references this example.

Functions

Insofar as IDs themselves cannot be re-selected via function calls, the IDs for objects from the lists in the system interface can be selected.

General

Server Available

Deklaration `boolean isAlive(void)`

Funktion Simple check whether the service is available.

Rückgabewert `TRUE`

Parameter *None*

Object data

Deklaration `struct getResourceInformation(struct auth, int resource_id)`

Funktion Basic attributes for a particular object. Can be used, for example, in order to receive a URL via an object's ID.

Rückgabewert `struct resource_information` with:

> **`int id`** ID that was transferred
>
> **`string name`** Internal name
>
> **`string comment`** Description
>
> **`string url`** URL to the object

Parameter

> **`int resource_id`** ID of the object

Rename an object

Deklaration `struct renameResource(struct auth, int resource_id, string name)`

Funktion Changes the field name of an object (name for internal use).

Rückgabewert `struct resource_information` as in `getResourceInformation`

Parameter

> **`int resource_id`** ID des Objektes. Aktuell sind hier nur E-Mailings und Artikel möglich.
>
> **`string name`** The name to be given the object

Querying process status

Deklaration `int getProcessInfo(struct auth, int process_id)`

Funktion Returns the current status of a process

Rückgabewert `int status` with following options:

- **0** Job is in the queue. Waiting for processing.
- **1** Currently processing job.
- **2** Job processed successfully.
- **3** Temporary error. Reinitiating job.
- **4** Permanent error. Job finished.

Parameter

- **`int process_id`** ID of the process

List of all available clients

Deklaration `struct getMandators(struct auth)`

Funktion Returns a list with all clients that the XML-RPC User (auth) has access to.

Rückgabewert `struct` with the `mandator_id` as key to a `struct` with the keys `id` and `name` of the client.

Profile

Search profile

Deklaration `struct doProfileFindBy(struct auth, int pool_id, struct search)`

Funktion Search for a profile with the aid of a search pattern.

Rückgabewert `struct` with the `profile_id` as key to a `struct`, which contains the attributes of the profile as keys resolved as a string.

Parameter

- **`int pool_id`** ID of pool
- **`struct search`** Keys and values as strings that should be searched for; several keys are linked with `and`

Beispiel

```
server.doProfileFindBy(
        dict(username='user', password='pw'),
        12345,
        dict(
                EMAIL='test@example.com',
        )
)
```

All modified profiles of a pool

Deklaration `struct getModifiedProfilesByPool(struct auth, int pool_id, int since_ts=0)`

Funktion Returns a list of IDs with modification time

Rückgabewert `struct` with the `profile_id` as key to the timestamp as `int`.

Parameter

`int pool_id` ID of pool

`int since_ts` Optional time from when the list should be delivered.

Create or update profile

Deklaration `int doProfileUpdate(struct auth, int pool_id, int profile_id, struct profile_data, int permission=3)`

Funktion Update of initial generation of a profile in the pool

Rückgabewert Profile ID that was generated or edited

Parameter

`int pool_id` ID of pool

`int profile_id` ID of the profile that should be updated or if <= 0, then a new profile is created.

`struct profile_data` Keys to the fields that are to be updated in the profile. At the same time, the keys refer to the internal names of the pool attributes. The values must be transferred according to the pool attribute's type. In doing so, the transfer as <title_reference>'UNIX time'</title_reference> (seconds since the beginning of the cycle) applies to the data fields in particular. Pool attributes with options are specified either as an index or as a list of option ID indices. Bool fields must be filled with 1 or 0.

`int permission=3` Optional permission that is to be set for the profile. Default is `OptOut`

Beispiel

```
server.doProfileUpdate(
        dict(username='user', password='pw'),
        12345,
        6789,
        dict(
             EMAIL='test@example.com',
             COMPANY='Example Inc.',
             P3=1168965880,
             P5=(101,102,104),
        )
)
```

Creating or updating several profiles

Deklaration `struct doProfileMassUpdate(struct auth, int pool_id, string search_key, struct profile_data_list, int ignore_missing=0)` `struct doProfileMassMerge(struct auth, int pool_id, string search_key, struct profile_data_list, int ignore_missing=0)`

Funktion Update or merge a list of profiles with the option for initial generation. During the merge, the fields such as multiple selections are attached to the existing values instead of overwriting them.

Rückgabewert List of the search values on the profiles concerned. "OK", or in the event of an error, a message is displayed for each profile.

Parameter

`int pool_id` ID of pool

`string search_key` Name of the pool field where the search is to be made.

`struct profile_data_list` List of search value on a `struct pool_data` (s. `doProfileUpdate`)

`int ignore_missing=0` If no profile is found, a new profile is created by default. If set to `True`, these profiles will be ignored.

Delete profile

Deklaration `boolean doProfileDelete(struct auth, int pool_id, int profile_id)`

Funktion Complete deletion of a profile

Rückgabewert `TRUE`

Parameter

`int pool_id` ID of pool

`int profile_id` ID of the profile

Deleting several profiles

Deklaration `boolean doProfileMassDelete(struct auth, int pool_id, int[] profile_ids)`

Funktion Complete deletion of a list of profiles

Rückgabewert `TRUE`

Parameter

`int pool_id` ID of pool

`int[] profile_id` List of IDs of the profiles to be deleted

Deleting profiles of a target group

Deklaration `int doProfileDeleteByTargetgroup(struct auth, int targetgroup_id)`

Funktion Triggers the deletion of profiles of a target group

Rückgabewert `int process_id`: See getProcessInfo

Parameter

`int targetgroup_id` Target Group ID

Unsubscribe profile

Deklaration `boolean doProfileRevokePermission(struct auth, int pool_id, int profile_id)`

Funktion Reset the profile's permission

Rückgabewert `TRUE`

Parameter

`int pool_id` ID of pool

`int profile_id` ID of the profile

Confirm profile

Deklaration `boolean doProfileGrantPermission(struct auth, int pool_id, int profile_id , int permission=2)`

Funktion Set the profile's permission

Rückgabewert `TRUE`

Parameter

`int pool_id` ID of pool

`int profile_id` ID of the profile

`int permission=2` Setting for the permission that is to be set

Read out scores of a profile

Deklaration `struct getScores(struct auth, int pool_id, int profile_id)`

Funktion Read out scores of a profile grouped according to scoring groups

Rückgabewert `struct` with the scoring group ID as the key and a `struct` with the following attributes as the value:

`int profile_id` ID of the profile

`int group_id` ID of the scoring group

`int group_name` Name of the scoring group

`int activity_score` : Sum of all activity scores of the profile in the scoring group

`int profile_score` Sum of all profile scores of the profile in the scoring group

Parameter

`int pool_id` ID of pool

`int profile_id` ID of the profile

Status of one or all of a profile's eMailings

Deklaration `struct getProfileMailingStatus(struct auth, int profile_id, int mailing_id = NULL)`

Funktion Status of one or all eMailings

Rückgabewert `struct` from `struct` throughout all mailings: `mailing_id` -> `struct`; the `struct` contains the keys per profile:

int status State of the delivery

int last_status_change Timestamp of the last status change

string url Absolute URL to an online email, how the recipient has viewed it. N.B.: Access to this URL will enable everything relating to tracking for the user as if the user himself had clicked.

Parameter

int profile_id ID of the profile

int mailing_id ID of a particular eMailing (optional)

Reading out all openings of a mailing

Deklaration `struct getMailingImpressions(struct auth, int mailing_id, int start_timestamp = 0, int end_timestamp = 0)`

Funktion Returns all openings/impressions of a mailing, whether via e-mail HTML, web/mobile, PDF, etc.

Rückgabewert `struct` with the tracking ID as key and the tracking information as `struct` value with the following keys:

int profile_id ID of the profile

int timestamp UTC UNIX timestamp

Parameter

int mailing_id ID of the transmission

int start_timestamp UTC UNIX timestamp from which (including) tracking data shall be searched for. Default value is 0

int end_timestamp UTC UNIX timestamp to which (including) tracking data shall be searched for. If 0 is returned, the current timestamp will be applied

Reading out all clicks of a mailing

Deklaration `struct getMailingClicks(struct auth, int mailing_id, int start_timestamp = 0, int end_timestamp = 0)`

Funktion Returns all clicks of a mailing, whether via e-mail HTML, web/mobile, PDF, etc.

Rückgabewert `struct` with the tracking ID as key and the tracking information as `struct` value with the following keys:

int profile_id ID of the profile

int timestamp UTC UNIX timestamp

int link_id ID of the link for which a click was recorded. If the value `-1` is returned, the `parent_id` should be considered.

int link_type_id ID of the format link. The significations are provided at the end under 'constants'

int parent_id ID of the resource in which the link was clicked; either the mailing ID or the ID of the article reference.

Parameter

`int mailing_id` ID of the transmission

`int start_timestamp` UTC UNIX timestamp from which (including) tracking data shall be searched for. Default value is 0

`int end_timestamp` UTC UNIX timestamp to which (including) tracking data shall be searched for. If 0 is returned, the current timestamp will be applied

Erklärung zu `link_id` und `parent_id`

The``link_id`` only then has a value greater or equal to `0` if it involves a "normal" link. Clicks, for example, on landing page links, PDF generation, voice, etc. always have a `link_id` equal to `-1`.

In this case, the `parent_id` must be used for closer consideration:

- If the `parent_id` is the same as the higher-level `mailing_id`, the click occurred in the mailing, or the mailing was depicted in the `link_type_id` format. This can for example be a click to read aloud, or the PDF generation in the mailing itself.
- If the `parent_id` is not the same as the higher-level `mailing_id`, the click referred to the article with the reference ID equivalent to the `parent_id`, for example by a click on a sharing link of the article.

Tagging a list of profiles with an attribute option

Deklaration `boolean tagProfilesWithOption(struct auth, int tag_pool_attribute_option_id, int primary_key_pool_attribute_id, string[] keys, boolean setProfileEditDate)`

Funktion Sets the specified pool attribute option in all profiles where the value of the specified primary key is found.

Rückgabewert TRUE

Parameter

`int tag_pool_attribute_option_id` ID of the Pool Attribute Option which should be set for the profiles.

`int primary_key_pool_attribute_id` ID of the pool attribute which should be checked for the specified primary keys. If a 0 is specified, then the value of the field PROFILEID of the pool is used.

`string[] keys` List of values for which the primary key field is checked.

`boolean setProfileEditDate` If `true` is indicated here, the change date for all relevant profiles is set to the present

Removing a pool attribute from a list of profiles Identical with `tagProfilesWithOption`:

Deklaration `boolean untagProfilesWithOption(struct auth, int tag_pool_attribute_option_id, int primary_key_pool_attribute_id, string[] keys, boolean setProfileEditDate)`

List of all profiles with a bounce within a certain period

Deklaration `struct getBouncesSince( struct auth, int since_ts, string[] additional_fields, int|int[] mandator_id=0)`

Funktion Provides a list of profile IDs and their mailings in which a hard bounce or soft bounce occurred

Rückgabewert bei einzelner Mandanten Id `struct` with the profile ID as the key to a `struct` with the values:

`bool bounced` `True`, if the profile had its third hard bounce and will not be sent anymore in future

`struct mailings` Mailing ID as key on `HB` (hard bounce) or `SB` (soft bounce)

`string ...` Name and value from the `additional_fields`, thus available in the profile

Rückgabewert bei mehreren Mandanten Ids `struct` with the client ID as the key to a `struct` as in `Rückgabewert bei einzelner Mandanten Id`

Parameter

`int since_ts` Start time of the search (end time is *now*)

`string[] additional_fields` String list of the field names that should also be returned (e.g. primary key from the external system)

`int||nt[] mandator_id=0` If the user which is used cannot be linked to a client, the client must be restricted with this parameter. Alternatively, an array of client IDs can be transferred.

List of all profiles that have unsubscribed within a certain period

Deklaration `struct getUnsubscriptionsSince( struct auth, int|int[] pool_id, int since_ts, string[] additional_fields)`

Funktion Provides a list of profile IDs that unsubscribed within the pool over a specific period

Rückgabewert bei einzelner Pool Id `struct` with the profile ID as the key to a `struct` with the values:

`int ts` Time of unsubscription

`int mailing_id` If the mailing is known, then its ID - otherwise *0*

`string ...` Name and value from the `additional_fields`, thus available in the profile

Rückgabewert bei mehreren Pool Ids `struct` with the pool ID as the key to a `struct` as in `Rückgabewert bei einzelner Pool Id`

Parameter

`int|int[] pool_id` ID of the pool whose profile is to be searched. Alternatively, an array of pool IDs can be transferred.

`int since_ts` Start time of the search (end time is *now*)

`string[] additional_fields` String list of the field names that should also be returned (e.g. primary key from the external system)

List of all profiles that have confirmed double opt-in within a certain period

Deklaration `struct getGrantedPermissionsSince( struct auth, int pool_id, int since_ts, string[] additional_fields)`

Funktion Returns a list of profile IDS that have confirmed double opt-in in the pool within a certain period

Rückgabewert `struct` with the profile ID as key on a `struct` with the values:

`int ts` Time of the confirmation

`string ...` Name and value from the `additional_fields`, thus available in the profile

Parameter

`int pool_id` ID of the pool whose profile should be searched for

`int since_ts` Start time of the search (end time is *now*)

`string[] additional_fields` String list of the field names that should also be returned (e.g. primary key from the external system)

Put an e-mail address or mail domain on the blacklist

Deklaration `bool addToBlacklist(struct auth, string email, string comment = ”, int mandator_id=0)`

Funktion Enter an e-mail address or mail domain into the local or global blacklist

Rückgabewert `int` with the ID of the blacklist entry

Parameter

`string email` E-mail address (such as `mustermann@example.com`) or mail domain (such as `@example.com`)

`string comment` Optional comment such as the reasons for the entry

`int mandator_id` ID of the client in which he blacklist is to be addressed. If the parameter is not transmitted, or is transmitted with the value 0, the global blacklist is used for global API users, and the blacklist of associated clients is used for client users.

Pool

List of all pools

Deklaration `struct getPools(struct auth, int mandator_id=0)`

Funktion List of all pool under the given client

Rückgabewert `pool_id` -> `struct pool_info` with:

`string name` Internal name

Parameter

`int mandator_id=0` If not given, the ID of the user; If it is a global user, an ID *must* be specified.

List of all pool attributes

Deklaration `struct getPoolAttributes(struct auth, int pool_id)`

Funktion List of a pool‘s every pool attribute

Rückgabewert `pool_attribute_id` -> `struct pool_attribute_info` with:

`string name` Internal name

string label Displayed name

boolean mandatory Mandatory field

int type_id Internal ID of this field‘s type

Parameter

int pool_id ID of pool

Specify pool attribute options

Deklaration `struct doPoolAttributeGetOptions(struct auth, int pool_id, int poolattribute_id)`

Funktion List of a pool attribute‘s every option

Rückgabewert `struct` with `poolattributeoption_id` => `string title`

Parameter

int pool_id ID of pool

int poolattribute_id Pool attribute ID

Update pool attribute option

Deklaration `int doPoolAttributeUpdateOption(struct auth, int pool_id, int poolattribute_id, int poolattributeoption_id, string value)`

Funktion Saves a new value for the option or generates a new option.

Rückgabewert ID of the (new) option

Parameter

int pool_id ID of pool

int poolattribute_id Pool attribute ID

int poolattributeoption_id ID of the Pool-Attribute-Option; if specified as `0`, a new option is attached. cp. `doPoolAttributeAddOption`

string value Name that is to receive the option

Extend pool attribute at another option

Deklaration `int doPoolAttributeAddOption(struct auth, int pool_id, int poolattribute_id, string value)`

Funktion A new option is attached to a pool attribute

Rückgabewert ID of the new option

Parameter

int pool_id ID of pool

int poolattribute_id Pool attribute ID

string value Name that is to receive the option

Delete pool attribute option

Deklaration `boolean doPoolAttributeDeleteOption(struct auth, int pool_id, int poolattribute_id, int poolattributeoption_id)`

Funktion Deletes the specified option - to the extent this is possible.

Rückgabewert `TRUE`, if the option was deleted- `FALSE`, if that is not possible, for example, due to dependencies on target groups.

Parameter

int pool_id ID of pool

int poolattribute_id Pool attribute ID

int poolattributeoption_id ID of the pool attribute option

Attach pool attribute option in a form

Deklaration `boolean doPoolAttributeAddOptionToForm(struct auth, int pool_id, int poolattribute_id, int poolattributeoption_id, int form_id)`

Funktion Said option is attached in the form at the end of the present options.

Rückgabewert `TRUE`, if the option was attached successfully - `FALSE`, if the option already exists in the form.

Parameter

int pool_id ID of pool

int poolattribute_id Pool attribute ID

int poolattributeoption_id ID of the pool attribute option

int form_id ID of the form

Remove pool attribute option off a form

Deklaration `boolean doPoolAttributeRemoveOptionFromForm(struct auth, int pool_id, int poolattribute_id, int poolattributeoption_id, int form_id)`

Funktion Deletes the named option off the form.

Rückgabewert `TRUE`, if the option was removed successfully - `FALSE`, if the option is already missing in the form.

Parameter

int pool_id ID of pool

int poolattribute_id Pool attribute ID

int poolattributeoption_id ID of the pool attribute option

int form_id ID of the form

Profiles of a pool

Deklaration `struct getProfilesByPool(struct auth, int pool_id)`

Funktion Returns a list of all profiles in a pool

Rückgabewert `profile_id` -> `struct profile`

Parameter

`int pool_id` ID of pool

Creating a pool attribute

Deklaration `boolean createPoolAttribute(struct auth, int pool_id, string name, string label, int type_id)`

Funktion Creates a new pool attribute

Rückgabewert `int attribute_id` ID of the newly created pool attribute

Parameter

`int pool_id` ID of pool

`string name` Name of the attribute. Must begin with a capital letter (A-Z) and may only contain capital letters (A-Z) and numbers.

`string label` Label (display name) of the attribute

`int type_id` Type ID of the attribute

Form

List of all forms

Deklaration `struct getForms(struct auth, int mandator_id=0)`

Funktion List of all forms under the given client

Rückgabewert `form_id` -> `struct form_info` with:

`string name` Internal name

`string comment` Description

`string url` URL to the form

Parameter

`int mandator_id=0` If not given, the ID of the user; If it is a global user, an ID *must* be specified.

List of a form's every form alias

Deklaration `struct getFormAliases(struct auth, int form_id)`

Funktion List of a form's every form alias via the assigned form ID

Rückgabewert `int formalias_id` -> `struct` with:

`string name` Internal name

`string comment` Description

string url URL to Resource

Parameter

int form_id ID of form

ID of the form from which the alias was derived

Deklaration `struct getFormAliasAliasedId(struct auth, int formalias_id)`

Funktion ID of the form derived of the specified alias

Rückgabewert `int form_id`

Parameter

int formalias_id ID of a form alias

Upload of an individual template for a form

Deklaration `struct doUpdateFormTemplate(struct auth, int form_id, string source)`

Funktion Updating the source codes of the form of an individual template

Rückgabewert `bool` - `True`, if successful

Parameter

int form_id ID of form

string source Valid XML with UTF-8 encoding

Statistics of a form (or form alias)

Deklaration `struct getFormStatistic(struct auth, string type, int form_id, boolean with_formaliases)`

Funktion List of all statistical data of a form via the transferred form ID

Rückgabewert `struct` with:

int id ID of the form

int name Name of the form

int successcount Number of successful registrations

int impressioncount Number of calls of a form

int identityerrorcount Number of calls of a (pre-filled) form that requires a valid pool_id for opening and has received an invalid pool_id.

int duplicationerrorcount Number of cases where a user has registered repeatedly with one email address using one form

int mandatoryerrorcount Number of cases where a mandatory field was not filled in when sending the form

int validationerrorcount Number of cases where a field had an invalid value when sending the form

struct links[id] = struct mit

int id The ID of the link

int clicks Number of clicks on a link

string url URL of the link

struct aliases[id] = struct List of aliases

Aliase

int id ID of the form

int name Name of the form

int successcount Number of successful registrations

int impressioncount Number of calls of a form

int identityerrorcount Number of calls of a (pre-filled) form that requires a valid pool_id for opening and has received an invalid pool_id.

int duplicationerrorcount Number of cases where a user has registered repeatedly with one email address using one form

int mandatoryerrorcount Number of cases where a mandatory field was not filled in when sending the form

int validationerrorcount Number of cases where a field had an invalid value when sending the form

Parameter

string type "Form" or "FormAlias"

int form_id ID of a form (or form alias)

boolean with_formaliases Optional: Additional return of the assigned aliases statistics

Checkpoint

Statistics of a checkpoint

Deklaration `struct getCheckpointStatistic(struct auth, int checkpoint_id)`

Funktion List of all statistical data of a checkpoint via the transferred checkpoint ID

Rückgabewert `struct` with:

int clicks Number of calls via any clicked link

int uniq_clicks Number of different persons who have clicked any link

int visits Displaying the number of tracking images

int uniq_visits Number of different users who have loaded a tracking image

struct links[id] = struct mit

int id The ID of the link

int clicks Number of calls of the link

int uniq_clicks Number of different users who have clicked on the link

string url The Url of the link

Parameter

int checkpoint_id ID of the checkpoint

Targetgroup

List of all target groups

Deklaration `struct getTargetGroups(struct auth, mandator_id=0)`

Funktion List of all target groups under the given client

Rückgabewert `int targetgroup_id` -> `struct` with:

string name Internal name of the target group

Parameter

int mandator_id=0 If not given, the ID of the user; If it is a global user, an ID *must* be specified.

Target group information

Deklaration `struct getTargetgroupInformation(struct auth, int targetgroup_id)`

Funktion Information about the target group

Rückgabewert `struct targetgroup_information` -> `struct` with:

string name Name of the target group

int profile_count Number of profiles of the target group at the time of the call

Parameter

int targetgroup_id ID of the target group

Profiles of a Target Group

Deklaration `struct getProfilesByTargetGroup(struct auth, int target_group_id)`

Funktion Returns a list of every profile in a particular specified target group.

Rückgabewert `profile_id` -> `struct profile`

Parameter

int target_group_id Target Group ID

Check whether a profile is in a target group

Deklaration `struct isProfileInTargetGroup(struct auth, int pool_id, int profile_id, id[] target_group_ids)`

Funktion Checking a profile for its presence in one or more target groups.

Rückgabewert `struct` with `target_group_id` -> `boolean`, in which `TRUE` represents an affiliation to the target group

Parameter

int pool_id ID of pool

int profile_id ID of the profile

id[] target_group_ids List if target group IDs

Alternatively, the call can be made just for one target group:

Deklaration `boolean isProfileInTargetGroup(struct auth, int pool_id, int profile_id, int target_group_id)`

Rückgabewert `TRUE`, if the entry is in the target group

Parameter

int target_group_id Target Group ID

Create target group by means of pool attribute option

Deklaration `int generateTargetGroupFromSelectionOption(struct auth, string name, int pool_id, int pool_attribute_id, int pool_attribute_option_id, int category_id=0)`

Funktion Creation of a new target group with the specified name and pool attribute option. Single and multiple selections are supported.

Rückgabewert ID of the new target group

Parameter

string name Name of the new target group in the interface

int pool_id ID of pool

int pool_attribute_id Pool attribute ID

int pool_attribute_option_id ID of the pool attribute option

int category_id Optional information of a target folder. If nothing is specified, the new object is created in the standard folder for target groups.

Reading out standard target group folder ID of a client

Deklaration `int getTargetGroupFolderIdByCustomerId(struct auth, int mandator_id)`

Funktion Return of the category ID of the standard target group folder of the client

Rückgabewert Category ID of the target group folder

Parameter

int mandator_id ID of the client

Emailing

List of all eMailings

Deklaration `struct getMailings(struct auth, int mandator_id=0)`

Funktion List of all mailings below the specified client (transmissions)

Rückgabewert `mailing_id` -> `struct mailing_info` with:

int timestamp UNIX timestamp for UTC with transmission starting point

int recipients Number of recipients

string name Internal name of the eMailing

boolean sent Emailing sent?

string preview_url Absolute URL in the preview

string admin_url Absolute URL to the eMailing in the admin interface

string report_url Absolute URL to statistics

string subject Subject line

Parameter

int mandator_id=0 If not given, the ID of the user; If it is a global user, an ID *must* be specified.

Emailing Status

Deklaration `struct getMailingStatus(struct auth, int mailing_id, boolean verbose=False, int timeframe=0)`

Funktion Returns the current state of every profile off the given eMailing

Rückgabewert `struct` from `struct` across all profiles: `profile_id` -> `struct`; the `struct` contains the keys per profile:

int status State of the delivery

int last_status_change Timestamp of the last status change

If `verbose==True`:

string url Absolute URL to an online email, how the recipient has viewed it. N.B.: Access to this URL will enable everything relating to tracking for the user as if the user himself had clicked.

struct profile Complete Profile Data (compare Profile Functions)

Parameter

int mailing_id Emailing ID

boolean verbose=False Extended result with complete profile data and additional information that refers to the profile if `True`

int timeframe=0 When specified, only the number of seconds in the past is considered for the profiles detected

Transmission of a mailing to a target group

Deklaration `int sendMailingToTargetGroup(struct auth, int mailing_id, int target_group_id, int start_ts=0)`

Funktion Starts the transmission of the eMailing to the target group. If the start time is set, the transmission then starts at this time.

Rückgabewert ID of the transmission or 0 in the event of an error

Parameter

int mailing_id Emailing ID

int target_group_id Target Group ID

int start_ts Optional time of the transmission

int speed Optional transmission speed in mails per hour

Send eMailing to profile

Deklaration `boolean doSendEmailing(struct auth, int profile_id | int[] profile_ids, int emailing_id)`

Funktion Transmits the eMailing with the ID *emailing_id* and profile with the ID *profile_id*

Rückgabewert `TRUE` if the transmission was initiated.

Parameter

int profile_id | int[] profile_ids One or more IDs of profiles

int emailing_id ID of the eMailings

Generating mailing from template

Deklaration `int createMailingDraft(struct auth, int mailing_template_id, string name, int category_id=0, int mandator_id=0, string subject=null)`

Funktion Generated in new eMailing from the template specified.

Rückgabewert ID of the generated mailing

Parameter

int mailing_template_id ID of the template

string name Name of the new mailing

int category_id Optional information of a target folder. If nothing is specified, the new object is created in the standard folder for mailings.

int mandator_id=0 ID of the client under which the mailing is created. Only necessary if the user is not created at client level and no explicit category_id was specified.

string subject=null Optional information of a subject line

Reading the subject line configuration of a mailing

Deklaration `struct getMailingSubjects(struct auth, int mailing_id)`

Rückgabewert Struct with target group as key and subject line as value. :`int mailing_id`: Mailing ID

Setting the subject line configuration of a mailing

Deklaration `boolean setMailingDraftSubjects(struct auth, int mailing_id, struct subjects)`

Rückgabewert True :`int mailing_id`: ID of the mailing :`struct subjects`: Struct with target group ID as the key, and subject line as the value. The standard subject line is filled with the target ID `0`.

Remove article from a mailing

Deklaration `boolean removeArticle(struct auth, int mailing_id, int article_reference_id)`

Rückgabewert `True` if successful, otherwise the method returns a corresponding fault

Parameter `:int mailing_id` : ID of the mailing `:int article_reference_id` : ID of the article reference in the mailing

List of all mailing templates

Deklaration `struct getMailingTemplates(struct auth, int mandator_id=0)`

Funktion List of all active templates for mailings in the system

Rückgabewert `struct`: (see the standard object lists)

Parameter

`int mandator_id` Optional client ID

List of all mailing drafts

Deklaration `struct getSendableMailingDrafts(struct auth, boolean unsent=false, int mandator_id=0, int timestamp=0)`

Funktion List of all mailing drafts below the specified client

Rückgabewert `mailing_id` -> `struct mailing_info` with:

`int id` ID of the mailing draft

`string name` Name of the mailing draft

`string url` Displaying URL of the mailing draft

`string admin_url` URL for direct editing.

`string comment` Comment

Parameter

`boolean unsent=false` If not given, all drafts are determined. If unsent true, only drafts will be returned that have not yet been sent.

`int mandator_id=0` If not given, the ID of the user; If it is a global user, an ID *must* be specified.

`int timestamp=0` Unix Timestamp (UTC) from which modification date the drafts should be returned.

Selective list of mailings

Deklaration `struct getMailingsByCriteria(struct auth, struct filter, mandator_id=0)`

Funktion List of all mailings below the specified client using a filter

Rückgabewert `mailing_id` -> `struct mailing_info` (s. getMailings)

Parameter

`struct filter`

int target_group_id Only mailings to this target group

int younger_than Only mailings younger than this timestamp

int older_than Only mailings older than this timestamp

More than one different criterion is allowed and the result is determined as an AND logic operation of all criteria.

int mandator_id=0 If not given, the ID of the user; If it is a global user, an ID *must* be specified.

Mailing statistics

Deklaration `struct getMailingStatistic(struct auth, int mailing_id)`

Funktion Provides statistical information on a mailing or an event mailing (basis, media, articles and links)

Rückgabewert `struct`

Parameter

int mailing_id Emailing ID

Profiles of the mailing statistics

Deklaration `struct getMailingImpressionProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingMultipleImpressionProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingClickProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingMultipleClickProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingRecipientsProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingBouncesProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingHardBouncesProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingSoftBouncesProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingUnsubscriptionsProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingAllArticlesProfileList(struct auth, int mailing_id, string[] fields = NULL)`

`struct getMailingArticleProfileList(struct auth, int mailing_id, int article_id, string[] fields = NULL)`

`struct getMailingAllLinkClicksProfileList(struct auth, int mailing_id, string[] fields = NULL)` `struct getMailingLinkClickProfileList(struct auth, int mailing_id, int link_id, string[] fields = NULL)`

Funktion Returns the profiles that one of the named functions in the eMailing has used.

Rückgabewert `profile_id` -> `struct profile`

Parameter

- **`int mailing_id`** Emailing ID
- **`string[] fields=NULL`** Optional list of field names that should be exported. If nothing is specified, all fields are then exported.
- **`int article_id`** Optional according to function: ID of an article. Compare results of getMailingStatistic
- **`int link_id`** Optional according to function: ID of a link. Compare results of getMailingStatistic

Article

Creating or updating articles

Deklaration `struct createArticle(struct auth, int article_id, int article_preset_id, string name, struct article_data, int category_id=0)`

Funktion Initial generation or update of an article

Rückgabewert ID of the article that was created or edited

Parameter

- **`int article_id`** ID of the article that should be updated or be 0 if a new article is to be created.
- **`int article_preset_id`** ID of the article type. Required only for initial generation.
- **`string name`** Name of the article.
- **`struct article_data`** Keys on the fields that should be updated in the article. The keys refer to the internal names of the article attributes. The values must be transferred according to the type of article attribute. The transfer as *UNIX* fields applies in particular to date fields (seconds since the beginning of the Epoch). Article attributes with options are either specified as an index or list of indices of the option IDs
- **`int category_id`** Optional information of a target folder. If nothing is specified, the new object is created in the standard folder for articles.

Beispiel

```
server.createArticle(
        dict(username='user', password='pw'),
        0,
        12345,
        'Artikelname'
        dict(
                HEADLINE='Mein Artikel',
                SHORTTEXT='Mein Kurztext',
                LONGTEXT='Mein Langtext',
                ...
```

```
            )
    )
```

Querying article data

Deklaration `struct getArticleData(struct auth, int article_id)`

Funktion Querying the data of an article

Rückgabewert `struct`

Parameter

`int article_id` ID of the article whose data should queried

Reading out landing page link of the article

Deklaration `struct getArticleLandingpageLink(struct auth, int article_id)`

Funktion Querying the landing page link of the article

Rückgabewert `string`

Parameter

`int article_id` ID of the article whose data should queried

`string landingpage_url` URL of the landing page

Setting landing page link of the article

Deklaration `struct setArticleLandingpageLink(struct auth, int article_id, string landingpage_url)`

Funktion Setting the landing page link of the article

Rückgabewert `boolean` Always True

Parameter

`int article_id` ID of the article whose data should queried

List of all article types

Deklaration `struct getArticleTypes(struct auth, mandator_id=0)`

Funktion List of all article types below the specified client

Rückgabewert `int articletype_id` -> `struct` with:

`string name` Internal name of the article type

Parameter

`int mandator_id=0` If not given, the ID of the user; If it is a global user, an ID *must* be specified.

Container

List of all container types

Deklaration `struct getContainerTypes(struct auth, mandator_id=0)`

Funktion List of all container types below the specified client

Rückgabewert `int articletypes_id` -> `struct` with:

> **`string name`** Internal name of the container type

Parameter

> **`int mandator_id=0`** If not given, the ID of the user; If it is a global user, an ID *must* be specified.

Report

Generation of a new report

Deklaration `int generateReport(struct auth, string name, int category_id)`

Funktion Generates a new report in the folder and returns the ID of the object

Rückgabewert ID of the new object

Parameter

> **`string name`** Object name of the new object
>
> **`int category_id`** ID of the folder where the object is to be generated

Reading out report folder ID of a client

Deklaration `int getReportFolderIdByCustomerId(struct auth, int mandator_id)`

Funktion Return of the category ID of the standard report folder of the client

Rückgabewert Category ID of the report folder

Parameter

> **`int mandator_id`** ID of the client

Pool Data Miner

Reading out information about a pool data miner

Deklaration `int getPoolDataMinerStatistic(struct auth, int pooldataminer_id)`

Funktion Returns information about the Pool Data Miner and current statistic

Rückgabewert `struct` with: :`string name`: Internal name :`string comment`: Description :`string url`: URL on resource :`string admin_url`: URL on edit mask of the resource :`struct data`: statistics of the Pool Data Miner

Parameter

> **`int pooldataminer_id`** ID of the Pool Data Miner

Image

Creation of a new image

Deklaration `int createImage(struct auth, string image, string name, int category_id=0)`

Funktion Creates a new image in the folder

Rückgabewert `int image_id -> struct` with: `:string name:` Internal name `:string comment`: Description `:string url`: URL on resource `:string admin_url`: URL on edit mask of the resource

Parameter

`string image` The image as base64 encoded string

`string name` The name of the image

`int category_id` Optional ID of the folder where the object is to be created. If nothing is specified, the new object is created in the standard folder for images.

Querying an image

Deklaration `int getImage(struct auth, int image_id)`

Funktion Returns statistical information about an image

Rückgabewert `int image_id -> struct` with: `:string name:` Internal name `:string comment`: Description `:string url`: URL on resource `:string admin_url`: URL on edit mask of the resource

Parameter

`int image_id` ID of the image

Querying all images of a folder

Deklaration `int getImageList(struct auth, int category_id)`

Funktion Returns all images of a folder

Rückgabewert `int image_id` -> `string name`

Parameter

`int category_id` ID of the folder

Reading out image folder ID of a client

Deklaration `int getImageFolderIdByCustomerId(struct auth, int mandator_id)`

Funktion Return of the category ID of the standard image folder of the client

Rückgabewert Category ID of the image folder

Parameter

`int mandator_id` ID of the client

SmartLink

List of all SmartLinks

Deklaration `struct getSmartLinks(struct auth, int mandator_id=0)`

Funktion Returns a list with all SmartLinks.

Parameter

`int mandator_id` Optional ID of the mandator in which to look. If the ID is 0 or not specified, the mandator in which the user is located will be used.

List of the links of a SmartLink

Deklaration `struct getSmartlinkLinks(struct auth, int smartlink_id)`

Funktion Returns a list of all links of a SmartLink

Parameter

`int smartlink_id` Id of the SmartLink

Complete statistics of a SmartLink

Deklaration `struct getSmartlinkStatistics(struct auth, int smartlink_id)`

Funktion Provides a list of all the links of a SmartLink including statistics

Parameter

`int smartlink_id` Id of the SmartLink

List of all link targets of a link within a SmartLink

Deklaration `struct getSmartlinkLinkTargets(struct auth, int smartlink_id, int smartlink_link_id)`

Funktion Returns the list of all the link targets of a link of the SmartLink

Parameter

`int smartlink_id` Id of the SmartLink

`int smartlink_link_id` Id of the link within the SmartLink

Statistics of link of the smartlink

Deklaration `struct getSmartlinkLinkStatistic(struct auth, int smartlink_id, int smartlink_link_id)`

Funktion Returns clicks and unique clicks on a link of a SmartLink

Parameter

`int smartlink_id` Id of the SmartLink

`int smartlink_link_id` Id of the link within the SmartLink

Statistics of all link targets of a link within a SmartLink

Deklaration `struct getSmartlinkLinkTargetStatistics(struct auth, int smartlink_id, int smartlink_link_id)`

Funktion Returns the statistics of all link targets of a link within the SmartLink

Parameter

int smartlink_id Id of the SmartLink

int smartlink_link_id Id of the link within the SmartLink

Profile list of all clicks on a link of the SmartLink

Deklaration `struct getSmartlinkLinkClickProfiles(struct auth, int smartlink_id)`

Funktion Returns profile Id and timestamp of all clicks on a link of a SmartLink.

Parameter

int smartlink_id Id of the SmartLink

Creating link in a SmartLink

Deklaration `struct createSmartlinkLink(struct auth, int smartlink_id, string link_name)`

Funktion Creates a new link in an existing SmartLink

Parameter

int smartlink_id Id of the SmartLink link

string link_name Name of the link

Rückgabewert

Array mit folgenden Keys

int link_id Id of the SmartLink link

string url URL of the SmartLink link for integration

Creating target in a SmartLink link

Deklaration `struct createSmartlinkLinkTarget(struct auth, int smartlink_id, int link_id, string url, int targetgroup_id=0)`

Funktion Creates a new target within a SmartLink link

Parameter

int smartlink_id Id of the SmartLink link

string link_id Name of the link

string url URL of the target

int targetgroup_id ID of the target group for the target. If set, the link only transfers to this target when the profile is in this target group.

Rückgabewert

int target_id ID of the target

Constants

Profile permission:

0 Unconfirmed

1 Single Opt-In

2 Double opt-in confirmed

3 Opt-Out (e.g., from Import)

State of the delivery in the course of an eMailing:

0 Queue

1 Delivered

2 Hardbounce

3 Softbounce

Internal ID of the type of this field:

1 Salutation

2 Country

3 State

4 Mobile network

5 Input

6 Multiline input

7 Selection

8 Multiple selection

9 Date

10 eMail

11 Datetime

12 Language

13 Profile ID

14 Datetime read only

15 Permission

16 Boolean

17 Hard bounces

18 User set state

19 Form

20 Int

Type ID for creation of pool attributes

1 Salutation

2 Country

3 State

4 Mobile network

5 Input

6 Multiline input

7 Selection

8 Multiple selection

9 Date

10 eMail

11 Datetime

12 Language

16 Boolean

20 Int

Link type ID for mailing click tracking

0 Unknown

1 Online view of a newsletter

2 PDF view of an article or newsletter

3 Text view of the newsletter

4 Web/Mobile view of the newsletter

5 Landing page of an article

6 Click on landing page link

7 LeadPage Opening

11 XSL-FO view of an article or newsletter

12 Display the RSS feed of a newsletter

19 Open newsletter

20 External link

21 Display the ATOM feed of a newsletter

22 Call preview service for an article or newsletter

23 Call the article or newsletter via the preview service

24 Call preview service for a newsletter podcast

25 Call the podcast data via the preview service

40 Article shared in Facebook

41 Article shared in Google Bookmarks

42 Article shared in MySpace

43 Article shared in del.icio.us

44 Article shared in Twitter

45 Article shared in Xing

46 Article shared in LinkedIn

47 Article shared in Google+

50 Click on article shared in Facebook

51 Click on article shared in Google Bookmarks

52 Click on article shared in MySpace

53 Click on article shared in del.icio.us

54 Click on article shared in Twitter

55 Click on article shared in Xing

59 Click on article shared in LinkedIn

60 Click on article shared in Google+

5.1.2 API Connectors for PHP

In order to simplify the connection of the system via API in PHP projects or applications, a PHP connector is offered. The connector serves to create an abstraction layer between the API methods and the system functions, in which updates can be made without having to change the own implementation.

The connector is open source and hosted on Github, which makes an installation or update very easy:

Reporting API Connector for PHP <https://github.com/SC-Networks/evalanche-reporting-api-connector>'_

SOAP API Connector for PHP <https://github.com/SC-Networks/evalanche-soap-api-connector>'_

5.2 FAQ for developers

Here, you can find frequently asked questions of web designers or developers. The answers assume technical expertise in **HTML**, **CSS** and **JavaScript**.

5.2.1 eMailing

I have written the style tag in the body tag. After saving, the style block is in the head of the template again.

Make the following changes on the template or mailing:

- Extend the style tag by adding the attribute `metal:define-macro="style"`
- Insert `<div metal:use-macro="style"> </div>` immediately after the introductory body tag

5.2.2 Form

When entering a profile, how can I also entered hidden information for a target group?

Make the following settings in the form:

- Attach the corresponding attribute for the target group to the form. Preferably, it should be a multiple selection.

- Remove all options except for the one required. This ensures that no other previously set options of this attribute are overwritten.
- Set *Hidden entry* for the display.
- Select the option as default.

How can I change from a form in an iframe to a view of the whole page?

Either you specify a `target` in the `form` tag of the source form. E.G.:

```
<form target="_top" action="" method="post">
```

or you use a JavaScript on the target form which ensures that the Full Screen page is reloaded in the `onload`. E.G.:

```
<body onload="if (top != self) top.location.href = location.href;">
```

The disadvantage of the `target` variant lies in the fact that when using an error whose validation can fail, then the whole form comes full-screen and you should display content accordingly in the event of an error so that the user is not fighting a losing battle.

How can I use my own form in an entry editor?

It is often desirable to use the entry editor only with your own form for an internal application or front desks. To do this, you can simply use the URL of the entry editor from the pool. Then attach `,form_id=<ID>` to the URL. You can get the *<ID>* of the form from the overview list of a form.

How can I design a form like the system interface.

Include the system CSS either with a standard form in the entry *Own CSS* or in your template via `@import url(https://IhreDomain.com/styles.php);`

How can I configure the data from the system directly under one URL of my own domain.

If you have control over the Rewrite-Rules on the customer‘s server, you can then also display all URLs of the system under the URL of the customer. Configure the following rules in the web server (example for Apache2):

```
RewriteRule ^art_resource.php$ http://<Domain des Systems>/art_resource.php [P]
RewriteRule ^a.php$            http://<Domain des Systems>/a.php            [P]
RewriteRule ^goto.php$         http://<Domain des Systems>/goto.php         [P]
RewriteRule ^olt.php$          http://<Domain des Systems>/olt.php          [P]
```

5.3 Programming in templates

Developers can use the following programming examples in TAL in order to integrate individual newsletter functions in newsletter templates and PDF newsletter templates. It is not possible to use these examples in text newsletter templates. In these instances you have to create your own replacement variables. If required, please consult with our Service & Support staff.

5.3.1 Newsletter

Table of contents (anchor link to a slot's every article)

Default table of contents for all articles in slot 0.

```
Inhaltsverzeichnis:
<ul tal:define="slot THIS/getNewsletter/getSlot0">
  <li tal:repeat="article slot/getArticles">
    <a href="#" tal:attributes="href article/getArticle/getAnchorHRef"
      tal:content="article/getArticle/getHeadline">Artikel...</a>
  </li>
</ul>
```

Must be in the article template as link destination then

```
<a name="atruntime" tal:attributes="name THIS/getAnchor">{$HEADLINE}</a>
```

Link to landing page - however only if there is a long text as well

"more..." Link in the article template which is only displayed if a long text exists in the article.

```
<div tal:condition="THIS/hasLongText">
  <a href="{$READ_LP_URL}">mehr...</a>
</div>
```

Display an image in an article if there is an image.

Displaying an image in the newsletter article - however only if one is even available.

```
<img border="0" src="{$IMG_URL}" tal:condition="THIS/hasImage" />
```

Display image in the article and link to landing page

Displaying an image in the newsletter article - however only if one is even available. Linking the image to the landing page or landing page link - however only if available.

```
<a href="#" tal:omit-tag="not:THIS/hasLandingPage" tal:attributes="href THIS/getReadLpLinkOrLpUrl">
  <img border="0" src="{$IMG_URL}" tal:condition="THIS/hasImage" />
</a>
```

Dynamic image formats

Dynamic generation of images in specified image formats:

Optional image attributes must be entered after `{IMG_URL}` separated by ","(comma). The default setting without parameter is: image format jpg with 82% compression. e.g:

```
{$IMG_URL},width=160
{$IMG_URL},format=png,width=120
```

The following options are known:

width=<pixel> The image is not displayed wider than `<pixel>`.

height=<pixel> The image is not displayed higher than `<pixel>`.

top=<pixel> The image starts from the top first from `<pixel>`. This operation is applied after scaling.

left=<pixel> The image starts from the left first from `<pixel>`. This operation is applied after scaling.

cutwidth=<pixel> The image is cropped in width after `<pixel>`. This operation is applied after scaling.

cutheight=<pixel> The image is cropped in height after `<pixel>`. This operation is applied after scaling.

cutfm=1 If a width and/or height is specified for cropping, the cut is made from the centre of the image.

rcrad=<pixel> The corners of the image are rounded by `<pixel>` after scaling and cropping. This operation is applied after scaling. Also see `rccol` for the filling background color.

rccol=<hex> Color that fills the background of the rounded corner. The color is entered as a 6 hex-number without `#`. For example: 99ff99

format=(png|gif|jpeg) Output occurs in the specified format.

dither=1 Output occurs dithered.

monochrome=1 Output occurs as monochrome.

grayscale=1 Output occurs as a grayscale image

colors=(3-256) Color reduction to the specified quantity (only with `format=gif`).

negate=1 Output occurs intertiated.

embed=1 Integrates images directly into the eMailing. The image must not be reloaded, however the email size will increase.

Tip: An image is always scaled proportionally; if you have specified `width` and `height`, `width` will then have priority.

Attention: You have to specify a format in the case of color and size transformations!

Link list in the article

Display of every link that is contained in the article (as a link list).

```
<p tal:condition="ARTIKEL/getLinks"> Weitere Links:
  <span tal:repeat="link ARTIKEL/getLinks">
    <a tal:attributes="href link/getUrl" tal:content="link/getText">Links zu diesem Artikel</a>
    <span tal:condition="not:repeat/link/end">, </span>
  </span>
</p>
```

Set the first article link to the image

For a graphic teaser, one can use the image, for example, and the first article link.

```
<a tal:condition="THIS/hasLinks" tal:attributes="href THIS/getLinks/0/getUrl; title THIS/getLinks/0/g
  <img border="0" alt="" src="{$IMG_URL}" />
</a>
<img tal:condition="not: THIS/hasLinks" border="0" alt="" src="{$IMG_URL}" />
```

Access to the profile data from TAL

Salutation once as string and once as ID.

```
<span tal:replace="THIS/getCurrentUser/getValues/SALUTATION/value">...</span>
<span tal:replace="THIS/getCurrentUser/getValuesRaw/SALUTATION/value">...</span>
```

Individual salutation

Avoid the salutation *Dear Mr Sampleman*

If (e.g. in a newsletter archive) reference is made to the online view of a transmitted mailing, the specification of the personalization variable `{$INDIVIDUAL_SALUTATION}` normally results in the salutation *Dear Mr Sampleman*. To avoid this, the following code section can be used in the mailing template:

```
<!-- Ist der aktuelle User Max Mustermann? -->
<span tal:omit-tag="" tal:condition="not:THIS/getCurrentUser/isDummy">{$INDIVIDUAL_SALUTATION}</span>
<span tal:omit-tag="" tal:condition="THIS/getCurrentUser/isDummy">Sehr geehrte Damen und Herren</span
```

Profile data from TAL for customizing the salutation

Customized salutation via P1 as indicator, whether a personal or a business form is to be selected.

```
<div tal:condition="THIS/getCurrentUser/isDummy">Sehr geehrte Damen und Herren,</div>
<div tal:condition="not:THIS/getCurrentUser/isDummy">
  <div tal:define="salutation THIS/getCurrentUser/getValuesRaw/SALUTATION; p1 THIS/getCurrentUser/get
    <div tal:condition="p1/contains/value/151">
      <div tal:condition="salutation/equals/value/0">
        Servus zusammen
      </div>
      <div tal:condition="salutation/equals/value/1">
        Hi <span tal:replace="THIS/getCurrentUser/getValues/FIRSTNAME/value">Vorname</span>
      </div>
      <div tal:condition="salutation/equals/value/2">
        Hi <span tal:replace="THIS/getCurrentUser/getValues/FIRSTNAME/value">Vorname</span>
      </div>
    </div>
    <div tal:condition="not: p1/contains/value/151">
      <div tal:condition="salutation/equals/value/0">
        Sehr geehrte Damen und Herren
      </div>
      <div tal:condition="salutation/equals/value/1">
        Sehr geehrte Frau <span tal:replace="THIS/getCurrentUser/getValues/NAME/value">Name</span>
      </div>
      <div tal:condition="salutation/equals/value/2">
        Sehr geehrter Herr <span tal:replace="THIS/getCurrentUser/getValues/NAME/value">Name</span>
      </div>
    </div>
```

```
  </div>
</div>
```

Use pre-generated salutation if available.

If, in the LETTERSALUTATION field, a pre-generated salutation is available in the profile (e.g. "Hello, Dear John") this should be used instead of the default salutation.

```
<span tal:omit-tag="" tal:condition="not:exists:THIS/getCurrentUser/getValues/BRIEFANREDE">{$INDIVIDU
<span tal:omit-tag="" tal:condition="exists:THIS/getCurrentUser/getValues/BRIEFANREDE">
  <span tal:omit-tag="" tal:define="BRIEFANREDE THIS/getCurrentUser/getValues/BRIEFANREDE">
    <span tal:omit-tag="" tal:condition="not:BRIEFANREDE/isEmpty/value/dummyvalue" tal:content="BRIEF
    <span tal:omit-tag="" tal:condition="BRIEFANREDE/isEmpty/value/dummyvalue">{$INDIVIDUAL_SALUTATIO
  </span>
</span>
```

Show SLOT only if there is an article.

The query for this is slightly more complex because in the visual edit mode it has to be ensured that the slot is usable.

```
<div tal:omit-tag="" tal:condition="THIS/isAppendMode"><h1>Slot</h1>{$SLOT0}</div>
<div tal:omit-tag="" tal:condition="not:THIS/isAppendMode">
  <div tal:omit-tag="" tal:condition="THIS/getSlot0/hasArticles"><h1>Slot</h1>{$SLOT0}</div>
</div>
```

For this purpose, one ideally uses TAL macros so that one does not have to deal with the slot code twice. However, one can use other code for editing and displaying here as well.

Integrate image in PDF template

Use the following code to insert an image in a PDF template: You are reminded to enter the width and height values:

```
<fo:external-graphic src="url('http://URL_ZUM_BILD')" scaling-method="integer-pixels" width="100px" h
```

Automatically include external XML content in an eMailing

Es ist möglich, externen Content im XML-Format automatisiert in ein Mailing laden zu lassen. Dazu muss zunächst die Datenquelle in der eMailing Konfiguration eingestellt werden. Der Inhalt wird, sofern die Quelle statisch ist, beim Versand des eMailing Objektes an eine Zielgruppe abgeholt und vorgehalten. Sollte es in der Quelle Anzeichen für Individualisierung geben (über die Personalisierungsvariablen) oder es sich um ein Event- oder Trigger-Mail handeln, wird der Inhalt bei jedem Versand neu abgeholt und eingefügt.

Structure of the data source (e.g. an RSS feed)

```
<!-- ... -->
<channel>
  <item>
    <title>Testartikel</title>
    <link>http://www.testlink.de</link>
    <description>Das ist ein Test</description>
  </item>
  <item>
    ...
```

```
  </item>
</channel>
<!-- ... -->
```

Structure of the TAL code in the email template

```
<div tal:condition="THIS/hasExternalXML">
  <table tal:repeat="item THIS/getExternalXML/channel/item[]">
    <tbody>
      <tr>
        <td tal:content="item/title">titel</td>
      </tr>
      <tr>
        <td tal:content="item/description">text</td>
      </tr>
      <tr>
        <td>
          <a href="#" tal:attributes="href item/link">link</a>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

5.3.2 Form

Display every mandatory field in a form

```
<div id="mandatoryerrors" tal:condition="THIS/hasMandatoryErrors">
  Folgende Felder enthalten keine Daten:
  <ul>
    <li tal:repeat="fa THIS/getMandatoryErrorAttributes" tal:content="fa/getLabel">...</li>
  </ul>
</div>
```

Flag mandatory field in the case of an error inline

E.G. Displaying a red star in the name field if it is configured as mandatory but not specified

```
<span style="color: red" tal:condition="THIS/getAttributesByName/NAME/hasMandatoryError">*</span>
```

Default information from the form-generated error if profile already available

```
<div tal:condition="THIS/hasDuplicationErrors">
  <div tal:condition="THIS/isShowDuplicationErrorText">
    <div id="duplicateerrors" tal:content="THIS/getDuplicationErrorText">...</div>
  </div>
</div>
```

Default error information from the form if the validation failed on uniqueness

```
<div tal:condition="THIS/hasIdentityErrors">
  <div tal:condition="THIS/isShowIdentityErrorText">
    <div id="identityerrors" tal:content="THIS/getIdentityErrorText">...</div>
  </div>
</div>
```

Individual salutation

In the form, there is no provision for replacing the variable `{$INDIVIDUAL_SALUTATION}`. Alternatively, the following code snippet can achieve the same result

```
<p tal:define="salutation THIS/getCurrentUser/getValuesRaw/SALUTATION">
  <span tal:condition="salutation/equals/value/1">Sehr geehrte Frau  <span tal:replace="THIS/getCurre
  <span tal:condition="salutation/equals/value/2">Sehr geehrter Herr <span tal:replace="THIS/getCurre
  <span tal:condition="salutation/equals/value/3">Sehr geehrte Damen und Herren,</span>
  <span tal:condition="salutation/equals/value/4">Sehr geehrte Damen und Herren,</span>
  <span tal:condition="salutation/equals/value/0">Sehr geehrte Damen und Herren,</span>
</p>
```

Read container via JavaScript to personalize web pages

Any website can be personalized by loading a container via JS and then using the content via JS in the website. All you have to do is create a container with the desired content and insert its URL into the script.

```
<script type="text/javascript">
    //<![CDATA[
            $(document).ready(function(){
                $.getJSON( "getjson.php", function( data ) { // Holt die Container Inhalte des gewüns

            $('#header > #salutation h1').text(data.HEADLINE); // Schreibt den Inhalt des Container-I

                })
                .done(function() {
            console.log( "second success" );
                    })
                    .fail(function(data) {
                    console.log( "error" );
                    $('body').html('CONTAINER NOT FOUND!');
                    })
                    .always(function() {
                    console.log( "complete" );
                    });

            });
    //]]>
    </script>
```

If the integration URL of a smart link is used instead of the CONTAINER_URL, it is also possible to play out individualized content according to target groups.

> **Attention:** To avoid CORS/CORB problems you need a 2nd file (e.g.: getjson.php) which fetches the content:

```
<?php
$json_form_config = file_get_contents(
        'CONTAINER_URL'
);
print $json_form_config;
```

5.4 Replacement variables

Here, you will find a complete list of all replacement variables in the system.

Hint: Not all variables can be used everywhere. Please note the permitted places of use.

5.4.1 Replacement variables for personalization

{$INDIVIDUAL_SALUTATION} **Individual salutation** - is configured in the basic data of the newsletter

{$SALUTATION} **Salutation**

{$FIRSTNAME} **First name**

{$LASTNAME}, {$NAME} **Last name**

{$COMPANY} **Company**

{$EMAIL} **email address**

{$ADDRESS} **Address**

{$STREETNUMBER} **Street number**

{$ZIPCODE} **Zip Code**

{$CITY} **City**

{$STATE} **State**

{$COUNTRY} **Country**

{$PHONEPREFIX} **Telephone area code**

{$PHONENUMBER} **Telephone number**

{$FAXPREFIX} **Fax area code**

{$FAXNUMBER} **Fax number**

{$MOBILENETWORK} **Mobile number** (provider or area code)

{$MOBILENUMBER} **Mobile number** (phone number)

{$P1} - {$Pn} **Freely defined fields** (please refer to the pool preview for the corresponding internal names)

{$POOLID} **ID of the pool**

{$RAW_...} **Internal value of a field (e.g. 0-4 for SALUTATION) **

5.4.2 Replacement variables in the eMailing

{$SUBJECT} **Subject line** of the eMailing

{$ONLINEMAIL_URL} **URL for eMailing** itself, displayed in the browser

{$NEWSLETTER_EMAIL_URL} **URL for eMailing** itself, displayed in the browser

{$NEWSLETTER_WEB_URL} **URL for web version** of the eMailing

{$NEWSLETTER_PDF_URL} **URL for PDF version** of the eMailing

{$NEWSLETTER_TEXT_URL} **URL for TEXT version** of the eMailing

{$NEWSLETTER_FEED_URL} **URL for Atom-Feed** of the eMailing (General outline; this placeholder will always link to the most widespread and most suitable format for feeds by the system)

{$NEWSLETTER_ATOM_URL} **URL for Atom-Feed** of the eMailing

{$NEWSLETTER_RSS_URL} **URL for RSS-Feed** of the eMailing

{$NEWSLETTER_VOICE_URL} **URL for reading aloud** of the eMailing Using the HTML comments <!–ISI_LISTEN_START–> and <!–ISI_LISTEN_STOP–> You can define which section you want or do not want to have read aloud.

{$NEWSLETTER_EN_VOICE_URL} **Reading out loud in English**

{$NEWSLETTER_IT_VOICE_URL} **Reading aloud in Italian**

{$NEWSLETTER_FR VOICE_URL} **Reading aloud in French**

{$NEWSLETTER_PODCAST_URL} **URL for Podcast** of the eMailing

{$NEWSLETTER_EN_PODCAST_URL} **Podcast in English**

{$NEWSLETTER_IT_PODCAST_URL} **Podcast in Italian**

{$NEWSLETTER_FR_PODCAST_URL} **Podcast in French**

{$DELETE_URL} **URL for cancellation of the newsletter**

{$REVOKE_URL} **URL for cancellation of the newsletter**

{$GRANT_URL} **URL for confirmation of the newsletter** (double opt-in)

{$REVOKE_TRACKING_URL} **URL zum Widerrufen der Trackingeinwilligung**

{$GRANT_TRACKING_URL} **URL zum Einwilligen in das Tracking**

{$NEWSLETTER_ID} **ID** of the eMailing unencrypted

{$NEWSLETTER_SID} **ID** of the eMailing encrypted

{$FORM...} **Web form link** - Link to pre-filled web form; `{$FORM:#}` with # as ID of the form

{$DATE} **Date** (Format: `Day.Month.Year`); date when the newsletter was last changed

{$DAY}, {$MONTH}, {$YEAR} **Date components**

{$CURRENT_DATE} **Current date** (Format: `Day.Month.Year`)

{$CURRENT_DAY}, {$CURRENT_MONTH}, {$CURRENT_YEAR} **Current date components**

{$DESCRIPTION} **Object name** of the eMailing in the system

5.4.3 Replacement variables in the article

`{$READ_PDF_URL}` **Link to article as PDF** (e.g. "more..." in an article)

`{$READ_LP_URL}` **Link to article as landing page** (e.g. "more..." in an article)

`{$READ_LPLINK_OR_LP_URL}` **Landing page link (if available), otherwise link to article landing page** (e.g. "more..." in an article)

`{$READ_TXT_URL}` **Link to online version TEXT** (e.g. "more..." in an article)

`{$READ_DE_VOICE_URL}` **Link for reading the article aloud** (only long text if available, otherwise short text)

`{$READ_EN_VOICE_URL}` **Reading out loud in English**

`{$READ_IT_VOICE_URL}` **Reading aloud in Italian**

`{$READ_FR_VOICE_URL}` **Reading aloud in French**

`{$READ_ALL_VOICE_URL}` **Link for reading the article aloud** (short text and long text)

`{$READ_EN_ALL_VOICE_URL}` **Reading out loud in English**

`{$READ_IT_ALL_VOICE_URL}` **Reading aloud in Italian**

`{$READ_FR_ALL_VOICE_URL}` **Reading aloud in French**

`{$SOCIAL_XING_URL}` **Article sharing in Xing**

`{$SOCIAL_TWITTER_URL}` **Article sharing in Twitter**

`{$SOCIAL_FACEBOOK_URL}` **Article sharing in Facebook**

`{$SOCIAL_MYSPACE_URL}` **Article sharing in MySpace**

`{$SOCIAL_GOOGLEBM_URL}` **Article sharing in Google Bookmarks**

`{$SOCIAL_LINKEDIN_URL}` **Article sharing in Linked In**

`{$SOCIAL_GOOGLEPLUS_URL}` **Article sharing in Google Plus**

`{$TEMPLATE_TYPE}` **Art/Type of template**; 1=email, 2=PDF, 3=Text, 4=Online, 5=Landing page, 12=RSS

`{$ARTICLE_ID}` **ID of the article**

`{$NAME}` **Name of the article**

`{$HEADLINE}` **Heading of the article**

`{$SHORTTEXT}` **Short text of the article**

`{$TEXT}` **Long text of the article**

`{$LINKCOUNT}` **Number of links to the article**

`{$LINKLIST}` **List of the links**

`{$IMG_URL}` **Article image**: Dynamic generation of images in predefined image sizes. For example, 200 pixels width with compression. Only if an image is available.

5.4.4 Replacement variables especially for LeadPage and websites

:{$RENDER_FORM:ID}: **Bindet ein Formular in ein LeadPage- oder Website-Objekt ein.** In diesem Fall muss das Action-Attribut in der Individualvorlage des Formulars händisch angepasst werden. :{$RENDER_FORM_UNPERSONALIZED:ID}: **Bindet ein Formular in ein LeadPage- oder Website-Objekt ein, ohne das Formular mit einem u-Parameter zu versehen, dadurch sind auch Neueintragungen möglich, obwohl bereits ein Profil erkannt wurde.**

5.4.5 Template and content management items

{$SLOT_0} - {$SLOT_9} **Article Slots** - Article of a slot; Slot for representing articles in the eMailing Slots can be used across all media.

{$INPUTFIELD_0} - {$INPUTFIELD_19} **Placeholder for single-line text input**

{$INPUTFIELD_0_VALUE} - {$INPUTFIELD_19_VALUE} **Placeholder for single-line text entry (will always return the value)**

{$TEXTAREA_0} - {$TEXTAREA_19} **Placeholder for multi-line text input**

{$TEXTAREA_0_VALUE} - {$TEXTAREA_19_VALUE} **Placeholder for multi-line text input (always returns the value)**

{$HTMLAREA_0} - {$HTMLAREA_19} **Placeholder for multi-line HTML editor**

{$HTMLAREA_0_VALUE} - {$HTMLAREA_19_VALUE} **Placeholder for multi-line HTML editor (always returns the value)**

5.4.6 Other replacement variables

{$UID} **Encrypted user ID** e.g. `http://...,u={$UID}`

{$ABSOLUTE_PATH} **Absolute URL of the system**

5.4.7 Form Variables

These variables find their use in the creation of forms with templates of their own:

{$ERROR_LIST} All intra-form validation errors involving text from the configuration; **Please note that this list does not display any other errors. More concerning the display of other errors under ** *Programming in templates*

{$...} A profile's every field along with the place holder from the pool is replaced by the appropriate input item. E.G. `{$NAME}` or `{$P1}`. Fields with several input items such as checkboxes are alternatively displayed by adding the option's ID. E.G. `{$P1_0}`, `{$P1_12345}`, `{$P1_67890}`. You extract the IDs from the preview of the pool.

{$LABEL_...} Like the fields - except that a corresponding `<label>` tag is created with the title of the input field

{$CAPTCHA} Creates a Captcha image (if activated)

{$VALUE_...} Returns the value of a field without creating a label or input.

{$CSRF_TOKEN_NAME} Returns the name of the CSRF token that is valid once.

{$CSRF_TOKEN} Outputs the value of the CSRF token that is valid once.

5.5 TAL Quick Start Guide

TAL is a very simple template programming language entered directly in HTML. The most important fields of application for TAL are the displaying of variable values, the displaying of HTML blocks based on conditions as well as moving through lists.

5.5.1 Integration of TAL in HTML

TAL is displayed directly on the HTML tags in the corresponding places with the help of own attributes on the affected tags and the name space *tal*. The entered instructions always affect the current tag and possibly all contents of the tag.

5.5.2 Displaying values

You can represent values from the system in two ways. Firstly, you can replace the tag with the content

```
<span tal:replace="THIS/getName">Name...</span>
```

The result in the source is then:

```
Ein Name
```

`tal:replace` ensures that the tag and all its children are replaced by the result of `THIS/getName`. This makes sense in all places where no tags are desired in the result anymore or where dummy contents were created nested within the tag. During the replacement, a text is normally always set as well so that there is a clean reproduction of the contents in a preview with the TAL instructions - without replacements of the system; e.g. in HTML editors - and you have a good assessment of the space required.

If you do not want to replace a tag because this has CSS classes, for example, then secondly, you can also replace just the content of the tag.

```
<span tal:content="THIS/getName" style="background-color: grey">Name...</span>
```

The result in the source is then

```
<span style="background-color: grey">Ein Name</span>
```

Here, everything that is set in the enclosing tag is therefore preserved - but not its original content.

5.5.3 Display blocks if a condition is fulfilled

With the `tal:condition` instruction you can display a tag and all of its children. The call takes place, for example, as follows

```
<div tal:condition="THIS/getCurrentUser/isInTargetGroup/selfRef/12345">
  Hallo, liebe Ski-Fahrer!
</div>
```

The result if the condition applies

```
<div>
  Hallo, liebe Ski-Fahrer!
</div>
```

If the condition does not apply, then none of the block (i.e. the `div`) is displayed.

If you want to reverse the condition or display an alternative, you then place the word `not` before the condition.

```
<div tal:condition="THIS/getCurrentUser/isInTargetGroup/selfRef/12345">
  Hallo, liebe Ski-Fahrer!
</div>

<div tal:condition="not:THIS/getCurrentUser/isInTargetGroup/selfRef/12345">
  Hallo, liebe Wintersportler!
</div>
```

5.5.4 Iterating via lists

To iterate via a list of objects, you use the command `tal:repeat`. In the course of this, the content of the tag is added for each element from the source of the object (mostly a list) and you can access the object currently passing though within this tag.

```
<ul>
  <li tal:repeat="link THIS/getLinks"><span tal:content="link/getText">Link</span></li>
</ul>
```

In this example, (`THIS/getLinks`) is therefore iterated via the list of the links. This is assigned within the loop to the temporary variable `link` for each element of the list, which can then be used again for all other operations within the tag. The result might then look like this

```
<ul>
  <li>Link 1</li>
  <li>Link 2</li>
  <li>Link 3</li>
</ul>
```

5.5.5 Modification of attributes

The attributes of a tag, similar to the content of a tag, can also be changed by TAL Here, you use the `tal:attributes` construct and always assign - separated by semicolons - attribute name and the value as well. The TAL attributes always have priority over those already set in the tag. This means you can preallocate specific values for representing the template - these are then overwritten by the value from the TAL command during runtime, however.

```
<img src="#" alt="Dummy-Text" tal:attributes="src THIS/getImageUrl; alt THIS/getHeadline" />
```

5.5.6 Output of a slot

The following command is recommended for reading out the content of a slot via Tal

```
<span tal:content="structure THIS/getFullNewsletter/getNewsletter/getSlot2">...</span>
```

The keyword structure ensures that any HTML tags that might possibly occur are handled correctly.

5.6 TAL-API

Tip: This chapter addresses itself to designers who are already familiar with how TAL works and require jump-off points in the system‘s API for the creation of eMailing and form templates. Many practical examples for the API functions provided can be found in the chapter entitled *Programming in templates*.

5.6.1 Newsletter

The newsletter is accessed with the variable `THIS`.

Access to a newsletter‘s slots

getSlots Dictionary of all the slots with the ID of the slot as key

hasSlots Bool, if slots are generally available

countSlots Number of slots

hasSlot0-hasSlot9 Bool, if the appropriately numbered slot is used

getSlot0-getSlot9 Correspondingly numbered slot

getFilledSlots Returns only the slots that also contain articles

getNewsletterDate Date object (s. below) from the last save

hasSelectedArticle True, if only one selected article is displayed (landing page)

getSelectedArticle Returns the article reference displayed on the landing page

hasArticleInAnySlot True, if there is at least one visible article in any slot

getContainer/<id> Returns the container with the specified ID

getSmartlink/<id> Returns the Smartlink with the specified ID

getBrowser

19. Browser

Slot

Methods of a slot (s. `getSlots` et al):

getArticles This slot’s every article reference

hasArticles Number of articles in the slot

countArticles Number of articles in the slot

hasOneArticle True, if the slot contains just one article

hasMoreThanOneArticle True, if the slot contains more than one article

getArticlesInTwoColumns the articles are returned as a two-dimensional array for display in two columns

getArticlesInColumns the articles are returned for display in multiple columns The number of desired columns is transferred as a parameter. `getArticlesInColumns/2` corresponds to the result of `getArticlesInTwoColumns`

renderWithTemplate/<id> Display of the complete slot with the article template `<id>`; if no ID is specified, the entire slot is then displayed with the standard template

Common newsletter and article procedures

Newsletter and article split the following procedures:

Access to the profile.

getCurrentUser (see Profile)

Mode that is currently displayed. This information can be useful when it comes to using an article template, for example, both for the website and for the landing page - albeit with different functions.

Important: These mode queries must be made to the newsletter reference. Example: *THIS/getNewsletter/isLandingPage*.

---

isLandingPage for checking whether you have a display for the landing page or the normal display (e.g. in PDF)

isEmailMode HTML email

isPdfMode PDF

isTextMode Text Email

isLPMode HTML landing page

isWebMode HTML-Web / Mobile

isAppendMode Visual editor

isAnyHTMLMode any HTML based mode

isSocialMode The object is called from a social network

isFacebookMode The object is called from the Facebook social network

isGooglePlusMode The object is called from the Google Plus social network

Values of the input fields

getInputFields List of 0-19 with the values of the fields `INPUTFIELD_0`--*'19'* (empty string, if field does not exist)

getTextAreas List of 0-19 with the values of the fields `TEXTAREA_0`--*'19'* (empty string, if field does not exist)

getHTMLAreas List of 0-19 with the values of the fields `HTMLAREA_0`--*'19'* (empty string, if field does not exist)

Access to the fields of the container type selected in the configuration.

getContentContainer Returns the container associated with the mailing.

Access to the date:

getCurrentDate Date object (s. below) with the current date

Subject:

getSubjectForCurrentUser Returns regular version of the subject taking the individualization into account

General:

getMandatorId Returns the client ID of the object

Common article reference and article procedures

Position an article inside the slot:

renderWithTemplate/<id> Display of the article with the article template `<id>`; if no ID is specified, the standard template of the slot is then used

getSortPos The article's index in the current slot (begins with 0)

getSortPosPlusOne `getSortPos` + 1 to be used for numbering e.g. in the table of contents or in headings

isEven Boolean, if the article is in an even position

isOdd Boolean, if the article is in an odd position

isFirst Boolean, if the First article is in a slot

isLast Boolean, if the Last article is in a slot

getPosCSSClass Returns `first`, ``last` or an empty string if the article is at the corresponding position.

getOddEvenCSSClass Returns `odd` or `even` depending on the position of the article in the slot.

getSlotId Returns the ID of the slot containing the article

Article Reference

Auxiliary object between article and slot.

getArticle actual article object

getUniqId Unique article ID within the object (for repeated use of an article in a newsletter)

Article

The article itself is accessed within an article template with the variable `ARTICLE` or `THIS`. It makes the following procedures available:

getId Article ID

getUniqId Unique ID of the article

getAnchor A unique anchor of the article can be used as a target/anchor for tables of contents

getAnchorHRef Link, which links to the corresponding Anchor (see getAnchor).

getName Name of the article

getAuthor Article author

hasHeadline Bool, if the article has stored a headline

getHeadline Article Headline

hasImage Bool, if the article has stored an image

getImageUrl Image URL

hasShortText Like `hasLongText` for the short text

getShortText Short text of the article

hasLongText Bool, if the article has a long text

getLongText, getText Long text of the article

hasLandingPageLink A landing page link exists

getReadLpLinkUrl A landing page link exists

hasLandingPage There exists either a landing page link or long text

getReadLpLinkOrLpUrl Landing page link or link to landing page

hasLinks Bool, if the article has links

countLinks Number of links

getLinks Returns a list of every link (see Links)

getNewsletter Newsletter object within which the articles is displayed.

getFullNewsletter Reference to the complete newsletter, e.g. to the landing page is required, since `getNewsletter` only returns the newsletter with the one current article here.

getReadLpUrl URL to the landing page

getReadPdfUrl URL to the PDF

getArticleDate Date object (s. below) with the storage date of the article

Shared methods of articles and containers

Tal kann innerhalb von Containern nur in Feldern des Typs `Code Macro` ausgeführt werden. `:SELF`: Führt, im Gegensatz zu `THIS`,in `Code Macro` Feldern auf den Container selber anstatt auf das Objekt in dem der Inhalt ausgegeben wird.

hasAttribute/<name> Bool, whether the article has an attribute with the name

getAttributeByName/<name> Returns the attribute object of the article with the specified name

getAttributesByGroup/<name> Returns a list of all attribute objects of an article from the category with the specified name

getAttributesByType/<type> Returns a list of all attribute objects of an article from the specified type (TYPE_IMAGE and TYPE_LINKLIST are currently supported)

getMandatorId Returns the client ID of the object

hasOptions Bool, whether the attribute has options

hasAnyOptionSet Bool, whether at least one option was set

getOptionByName Returns an option object with the specified name

Attribute object

If you access the attribute object directly, the system will then output the value of the field. This value is customized and possibly formatted to the current display format (HTML-Mail, Text-Mail, etc).

getValue

19. Direct access

getValueRaw Returns the value without formatting. The returned value can be a string (standard), list (link list) as well as an object (color, date).

hasValue Checks whether the field in the article was filled out

getName Outputs the name of the attribute

Field Type Color

This field returns a color object. In the case of direct access, the hexadecimal representation of the color value is returned.

toHex Returns the color value in hexadecimal representation

toRGB Returns the color value in RGB notation

lighten Lightens the color value

darken Darkens the color value

isDark Bool, whether the color is dark

isBright Bool, whether the color is light

A color object can also be used nested, e.g. color/darken/toRGB

Field type Value list

This field type returns a value list object. The default call via getValue returns a standard HTML representation of the list.

getKey Description of the value

getValue Contents of the value

Field type link list

This field type returns a link list object. During a default call via getValue, a standard HTML representation of the list is output.

getText Link Description

getUrl URL of the link

When calling via getValueRaw, you obtain an array of all the links. During iteration, the following accessories are available:

getText Returns the description of the link

getLink Returns the link URL (is **not** tracked in transmissions)

getTrackLink Returns the link URL (is tracked in transmissions)

Field type Image

This field type returns an image object.

getId Returns the ID of the image

getDescription Returns the name of the image

getUrl Returns the URL of the image

Field Type Image List

This field type returns an image list object. In the case of a default call up via getValue, a standard HTML representation of the list is displayed.

asHtml Returns the HTML representation of the image list

asText Returns the text representation of the image list

When calling via getValueRaw, you get an array of all the images. The following accessories are available when iterating:

getId Returns the ID of the image

getDescription Returns the name of the image

getUrl Returns the URL of the image

Field type Container

In this field type, the set container is returned whose attributes can then be accessed again via TAL.

Field type Code-Macro

This field type returns the rendered code macro from the container. The macro is rendered in the context of the call

Field type Container List

When calling `getValue` or `getValueRaw`, a list of the stored containers is returned via which you can then iterate. The individual containers can then be accessed via the usual article/container interface.

Field type Text (single-line input and multi-line inputs)

toUpper Converts all characters of the string to upper-case letters

toLower Converts all characters of the string to lower-case letters

length Returns the length of the string (ignores HTML)

countWords Returns the number of words in the string (ignores HTML)

stripHtml Removes HTML tags from the string

cut/xx Truncates the string from character xx (HTML tags are removed automatically at the same time)

md5 Returns the md5 hash of the string

All functions, with the exception of the count function and md5, can be nested in one another (e.g. `getValue/toUpper/cut/15`)

If the text is a table in CSV format, this can be divided up into a two-dimensional array by means of the following functions. When doing so, the lines are each separated by a line break and the columns are then separated either by a comma or tab.

separateByComma Separates the columns by means of comma characters

separateByTab Separates the columns by means of tab characters

Options Object

hasValue Bool, whether the option is set in the current object (container or article)

getValue Returns the value (the label) of the option selected

Smartlink object

The Smartlink object provides access to the links stored in a Smartlink

getLinklist Returns a link wrapper that returns an HTML formatted list of the links as default

Link wrapper

hasLinks Bool, whether links exist

getCount Returns the number of links

getLinks Returns a list of all links to iterate about (s. links)

asHtml Returns the links in an HTML formatted list

asText Returns the links as a single list

Links

getText Link Description

getUrl URL of the link

When calling via getValueRaw the following accessors are available:

getDescription Returns the description of the link

getLink Returns the link URL (is **not** tracked in transmissions)

getTrackLink Returns the link URL (is tracked in transmissions)

Date object

Used for working with dates and allows access to various types of information of the newsletter and article (s. above). If no other method is called, the result `getDate` is displayed.

getEpoch Number of seconds since the UNIX Epoch

isFirstQuarter, isSecondQuarter, isThirdQuarter, isFourthQuarter Bool, in which quarter the date is

isNorthernSpring, isNorthernSummer, isNorthernAutumn, isNorthernWinter, isSouthernSpring, is Bool, the time of year in the northern or southern hemisphere

getNorthernSeasonCSSClass, getSouthernSeasonCSSClass Time of year as text for using as CSS class. Returns *winter*, *spring*, *summer*, *autumn*

getDate Date as text

getYear, getYearTwoDigits Four-digit, two-digit year

getMonth Month

getDay Day

getMonthName Name of the month in local language This language is either taken from the object or can be set as a parameter (ISO code: *de*, *en*, ...)

getHour24Format, getHour12Format Hours

getMinutes Minutes

To output date fields formatted from the pool, a date object must first be generated from the values: e.g.: tal:define="date_object toDateObject:THIS/getCurrentUser/getValuesRaw/BIRTHDAY".

5.6.2 Website, SVG

getId Returns the ID of the object

getName Returns the name of the object

getCategoryId Returns the ID of the folder where the object is located

getContainer/<id> Returns the container with the specified ID

getContentContainer Liefert den Container aus der Konfiguration des dynamischen Bildes

hasContentContainer Prüft, ob ein ContentContainer in der Konfiguration des dynamischen Bildes hinterlegt wurde

getSmartlink/<id> Returns the Smartlink with the specified ID

getCategoryContent

19. Access to objects in the system

getCurrentUser

19. Profile

getBrowser

19. Browser

5.6.3 Form

The renderer of the form is accessed via the attribute `THIS`.

getOrderedAttributes Sorted, two-dimensional list of all form attributes

isSubmit Bool, if a display following a Submit is involved

getCurrentUser (see Profile)

getContainer/<id> Returns the container with the specified ID

getBrowser

19. Browser

Display and labels

getTitle Title of form.

getSubmitLabel Title of the Submit button

getShortText Introductory text of the form (must be displayed with `structure`)

hasShortText Bool, if an introductory text is present

hasResourceImageUrl Bool, if an image is stored

getResourceImageUrl URL to a stored image

getBaseCSS CSS generated from the form configuration data

getAdditionalCSS Open CSS input from the form configuration

isShowValidationErrorsAsList Bool, if errors are to be displayed as a list, or for the attribute

getValidationErrorPrefixText Introductory text for the validation errors list

getFormLanguage Language of the form/language of the entry following registration

getLanguage Language of the object (e.g., for internationalization solely for the display)

renderCaptcha Renders a Captcha image (if activated)

Error handling

renderErrorListHTML HTML list with all error messages; must be output with `structure`

hasErrors Bool, if there was any error in the form (see the following more detailed checks)

Content-related errors:

hasValidationErrors Bool, if there was a validation error; this value can be used in order to read out blocks with global error messages

getValidationErrorAttributes List of all form attributes for which there was a validation error

hasCaptchaError Bool, whether there was a Captcha validation error

Mandatory Errors:

hasMandatoryErrors Bool, if there are missing data in mandatory fields

getMandatoryErrorAttributes List of all form attributes in which no entry was made and are marked as *mandatory*

Uniqueness errors:

hasIdentityErrors Bool, if there was an error during the user validation

isShowIdentityErrorText Bool. if a message is saved in the form

getIdentityErrorText Text message from the admin interface

Duplicate errors:

hasDuplicationErrors Bool, if when registering it caused a conflict with an existing entry

isShowDuplicationErrorText Bool. if a message is saved in the form

getDuplicationErrorText Text message from the admin interface

Form attribute

Form attribute procedures

getLabel Attribute label

renderLabelHTML Label of the attribute as HTML tag (`label`); must be output with `structure`

renderWidgetHTML Input element of the attribute as HTML tag; must be output with `structure`

hasError Bool, if there was any error involving this field (see the following more detailed field checks)

hasValidationError Bool, if there is a validation error

isMandatory Bool, if the field is a mandatory field

hasMandatoryError Bool, if no entry is made for a mandatory field

getValidationErrorText Text in the case of an error

getSuccessfulSubscriptions Int, number of profiles already entered successfully.

getSubscriptionLimitCount Int, maximum total number of registrations before the registration limit takes effect.

isSubscriptionLimitSet Bool, whether the entry limitation was activated (checkbox).

getSubscriptionLimitLeft Int, number of remaining registrations before the registration limit takes effective.

getSubscriptionLimitRedirectUrl String, URL to be redirected to when the entry limitation takes effect.

isSubscriptionLimitRedirectActive Bool, whether the entry limitation was activated and the limit takes effect.

getSubscriptionLimitStartTime Int, timestamp from when a form is accessible if this limit was activated.

getTimeIntervalSinceSubscriptionLimitStarts Formatted time interval in "DD days HH hours MM minutes", time from when the form is accessible.

getTimeIntervalUntilSubscriptionLimitStarts Formatted time interval in "DD days HH hours MM minutes", time until when the form is accessible.

isSubscriptionLimitStartTimeSet Bool, whether the time limitation for the start was activated (checkbox).

getSubscriptionLimitStartTimeRedirectUrl String, URL to be redirected to when the time limitation for the start takes effect.

isSubscriptionLimitStartTimeRedirectActive Bool, whether the time limitation for the start was activated and the limit takes effect.

getSubscriptionLimitEndTime Int, timestamp until when a form is accessible if this limit was activated.

getTimeIntervalSinceSubscriptionLimitEnds Formatted time interval in "DD days HH hours MM minutes", time from when the form is no longer accessible.

getTimeIntervalUntilSubscriptionLimitEnds Formatted time interval in "DD days HH hours MM minutes", time until when the form is no longer accessible.

isSubscriptionLimitEndTimeSet Bool, whether the time limitation for the end was activated (checkbox).

getSubscriptionLimitEndTimeRedirectUrl Int, maximum number of registrations before the registration limit takes effect.

isSubscriptionLimitEndTimeRedirectActive Bool, whether the time limitation for the end was activated and the limit takes effect.

hasAutomatedEntryProtectionError Bool whether the protection against automatic entries has prevented an entry.

isAutomatedEntryProtectionEnabled Bool whether the protection against automatic entries is activated.

isMultipleSubmitProtectionEnabled Bool, whether the multiple entry protection is enabled.

getCsrfToken CSRF token that is valid once only.

getCsrfTokenName Name of the CSRF token that is valid once.

If protection against automatic entries is used with an individual template, the CSRF token field must be added manually: `<input type="hidden" tal:attributes="value THIS/getCsrfToken; name THIS/getCsrfTokenName;" name="name" value="value"/>`

5.6.4 Profile

The following functions return objects for a simple comparison using relational operators (see below). Access to the fields occurs immediately using the name of the field from the pool (compare Pool Preview). You can then read the corresponding values with the `value` function.

getValues List of values for customization

getValuesRaw List of values for customization; this list contains the internal presentation of data (ID values, etc.)

isDummy If it involves *Max Sampleman*; e.g. in previews

isTrackable If the profile did not object to tracking

isInTargetGroup Returns true if the profile is in the specified target group: `THIS/getCurrentUser/isInTargetGroup/selfRef/1234`

In order to output date fields from the pool formatted, a date object must first be generated from the values: e.g.: tal:define="date_object toDateObject:THIS/getCurrentUser/getValuesRaw/BIRTHDAY"

5.6.5 Comparison operators

Virtually every object in the system can perform simple checks on your attributes (newsletter, article, link, form, form attribute, ...). The syntax of the comparisons is always:

```
<object>/<operator>/<method>/<value>
```

With:

object Object affected; mostly `THIS`

operator One of the operators for comparisons. see below

method An object's TAL procedure whose result is to be compared

value Value against these is to be checked; if an operator does not make a comparison, then a dummy text at a minimum must obtain here.

E.G. `THIS/equals/getLanguage/DE` checks whether the language *German* is set.

equals Value equal

contains includes the string anywhere (caseINsensitive)

isEmpty Blank string

getSuffix Returns the file extension - or a blank string

stripSuffix Returns the text without the file extension - or the full text

toLower String only with lower case letters

toUpper String only using capital letters

maxlen Truncates the string from the position specified

Additional formatting prefixes:

string Handles the following characters as a string.

quotenltobr Replaces line breaks in multiline inputs by HTML line breaks (
)

httpToHttps Replaces http:// with https:// in each string.

In addition, the following modifiers are available for string operations.

equals Value equal

greaterthan Greater value

lessthan Lesser value

E.G. <div tal:omit-tag="" tal:define="num string:4"><div tal:condition="greaterthan:string:$
The div element is displayed if 4 is greater than 3. At the same time, TAL variables that have already been defined can also be used, of course.

containsString Second text is a subset of the first text.

e.g. <div tal:define="string1 string:foo; string2 string:oo"><div tal:condition="containsString:string:${string1}/${string2}">AGGI</div></div>

Client check

Methods for determining whether the TAL is in a specific client.

THIS/getApi/getOwnershipAccessor/compareMandatorId/ID_GEGEN_DIE_GEPRUEFT_WERDEN_SOLL

The condition is true if the TAL was executed in the specified client. This method is limited to eMailing templates and eMailings.

Access to objects in the system

A folder‘s contents can be read out as a list. The <id> used is the ID of the folder in the system (compare URL in the Admin interface).

THIS/getCategoryContent/byDescription/<id> Lists sorted according to name

THIS/getCategoryContent/byLastModifiedDesc/<id> List in descending order of the time of last change

The result supports the following procedures:

getDescription Name of the object

getComment An object‘s comments

getURL URL to the object

renderResource Display objects (not every object supports this call)

Display of objects

The following objects support the display with the method renderResource or renderResource/<format>:

Artikel The ID of an article template is expected as <format>.

Browser

The current browser of the user can be queried via this API.

getName Returns the name of the browser (e.g. Firefox, IE)

getMajorVersion Returns the version number of the browser before the comma

getMinorVersion Returns the version number of the browser after the comma

isFirefox Bool, whether the browser is Firefox

isIE Bool, whether the browser is Internet Explorer

isSafari Bool, whether the browser is Safari

isChrome Bool, whether the browser is Chrome

isMSOffice Bool, whether the browser is Microsoft Office

isIphone Bool, whether the browser is running on iPhone

isIpad Bool, whether the browser is running on iPad

isAndroid Bool, whether the browser is running on Android telephone/tablet

isMobileDevice Bool, whether the browser is running on any telephone/tablet

getOS Returns the operating system (e.g. MACOSX, WinXP)

isMac Bool, whether the operating system used is MacOS X

isWindows Bool, whether the operating system used is Windows

isLinux Bool, whether the operating system used is Linux

Request

You can access parameters in the URL with TAL.

REQUEST Reads out a parameter from the URL.

Z.B.: `tal:condition="exists:REQUEST/online"` prüft ob der Parameter: "online" in der URL vorkommt. `tal:define="req REQUEST/online|string:fallback"` speichert den Inhalt des Parameters: "online" in der Tal-Variable "REQUEST". Ist der Parameter nicht in der URL enthalten wird der String: "fallback" gespeichert.

A common application scenario is to hide the online view link if you are already in the browser. The above condition can be used for this and only the parameter "online" must be added to the replacement variable for the online version: `{$ONLINEMAIL_URL},online`.

Access to the DOM of an integrated XML document

A simple TAL integration of a document's DOM tree is available for the integration of contents off a transferred XML document. Access to the nodes in the DOM is gained via the name of the node - lower levels are accessed via nested paths. The end of a TAL path is always interpreted as a string.

In order to iterate in a certain path via a list of nodes, an `[]` is attached to the last node. The system returns a list of every node along with the name.

Attributes are accessed from the node by the prefix of an `@` sign. The attribute with the name following the `@` of the current node in the path is then meant.

Tracking of links can be achieved by attaching the '/tracklink' postfix to the respective node path when the external XML is accessed. This makes it possible to track links in nodes with CDATA contents as well as links in nodes with substructures. In addition, the content of a text node can also be specifically registered and tracked in this way.

Example XML document

```xml
<?xml version="1.0" encoding="utf-8"?>
<root>
  <some_text_node>Irgendein Text</some_text_node>

  <some_node_with_childs>
    <next_level>
```

```
      <next_level>Irgendein Text - aber tiefer im Baum</next_level>
    </next_level>
  </some_node_with_childs>

  <more_than_one_node>1</more_than_one_node>
  <more_than_one_node>2</more_than_one_node>
  <more_than_one_node>3</more_than_one_node>

  <more_than_one_node_with_childs><child>1</child></more_than_one_node_with_childs>
  <more_than_one_node_with_childs><child>2</child></more_than_one_node_with_childs>
  <more_than_one_node_with_childs><child>3</child></more_than_one_node_with_childs>

  <some_node_with_attribute attribute="Irgendein Attribut" />

  <some_node_with_cdata><![CDATA[<h1>Hier kann auch HTML stehen</h1>]]></some_node_with_cdata>

  <node_with_links>
     <a href="http://www.example.com">Example.com</a>
     <a href="http://www.example.com/example">Example.com Example</a>
  </node_with_links>
</root>
```

Example TAL for accessing this document (starting point for the above document is `xml` in TAL)

```
<div id="showcase" tal:define="xml THIS/getExternalXml">
  <div tal:content="xml/some_text_node"></div>

  <div tal:content="xml/some_node_with_childs/next_level/next_level"></div>

  <div tal:content="xml/some_node_with_attribute/@attribute"></div>

  <div>
    <span tal:repeat="node xml/more_than_one_node[]" tal:content="node"></span>
  </div>

  <div>
    <span tal:repeat="node xml/more_than_one_node_with_childs[]" tal:content="node/child"></span>
  </div>

  <div tal:content="xml/some_node_with_cdata"></div>

  <div tal:content="structure xml/some_node_with_cdata"></div>

  <div tal:content="structure xml/some_with_links/tracklink"></div>
</div>
```

If the XML content cannot be retrieved for some reason or an error occurs during the retrieval, the attempt is repeated two more times (with 10 and 60 minutes delay in each case). If the transmission also fails the third time, this is saved and displayed as *Transmission error (XML)* in the statistic of the transmission.

5.7 Bookmarks for developers

Lots of things can be done much more easily when you have the right tools at hand. For this reason, we have compiled here the most important tools and documentation that our developers also use on a daily basis.

5.7.1 Documentation for internal components

- PHPTAL (deutsch)
- PHPTAL (english)
- XML-RPC
  - Bibliotheken in allen wichtigen Programmiersprachen
  - Generische Anwendungsbeispiele von XML-RPC

5.7.2 Validators

- CSS-Validator
- CSS Validator

5.7.3 JavaScript/AJAX/Web 2.0 libraries

- prototype.js multi-browser library functions for DOM access and -manipulation, AJAX, ...
  - Documentation
  - Developer Notes

5.7.4 Firefox Extensions

- Web Developer Extension - Swiss army knife extension
- Firebug - DOM-Inspector, JavaScript-Console and Script-Debugger rolled into one

5.7.5 General

- Lorem Ipsum Generator - also for Chinese/Japanese

5.8 System installation and configuration

5.8.1 Checklist for your own domain

1. register your own domain - e.g. yourCompany-newsletter.de or create a subdomain - e.g. newsletter.yourCompany.de
2. Ask the support for an IP address
3. Enter this IP address as A-entry in the (sub) domain

5.8.2 Checklist for white label

1. After concluding a contract, you are informed about your login domain. In case of change requests, please contact your sales contact person or the support team.
2. To customize the appearance in the system to a desired look and feel, change to the global level in the system. There, under the `Settings` folder is the configuration for the `appearance`. After creating a new image theme, you have to assign the corresponding image theme once to the system users under `Users`. You must also do the same when creating a new user.
3. To customize the login page, change to the global level in the system. In the `Appearance` folder under `Settings` you will find the area `Login page`. Select the domain for which you want to customize the login page, add a background image and / or your log and customize the font color and font type.

5.8.3 Checklist for the configuration of the DNS entries for bounce management on your own domain.

If you have your own email address for the transmission, you can send out emails of your company so that there is no longer any information indicating a transmission by an ASP service provider. Since the mail servers of the system are stored on the corresponding domain and real email addresses of the system are used as bounce and sender addresses, you reach an optimum level for a reliable delivery of your emails.

1. Your own domain must be registered or an existing domain can be referred to, but this must not already be in use for the mail transmission.
2. Contact the support and tell them which domain should be used and which addresses are to be set up for which customers/clients (localpart@...). When doing so, also specify which address the incoming emails are to be routed to adjusted for hard and soft bounces. You will then receive the settings from the support for MX and SPF records in your DNS server.
3. DKIM: For the integration of a DKIM configuration you should either provide us with your private key (not recommended) or contact the support to start the configuration. You will then receive a TXT entry as information which must be stored in the DNS This might look like this:

   ```
   ems._domainkey.example.com IN TXT "v=DKIM1; p=somekeyhere...; t=y"
   ```

   After entry in the DNS, DKIM for the email domain is activated in the system. The value `t=y` can then be removed at a later stage, since this indicates that the settings have been entered for testing. By removing this, the configuration is then focused.

5.9 New functions of the last updates

Here you will find a brief overview of all features and functions that have been added to new versions of the system.

5.9.1 v7.129 (27.11.2020)

Detail improvements:

- Freischaltung AddOns
- Neue Anredeform “Divers”
- General troubleshooting and performance optimization.

5.9.2 v7.128 (20.11.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.3 v7.127 (13.11.2020)

Detail improvements:

- Anpassungen in der GUI-Darstellung.
- General troubleshooting and performance optimization.

5.9.4 v7.126 (06.11.2020)

Detail improvements:

- Änderung der Attribut-Optionen Struktur im Formular-Api Result.
- General troubleshooting and performance optimization.

5.9.5 v7.125 (30.10.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.6 v7.124 (23.10.2020)

Detail improvements:

- Änderungsdatum bei Profile-Update in Campaing-Designer optional setzen.
- General troubleshooting and performance optimization.

5.9.7 v7.123 (16.10.2020)

Detail improvements:

- API-Toggle und CORS-Whitelisting für Formular API
- General troubleshooting and performance optimization.

5.9.8 v7.122 (09.10.2020)

Detail improvements:

- Mailing/Template-Id wird beim Anwenden von Vorlagen angezeigt.
- General troubleshooting and performance optimization.

5.9.9 v7.121 (02.10.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.10 v7.120 (25.09.2020)

Detail improvements:

- Anzeige des ungefähren Versandzeitpunkt des Gewinner-Mailings bei Splitkampagnen.
- General troubleshooting and performance optimization.

5.9.11 v7.119 (18.09.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.12 v7.118 (11.09.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.13 v7.117 (04.09.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.14 v7.116 (28.08.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.15 v7.115 (21.08.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.16 v7.114 (14.08.2020)

Detail improvements:

- Optimierung des Texts in der Standardmail bei Vergabe eines temporären Passworts
- General troubleshooting and performance optimization.

5.9.17 v7.113 (07.08.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.18 v7.112 (31.07.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.19 v7.111 (24.07.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.20 v7.110 (17.07.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.21 v7.109 (10.07.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.22 v7.108 (03.07.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.23 v7.107 (26.06.2020)

Detail improvements:

- Neues Zielgruppen-Statement zur Selektion von Formular-Absendern.
- General troubleshooting and performance optimization.

5.9.24 v7.106 (19.06.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.25 v7.105 (12.06.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.26 v7.104 (05.06.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.27 v7.103 (26.06.2020)

Detail improvements:

- Zielgruppen-Statement zur Selektion von Formular-Absendern.
- General troubleshooting and performance optimization.

5.9.28 v7.102 (19.06.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.29 v7.101 (12.06.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.30 v7.100 (05.06.2020)

Detail improvements:

- Campaigndesigner V0 deaktiviert.
- General troubleshooting and performance optimization.

5.9.31 v7.99 (29.05.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.32 v7.98 (22.05.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.33 v7.97 (15.05.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.34 v7.96 (08.05.2020)

Detail improvements:

- Optionale Objekt-Felder stehen via Tal zur Verfügung.
- General troubleshooting and performance optimization.

5.9.35 v7.95 (01.05.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.36 v7.94 (24.04.2020)

Detail improvements:

- LeadPage-Vorlagen erlauben die Nutzung weiterer Vorlagen als Master.
- Formular-Felder können in Pool-Alias-Szenario ausgeblendet werden.
- General troubleshooting and performance optimization.

5.9.37 v7.93 (17.04.2020)

Detail improvements:

- Einführung eines neuen GUI-Headers.
- Splitkampagnen-Statistiken erweitert.
- General troubleshooting and performance optimization.

5.9.38 v7.92 (10.04.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.39 v7.91 (03.04.2020)

Detail improvements:

- X-Unsubscribe-Header wurde entfernt, da er keine Relevanz mehr hat.
- Alternative Browser-Abmelde-Möglichkeit für List-Unsubscribe-Header.
- Für komplexe Newsletter Szenarien, kann der Abmelde-Link für Mail Clients umkonfiguriert werden.

- General troubleshooting and performance optimization.

5.9.40 v7.90 (27.03.2020)

Detail improvements:

- Die automatische Übersetzung von Artikeln übersetzt nun auch die Beschreibungen in einer Link-Liste. Die Werte in der Link-Liste sind davon ausgenommen.
- General troubleshooting and performance optimization.

5.9.41 v7.89 (20.03.2020)

Detail improvements:

- Campaign-Designer Beta
- General troubleshooting and performance optimization.

5.9.42 v7.88 (13.03.2020)

Detail improvements:

- Event- & Triggermailing Versendungen werden zeitlich korrekt zugeordnet.
- Abfragen von Ordner-Namen und Übergeordneten Ordnern via SOAP.
- General troubleshooting and performance optimization.

5.9.43 v7.87 (06.03.2020)

Detail improvements:

- Anlegen/Aktualisieren von Artikel-Vorlagen über SOAP.
- gCaptcha von 2 auf 3 aktualisiert.
- Möglichkeit in standard-Formularen den Viewport-Metatag, für bessere Darstellung auf mobilen Endgeräten, zu aktivieren.
- General troubleshooting and performance optimization.

5.9.44 v7.86 (28.02.2020)

Detail improvements:

- Milestones können im Report ausgewertet werden.
- General troubleshooting and performance optimization.

5.9.45 v7.85 (21.02.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.46 v7.84 (14.02.2020)

Detail improvements:

- Darstellungstests für Mobilgeräte sind nun scrollbar.
- General troubleshooting and performance optimization.

5.9.47 v7.83 (07.02.2020)

Detail improvements:

- Erweiterung der Deepl-Integration zur Übersetzung von Dateien der Typen .txt, .html, .docx oder .pptx
- General troubleshooting and performance optimization.

5.9.48 v7.82 (31.01.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.49 v7.81 (24.01.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.50 v7.80 (17.01.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.51 v7.79 (10.01.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.52 v7.78 (03.01.2020)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.53 v7.77 (27.12.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.54 v7.76 (20.12.2019)

Detail improvements:

- Neue Ersetzungsvariable für Tracking-Einwilligung.
- General troubleshooting and performance optimization.

5.9.55 v7.75 (13.12.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.56 v7.74 (06.12.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.57 v7.73 (29.11.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.58 v7.72 (22.11.2019)

Detail improvements:

- Erweiterung der Löschabhängigkeitsprüfungen.
- General troubleshooting and performance optimization.

5.9.59 v7.71 (15.11.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.60 v7.70 (08.11.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.61 v7.69 (01.11.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.62 v7.68 (25.10.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.63 v7.67 (18.10.2019)

Detail improvements:

- Neue Tab-Bezeichnungen der Versandvorschau-Tabs.
- General troubleshooting and performance optimization.

5.9.64 v7.66 (11.10.2019)

Detail improvements:

- Neue TAL-Variable 'SELF' um in Tal-Macro Attributen auf das Elternobjekt zu referenzieren.
- General troubleshooting and performance optimization.

5.9.65 v7.65 (04.10.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.66 v7.64 (27.09.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.67 v7.63 (20.09.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.68 v7.62 (13.09.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.69 v7.61 (06.09.2019)

Detail improvements:

- Systemsuche sucht je nach Berechtigung auch in globalen Daten.
- General troubleshooting and performance optimization.

5.9.70 v7.60 (30.08.2019)

Detail improvements:

- Der Artikel-Import (RSS-Feed) kann Links nun wahlweise in die Linkliste oder den Landingpage-Link importieren.
- General troubleshooting and performance optimization.

5.9.71 v7.59 (23.08.2019)

Detail improvements:

- Fehlende Übersetzungen für standard Pool-Felder wurden hinzugefügt.
- General troubleshooting and performance optimization.

5.9.72 v7.58 (16.08.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.73 v7.57 (09.08.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.74 v7.56 (02.08.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.75 v7.55 (26.07.2019)

Detail improvements:

- Erweiterte IP-Restriktionen in Sicherheitsrichtlinien erlauben Einschränkung von personenbezogenen Daten.
- General troubleshooting and performance optimization.

5.9.76 v7.54 (19.07.2019)

Detail improvements:

- Erweiterte Löschabhängigkeitsprüfung
- General troubleshooting and performance optimization.

5.9.77 v7.53 (12.07.2019)

Detail improvements:

- Own roles can now be assigned to other users from within the role.
- General troubleshooting and performance optimization.

5.9.78 v7.53 (05.07.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.79 v7.52 (28.06.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.80 v7.51 (21.06.2019)

Detail improvements:

- New, improved link tracking for trigger mails
- General troubleshooting and performance optimization.

5.9.81 v7.50 (14.06.2019)

Detail improvements:

- Roles show all users who use this role.
- General troubleshooting and performance optimization.

5.9.82 v7.49 (07.06.2019)

Detail improvements:

- With activated eMailings (Event) now the status of the ‚Weekend' setting is displayed.
- General troubleshooting and performance optimization.

5.9.83 v7.48 (31.05.2019)

Detail improvements:

- The SOAP method for querying bounces for profiles from global pools now also supplies the bounces that occurred in client eMailings.
- General troubleshooting and performance optimization.

5.9.84 v7.47 (24.05.2019)

Detail improvements:

- Data export of privacy data now has a language selection.
- General troubleshooting and performance optimization.

5.9.85 v7.46 (17.05.2019)

Detail improvements:

- File downloads automatically receive a file extension.
- General troubleshooting and performance optimization.

5.9.86 v7.45 (10.05.2019)

Detail improvements:

- New activities tab in the profile details.
- General troubleshooting and performance optimization.

5.9.87 v7.44 (03.05.2019)

Detail improvements:

- Conversion of all internal and integration links to https.
- General troubleshooting and performance optimization.

5.9.88 v7.43 (26.04.2019)

Detail improvements:

- Google+ has been removed.
- A tracking revocation at profile level has priority over anonymized data storage.
- General troubleshooting and performance optimization.

5.9.89 v7.42 (19.04.2019)

Detail improvements:

- Article translation now also supports Portuguese and Russian.
- Profile tracking data can be exported via the Privacy tab of the profile details.
- Mailing Rollout of global mailings can be activated via template.
- Optical adjustments to profile overview page.
- The shipping start column of the shipping list now indicates the time zone UTC.
- General troubleshooting and performance optimization.

5.9.90 v7.41 (12.04.2019)

Detail improvements:

- All list fields in articles and containers are now sortable.
- General troubleshooting and performance optimization.

5.9.91 v7.40 (05.04.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.92 v7.39 (29.03.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.93 v7.38 (22.03.2019)

Detail improvements:

- New BCC function for eMailings configurable in eMailing template.
- General troubleshooting and performance optimization.

5.9.94 v7.37 (15.03.2019)

Detail improvements:

- New profile detail page
- New SOAP method for deleting pool fields
- General troubleshooting and performance optimization.

5.9.95 v7.36 (08.03.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.96 v7.35 (01.03.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.97 v7.34 (22.02.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.98 v7.33 (15.02.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.99 v7.32 (08.02.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.100 v7.31 (01.02.2019)

Detail improvements:

- New SOAP method for reading all tracking data on profile level (profile::getTrackingHistory)
- General troubleshooting and performance optimization.

5.9.101 v7.30 (25.01.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.102 v7.29 (18.01.2019)

Detail improvements:

- General troubleshooting and performance optimization.

5.9.103 v7.28 (11.01.2019)

Detail improvements:

- The standard confirmation pages, such as "Registration/deregistration confirmation", "Tracking revocation" and "Delete tracking" have been revised, modernised and standardised.
- Trigger eMailings now display the same dispatch warnings and dispatch error messages in the Test Dispatch tab as eMailings in the Test Dispatch tab.
- General troubleshooting and performance optimization.

5.9.104 v7.27 (28.12.2018)

Detail improvements:

- A “Deactivate tracking” link has been added to the mailings in the sample environment. This link is only displayed if tracking has not yet been deactivated.
- General troubleshooting and performance optimization.

5.9.105 v7.26 (14.12.2018)

Detail improvements:

- Extended error messages in the mailing and template preview.
- Extended HTML check in shipping tab.
- Soap method for creating target groups using PoolAttributeOption now also supports Pool-Alias object
- General bug fixes

5.9.106 v7.25 (30.11.2018)

Detail improvements:

- Enhancement of the new profile detail page with form log data
- General bug fixes

5.9.107 v7.24 (16.11.2018)

Detail improvements:

- New Campaign Designer Campaign “Copy in Campaign”
- New Campaign-Designer condition “receive eMailing”
- Campaign Designer Starting point “Form” now allows multiple passes.
- Verwendung des List-Unsubscribe-Post Header (RCF 8058).
- General bug fixes

5.9.108 v7.23 (02.11.2018)

Detail improvements:

- eMailing templates can enable or disallow file attachments.
- When creating article types, all system fields can be automatically hidden.
- General bug fixes

5.9.109 v7.22 (19.10.2018)

Detail improvements:

- eMailing-history can be queried via the reporting API. This is possible for sending, eMailing (Trigger) and for eMailing (Event).
- The personalization of the LeadPage supports user identification via cookies.
- The Query Manager delivers standard system fields and their contents uniformly in English only.
- General bug fixes

5.9.110 v7.21 (05.10.2018)

Detail improvements:

- General bug fixes

5.9.111 v7.20 (21.09.2018)

Detail improvements:

- New Tal method to compare strings "containsString".

5.9.112 v7.19 (07.09.2018)

Detail improvements:

- General bug fixes

5.9.113 v7.18 (24.08.2018)

Detail improvements:

- New privacy tab in the profile history, which displays information about the entry date, tracking revocation, permission IPs, etc.
- The icons for locked objects, such as images, were partially placed unfavorably and were reworked.
- The shipping folder and the subfolder are now freely renameable.
- General bug fixes

5.9.114 v7.17 (10.08.2018)

Detail improvements:

- General bug fixes

5.9.115 v7.16 (27.07.2018)

Detail improvements:

- General bug fixes

5.9.116 v7.15 (12.07.2018)

Detail improvements:

- New scoring globe for displaying scoring behavior.
- Optimization of object and system icons.
- General bug fixes

5.9.117 v7.14 (28.06.2018)

Detail improvements:

- Free and automatic translation function for articles in up to seven different languages.
- New column in object lists: Language
- General bug fixes

5.9.118 v7.13 (14.06.2018)

Detail improvements:

- General bug fixes

5.9.119 v7.12 (31.05.2018)

Detail improvements:

- New Reporting API table for querying the profile change log.
- Display form integration by default with https.
- The Evalanche Performance Monitor (Beta) is linked on the login page.
- Litmus Userengagement (read time analysis) deactivated for data protection reasons.
- General bug fixes

5.9.120 v7.11 (17.05.2018)

Detail improvements:

- The Test Dispatch and Individualization tabs have been unified for all eMailing-related objects as well as LeadPages and templates. The selection of the individualization target group can now be made in the tab "Configuration" under "Individualization & test dispatch". Within the framework of the migration, the originally saved target group selection of the test mailing was adopted.
- SOAP: Dispatches can be inserted into a reporting via SOAP.
- General bug fixes

5.9.121 v7.10 (03.05.2018)

Detail improvements:

- In profile lists, all columns can now be faded in and out with one click.
- All mailings that can be sent can now be queried per client via Soap.
- General bug fixes

5.9.122 v7.09 (19.04.2018)

Detail improvements:

- General bug fixes

5.9.123 v7.08 (05.04.2018)

Detail improvements:

- General bug fixes

5.9.124 v7.07 (22.03.2018)

Detail improvements:

- Image upload limitation to 1MB

5.9.125 v7.06 (08.03.2018)

Detail improvements:

- Upgrade the FusionCharts
- General bug fixes

5.9.126 v7.05 (22.02.2018)

Detail improvements:

- General bug fixes

5.9.127 v7.04 (09.02.2018)

Detail improvements:

- New possibility to overwrite scoring values for all individualized objects.
- General bug fixes

5.9.128 v7.03 (25.01.2018)

Detail improvements:

- General bug fixes

5.9.129 v7.02 (11.01.2018)

Detail improvements:

- General bug fixes

5.9.130 v7.01 (28.12.2017)

Detail improvements:

- Global pools can now be used in the Campaign Designer.
- General bug fixes

5.9.131 v6.37 (30.11.2017)

Detail improvements:

- General bug fixes

5.9.132 v6.36 (16.11.2017)

Detail improvements:

- New inverted display option for checkboxes.
- General bug fixes

5.9.133 v6.35 (02.11.2017)

Detail improvements:

- New methods to create and change article- and container types.
- General bug fixes

5.9.134 v6.34 (19.10.2017)

Detail improvements:

- The Campaign object now has a list of all profiles that are active in the campaign.
- General bug fixes

5.9.135 v6.33 (05.10.2017)

Detail improvements:

- General bug fixes

5.9.136 v6.32 (22.09.2017)

Detail improvements:

- General bug fixes

5.9.137 v6.31 (08.09.2017)

Detail improvements:

- General bug fixes

5.9.138 v6.30 (24.08.2017)

Detail improvements:

- LeadPage templates can be deactivated for new creation
- Profile scoring now also allows negative numerical values
- Successful delivery of temporary passwords is acknowledged with a message.
- The TouchPoint statistics now show the names of the pages as well as their paths.
- Replace the Recaptcha library -> New ReCaptcha.
- Dispatch time settings in sent mailing.
- List of countries supplemented by "Caribbean Netherlands", "Curaçao", "Southern Sudan", "Sint Maarten (Dutch part)".
- New iPhone versions in Client Check.
- Remove subject line check because service has been discontinued.
- General bug fixes

5.9.139 v6.29 (10.08.2017)

Detail improvements:

- General bug fixes

5.9.140 v6.28 (27.07.2017)

Detail improvements:

- Pool-Alias can now be used in the Campaign-Designer
- The Reporting API now displays the scoring description in the Soring table.
- General bug fixes

5.9.141 v6.27 (13.07.2017)

Detail improvements:

- General bug fixes

5.9.142 v6.26 (29.06.2017)

Detail improvements:

- General bug fixes

5.9.143 v6.25 (15.06.2017)

Detail improvements:

- General bug fixes

5.9.144 v6.24 (01.06.2017)

Detail improvements:

- General bug fixes

5.9.145 v6.23 (19.05.2017)

Detail improvements:

- Trigger eMailings can now be selected in the report
- General bug fixes

5.9.146 v6.22 (04.05.2017)

Detail improvements:

- General bug fixes

5.9.147 v6.21 (20.04.2017)

SOAP Api

- Subfolders of folders can now be queried

Detail improvements:

- General bug fixes

5.9.148 v6.20 (06.04.2017)

SOAP Api

- Folders can now be created and deleted

Detail improvements:

- General bug fixes

5.9.149 v6.19 (23.03.2017)

Detail improvements:

- General bug fixes

5.9.150 v6.9 (09.03.2017)

Detail improvements:

- General bug fixes

5.9.151 v6.8 (23.02.2017)

Detail improvements:

- General bug fixes

5.9.152 v6.7 (09.02.2017)

Detail improvements:

- General bug fixes

5.9.153 v6.6 (26.01.2017)

Detail improvements:

- General bug fixes
- {$FORM:<id>} Variables are now usable in website objects

5.9.154 v6.5 (12.01.2017)

Detail improvements:

- General bug fixes
- Entry limit now also works in Auto-Submit forms

5.9.155 v6.4 (29.12.2016)

Formular:

- New, optional protection mechanisms protect against duplicate and automated entries

Detail improvements:

- General bug fixes

5.9.156 v6.3 (15.12.2016)

Detail improvements:

- General bug fixes
- The form now optionally supports an automatic address correction

5.9.157 v6.2 (2.12.2016)

Detail improvements:

- General bug fixes
- In the Importer a / character can now be used again as separator for selection options

5.9.158 v6.1 (18.11.2016)

Detail improvements:

- General bug fixes

5.9.159 v6 (04.11.2016)

Changelog V6:

- Campaign Designer - Visual configuration of campaigns and nurture routes for fully automated campaign execution.
- LeadPage - Visual editor for building personalizable landing pages
- Conversion Funnel - Statistical evaluation of campaign successes
- Calendar - display and subscription of important campaign data
- World Activity Globe - 3D visualization of scoring data
- Persona Match - Content (article) rating based on defined personas (test profiles)
- Pool Alias - Client-capable address management and segmentation
- Enhanced email statistics - more detailed display of browsers, mail clients and devices
- Shipping time optimization - behavior-based determination of the optimal shipping period
- Enlargement of international federal states
- Neues Zielgruppen-Statements: „Gleicher Monat wie“
- Manage container content via SOAP

- Container Connection in Dynamic SVG Screens
- Container connection in eMailings
- New Client Checks (Litmus) - Latest mobile devices
- Performance Optimization of Forms
- Tracking-revocation on profile level
- New system language Spanish
- Numerous bug fixes and infrastructure improvements

Partner applications.

- SMART DIRECT MAIL - Print letter delivery
- BUSINESS INTELLIGENCE - Data visualization
- INBOX MONITORIG - Control of mail delivery
- SALES COCKPIT - Lightweight Lead CRM

Detail improvements:

- Error in flowchart when using certain objects fixed
- General bug fixes

Note: Some of the functions contained in this manual have been partially played out in advance.

5.9.160 v5.21 (20.10.2016)

Detail improvements:

- General bug fixes

5.9.161 v5.20 (06.10.2016)

Detail improvements:

- Users configured in an acceptance process can now be deleted
- General bug fixes

5.9.162 v5.19 (22.09.2016)

Detail improvements:

- If the article template for an article is changed, the change is immediately displayed again.
- General bug fixes

5.9.163 v5.18 (08.09.2016)

Pool

- Federal states for Australia added

Detail improvements:

- General bug fixes

5.9.164 v5.17 (25.08.2016)

Detail improvements:

- General bug fixes

5.9.165 v5.16 (11.08.2016)

eMailing:

- When creating a new container, any existing default from a container type set in the eMailing template will be taken over

Detail improvements:

- General bug fixes

5.9.166 v5.15 (28.07.2016)

Target groups:

- New statement *is same month as*

5.9.167 v5.14 (14.07.2016)

Detail improvements:

- General bug fixes

5.9.168 v5.13 (30.06.2016)

Detail improvements:

- General bug fixes

5.9.169 v5.12 (16.06.2016)

Detail improvements:

- General bug fixes

5.9.170 v5.11 (02.06.2016)

Interface:

- Container contents can be managed via SOAP Api

LeadPage:

- Link tracking integrated

Detail improvements:

- General bug fixes

5.9.171 v5.10 (12.05.2016)

Dynamic image:

- Selection of a container that can be accessed via TAL

Event eMailing:

- Extensive checks at email dispatch

5.9.172 v5.9 (14.04.2016)

Detail improvements:

- Funnel chart can be integrated within reportings
- Funnel-Chart can be added as a widget on the dashboard
- General bug fixes

5.9.173 v5.8 (24.03.2016)

Campaign calendar:

- Visualization of dispatch times and individual events

Funnel chart:

- Visualization of leads

eMailing settings:

- Use of configurable fields in eMailing for easy provision of content

5.9.174 v5.7 (03.03.2016)

LeadPage object:

- Design and configuration of dynamic LeadPages by Drag & Drop
- Management of LeadPage templates via a dedicated LeadPage template object

Detail improvements:

- New client checks
- Statistics of event- and trigger eMailings can be deleted
- Tabular display of browsers, mail clients and end devices in the eMailing statistics

5.9.175 v5.6 (11.02.2016)

Extension of the eMailing statistics

- The Double-Opt-In confirmations made via eMailing are now shown in the statistics.

Detail improvements:

- General bug fixes

5.9.176 v5.5 (21.01.2016)

Visual editor for NewsManager:

- The contents of the NewsManager can be edited using a visual editor

Extension of the profile history:

- SmartLinks and touchpoints that have been visited are now displayed in the profile history

Detail improvements:

- General bug fixes

5.9.177 v5.4 (10.12.2015)

Performance:

- Performance optimizations in the area of web forms

Detail improvements:

- Object search at global level now also searches in clients
- Forms can also now be integrated into website objects by replacement variables
- Trigger-Mails can now also be sent via SOAP
- Statistics of trigger mails, event mails and split campaigns are now also available in the Reporting API
- Cross-client data retrieval by reporting API
- Removal of many minor errors

5.9.178 v5.2 (29.10.2015)

Disabling of tracking at profile level:

- Personal tracking for an individual profile can be disabled by means of a new pool field *TRACKING_DISABLED* (true/false) in the pool.

Integration of forms in NewsManager:

- Forms can be inserted into a NewsManager object by means of the new replacement variables *{$RENDER_FORM:ID}*.

Conversion Tracking controllable

- Conversion Tracking can be fully disabled in the settings in order not to set Conversion Cookies anymore.

5.9.179 v5.1 (08.10.2015)

Persona Matching for Content Marketing:

- Evaluation of the relevance of your content with respect to your articles for defined persona profiles (a profile requires a persona flag as property as well as enabled pool fields for Content Profiling).

Entry reporting web form:

- Request forwarding of new entries and changes to web forms. Simple form of the Request Manager for freely definable eMailings and automatic transmission logging.

Request Manager:

- Individual report mailings for free definition in the Request Manager can be used.

Send Time Optimization:

- Dispatch time determined automatically based on the click behavior of recipient profiles in the past for the dispatch periods 24 hours and 7 days.

Web Services:

- For various new SOAP methods, see documentation.

Target groups:

- New target group statement "vicinity search" for geographic evaluation of profiles.

Statistics for trigger eMailing:

- Trigger eMailing with its own statistics (e.g. for sending web forms).

Website Scoring:

- WebTouchPoints for tracking and scoring external websites for mapping integrated scoring. Requires activated scoring functionality.

Interface:

- Spanish as new system language.

5.9.180 v5.0 (13.10.2014)

- Nokia Maps for dashboard and statistics
- Scoring (Profile- und Activity-Scoring)
- Progressive Profiling in web forms
- Write protection for prefilled pool fields in web forms
- Content profiling to show content according to profile traits
- Profile tagging to show profiles according to their behavior
- Free DKIM keys and configuration
- Smarter, self-learning recognition of client-agents
- Target group statements for the amount of selected options in a multiple choice
- Target group statements for activity and profile scoring
- Improved scalable demo environment
- New Dashboard widgets
- Special reportin API for external business intelligence statistics

5.9.181 v4.11 (20.03.2014)

- New fusion charts for reports and mail transfer statistics
- New fields for article- and container types
- Containerfield "Code-Macro" now with improved TAL functionality
- Mandators can now be copied more easily
- Version-update of the CK-editor
- Improvement of the load balancing for better performance
- Smartlink: field dates now have a string-to-date field
- Amount of the input-, HTML- and textfields has been doubled
- Automatic forwarding to the secure https connection in login
- API: Mass-update now possible via SOAP
- API: getGrantedPermissionsSince via SOAP
- API: getModifiedProfilesByPool via SOAP
- API: getMailingTemplates via SOAP
- API: addToBlacklist via SOAP
- API: Improved methods to configure mailings with SOAP
- API: Methods to manage articles within a mailing with SOAP

5.9.182 v4.10 (2013-07-20)

- Individual configuration of manual export of profiles via drag and drop
- New and optimized check for sending
- New input elements for date values and numbers
- Creating eMailings using 3 different views (carousel, gallery, cover flow)
- New Nokia maps with grouped view and heat map
- Sorting form fields via drag and drop
- Various optimizations for increasing the performance and stability of the system
- Modified target group view
- Optimized dispatch preview
- SmartLink fowarding can now be made depending on browser and platform
- Forms now indicate option ID and internal name of pool fields
- Duplicate check can be adjusted in the eMailing
- Report is displayed when clicked and not automatically when opened
- New target group operator for e-mail fields: is equal to
- New button on dashboard for article and mailing
- Activation/deactivation of articles and container types for creating new items

- Modified target group configuration
- The check for individualisation is no longer displayed with missing rights in the list of profile settings
- When the sending time is set in the past, an error message is displayed in the sending time configuration
- Hard and soft bounces can now be exported in eMailings
- Objects with favourite flag can be used in the object picker
- Trial accounts can be directly activated in the customer account
- Establishing pilot environment: pool: post code field has a correct data type in the new customer account
- hasValue functions with listing attributes
- Entry limit in the form now refers to identified entries
- Preview tab of a sent cron mailing no longer generates openings
- Upload limitation of 10MB introduced
- Compatibility update of the HTML editor
- Visual editor: IE10 problems solved
- Speed-optimized list of dispatches
- Safety: sessions become invalid across domains
- API: web services: set/read cursor for collecting data
- API: profile IDs and time stamps of all clicks and SmartLinks are read
- API: web services: single sign on token solution
- API: New user/-manipulation possible
- API: Blacklist is considered with doSendEmailing
- API: Modification of WSDL type to literal
- TAL: getBrowser also available with the form

5.9.183 v4.9 (2012-11-17)

Interface

- In the templates, or directly when transmitting, you can determine which values and diagrams should be automatically appended to the reports.
- The keyboard actuation triggers the correct action within the tree navigation (new, rename, insert)
- Durch unabsichtliches wiederholtes Klicken der Schaltfläche "Versenden" kann nun kein mehrfaches Versenden von eMailings ausgelöst werden

SmartLink

- In the statistics, the number of clicks is now displayed cumulatively
- When a user ID is transferred, the fallback applies if none of the target configurations are relevant

Web Services

- New pool fields can now be created
- All the statistics of a SmartLink can now be retrieved with a single request

5.9.184 v4.8 (2012-10-06)

Highlights

- Zip files, including the file structure and content, can be unzipped and transferred (imported) by the system

Interface

- You can select any number of entries in the blacklist by checking them
- The time is displayed when an e-mail entered in the send queue
- The picture gallery can be sorted in the object picker
- The aspect ratio of preview images has been corrected
- The CSV export in the send overview now contains the entire subject line
- A URL picker for SmartLink links has been integrated
- By means of a drop-down list, you can navigate to a page directly in the profile and object lists
- The creation and activation date of a client is now displayed
- Favorites are now uniformly named within the system
- The profile update icon of SmartLinks is now displayed as being active when pool attributes are set for the update
- Activated event mailings can be now be found in the object pop-up within subfolders of mailings
- In mailings, the color codes for the best and worst values are corrected in a branch comparison

eMailing

- After an article has been exchanged, the expanded editor contains the respective article template configuration

Article

- Articles now indicate the underlying article types

SmartLink

- You can now create SmartLinks in web folders as well

Targetgroup

- In a target group, you can query whether a profile has activated at least n options of a multiple selection
- As long as mailings are part of a target configuration, they cannot be deleted
- Die fehlerhafte Darstellung der Konfiguration bei „beinhaltet irgendeine“ wurde behoben

Security

- The validity period of the temporary password was extended to 24 hours

Web Services

- The transmission speed can be adjusted by XML-RPC

5.9.185 v4.7 (2012-08-25)

Interface

- Standard pop-ups can be closed using the escape key

eMailing

- A button for stopping a mailing in an emergency has been integrated
- The mailing confirmation contains a mailing name, the number of recipients and a subject line
- The variable IMG_SRC automatically appends the UID to each picture
- New TAL method for generating a unique anchor name for internal linking

Targetgroup

- New target group selection: Number of hard bounces
- Das Zielgruppenkriterium „Ursprungsformular“ wurde um Leer-Selektion erweitert

Form

- E-mail addresses that are in the blacklist can no longer be used for entry by means of a form

SmartLink

- SmartLink enters the change date in profiles when the data are updated

Users

- In the user settings under role assignment, the location of each role in the system is displayed

Template Attribute Language (TAL)

- You can access the user agent via TAL (website, SVG, mailing)

Web Services

- Article and container types can be read out by means of XMLRPC
- Landing page links of articles can be read out or set via XMLRPC
- The XML-RPC function doProfileUpdate presently also functions with encoded user IDs
- Via XMLRPC, you can access a list of all clients of a login
- SmartLink targets can be created via XMLRPC

5.9.186 v4.6 (07-14-2012)

Highlights

- **SVG-Object** - Generation of dynamic SVG (Scalable Vector Graphics) images via a new object

Interface

- Drag&Drop: When exchanging images in the visual editor, the storage location of the old image is used for the new image, too

eMailing

- Google Plus Sharing has been integrated

Container and article types

- Container list as article and container type field

Template Attribute Language (TAL)

- String manipulation e.g. Use of upper and lower case in text fields of article and container objects
- The website object now processes replacement variables or TAL for personalized websites
- Option to parse the content of multi line inputs as table

Web Services

- Uploading/Reading out images

5.9.187 v4.5 (04-28-2012)

Highlights

- **New dashboard** - with customized widgets and can be subdivided into any columns. Additionally, external data sources can also be integrated into iFrames.
- **HTML 5 - All diagrams** are fully HTML 5 compatible, i.e. the print output of diagrams and charts is supported as well as iPhone and iPad.
- **Engagement Analysis** - with reading time measurement for meaningful engagement analysis of eMailings
- **Spellchecking** - Spell-checking and grammar checking of DUDEN, the Institution for spellchecking in the German-speaking world
- **Extended Drag&Drop** - any number of images or files can be imported with one mouse click.
- **Live-View 24** - Real time display with live statistics of the first 24 hours of the transmission
- **Smart Link** - Multifunction object for link tracking, target group-specific routing and query manager
- **Web DAV Interface** - edit your templates directly with Dreamweaver, Eclipse or other WebDAV-capable editors.

eMailing

- Visual subject line check and preview for distributed e-mail clients
- Tracking of URLs can be activated individually
- Status display for transmission overview lists
- Checking whether the text variant of an eMailing exists
- Article detail information is displayed in the visual editor
- Use of alternative eMailing templates

Interface

- Many design and usability improvements
- Context-related help for all system components
- Fullscreen mode
- Integration of the search function in Object Picker
- Status/progress message during deletion of objects
- Display of failed login attempts on start page
- Redesign of the rights/roles configuration with improved usability

General

- Creation of a completely empty pilot environment is possible
- In the event of locked objects, external access is routed to individual URLs
- File size is displayed in overview lists
- Query Manager: Annual transmission option

Statistics

- Integration of statistical link evaluation for article evaluation
- Automatic statistics transmission after transmission of eMailings
- Export of conversion tracking data of an eMailing

Form

- reCaptcha for web forms has been integrated
- Registration limits for time and number of registrations

Targetgroup

- Target group configuration visually redesigned
- New target group criterion for number: Is not the same
- New target group criterion for: Has received/opened X eMailings in period Y

Pool

- IP address as separate field type in the pool
- Data field "federal state" now with Chinese provinces

Split campaign

- Additional transmission times can be selected

Container and article types

- Container objects can now be nested
- URL field for container
- Lists for images and key value list in container
- Containers support code macros(TAL)
- Data fields can be hidden

Users & Profiles

- User accounts can be limited to clients
- Profile ChangeLog is displayed in the profile details

Security

- Extended security guideline for forcing special characters for passwords
- New Session Management (security guidelines: secure cookie, deletion of sessions during login, session time-out)

Template Attribute Language (TAL)

- TAL functions for access to options of containers and articles
- TAL function for access to SLOT-ID in the article
- Link tracking also for content from external XML

5.9.188 v4.0 (07-30-2011)

HIGHLIGHTS

- **Dynamic article sort** - Articles for newsletter campaigns can now be classified with relevance attributes indicating which target groups they can arouse the greatest interest for.
- **Multi-variant split campaign test** - The Split Test determines which newsletter variant is the most successful in advance based on defined success parameters and then provides the remaining recipients automatically with this variant.
- **Dynamic Content Management** - The new version now allows different article objects to be customized in a newsletter. Examples include articles for seminar offers, shop articles, press reports, events and training courses.
- **Newsletter Composing per Drag & Drop** - Die Artikel können nun noch komfortabler und effizienter per Drag & Drop mit der Maus angeordnet werden und das entsprechende Artikelerscheinungsbild (bspw. Bild links, Bild rechts, usw.) passt sich „on-the-fly“ an.
- **QR-Code Generator** - Texts, personalization variables as well as a URL can now be created with the QR Code Generator for iPhone, Blackberry or Android in machine-readable form.
- **Additional security with security key** - Two-factor authentication via the USB Security Key is possible from now on. The Security key can also be used to authorize access outside the IP address range of the company.
- **Formal acceptance & approval** - Responsible persons and their representatives who must grant their approval before a transmission can be designated.
- **Mobile Check for Google Android** - Checking representation of the newsletter on mobile systems with Google android.
- **Differentiated Eye-Tracking** - Any number of eye-tracking areas can be defined and then analyzed differentiated. In this way, for example, the impact level of different designs of newsletter elements can be measured.

SECURITY

- Possible to create your own customized security guidelines
- Locking the account after 5 failed login attempts
- User Management: Identification of fields that contain incorrect values
- “forgotten password” function implemented
- User Management: automatic password assignment with password email to user as an option
- When assigning a new password, this cannot be used for the password last used
- Validity period of a password can be set by means of a security guideline
- Use of a security key (two-factor authentication via USB key) for optional authentication

INTERFACE

- Additional menu box with bookmarks and history of the objects last used
- Objects can be added to your bookmarks by drag & drop
- Cross-client bookmarks possible

EMAILING

- Multi-variant split campaigns with automatic decision
- Function for the acceptance and approval of eMailings implemented by authorized users
- Automatic article sort by means of adjustable criteria

- Adjustment options for slot sort integrated in template
- “Web/Mobile” view inserted
- Social Sharing for Linkedin
- Visual Editor: cross-slot movement of articles possible
- Separate identification of local and global templates when creating a new eMailing
- Possible to select prefix yourself for identification of test transmissions
- When moving articles, a placeholder displays the current or new position of the article
- Email check extended for Android
- Integration of DKIM (Domain Keys Identified Mail)

ARTICLE

- Creation of a flexible article object (container) for central storage of translations, TAL macros, text modules, etc
- Article type and container type can be created / maintained at global level
- Tool list for editing texts in articles extended
- Function for generating QR codes implemented

MEDIA

- Image server supports cropping of images in realtime
- Areas within an image can be cut out and put together on the fly
- Edges of images can be represented in templates rounded off automatically

POOL

- Dependencies on attribute options are shown again
- A change of the email address is registered in the Profile Change Log as well
- Creation of pool notes possible
- Profiles can be searched for by profile ID

NEWS MANAGER

- News manager with its own article statistics
- Controllable by its own role

QUERY MANAGER

- Select or deselect all field checkboxes with a single click

PROFILE IMPORT

- Rescue Center: Backups are also created at global level

REPORT

- Pool-Data-Miner can be used on global pools

PILOT ENVIRONMENT

- Pilot environment for two new article templates, LinkedIn, mobile view and QR code example updated

TAL-API

- TAL method for access to newsletter subject line

- Direct output of the date format possible
- TAL function implemented for displaying articles in any columns

WEB SERVICE

- Articles can be created by means of XML-RPC interface

CHAPTER 6

Statistical parameters

6.1 Transmission

- *Subject* Subject line of the e-mailing
- *Target group* Target to which the e-mailing was sent
- *Start* Date and time of the start of transmission
- *Progress* Current transmission status in percent
- *End* Date and time of the end of the transmission
- *Initiator* Email address of the user that sent the e-mailing

6.2 Impressions

- *Openings* Number indicating how often the e-mailing was opened by recipient(s) **o**

  gross (1st value) = Sum of all openings

  net (2nd value) = explicitly identified users

- *Opening rate* Percentage of recipients that opened the e-mailing at least once

  $$\frac{o(netto)}{e} * 100$$

- *Opening rate POR* Forecast percentage of recipients that opened the e-mailing at least once. This value is based on the fact that the opening rate must consider only explicit openings of the mailing so that profiles without an opening but only a click and that should logically be followed by an opening, are ignored. The ratio between clicking openers, non-clicking openers and clicking non-openers is used to determine an estimated opening rate.
- *Multiple opening rate* Percentage of recipients that opened the e-mailing more than once

$$\frac{o(brutto)}{e} * 100$$

6.3 Clicks

- *Clicks*

  Number signifying how often any links in the eMailing were clicked on by the recipients:**k**

  gross (1st value) = Sum of all clicks

  net (2nd value) = explicitly identified users

- *Click rate (customer activity indicator)* Percentage of recipients that clicked any of the links in the e-mailing at least once

$$\frac{k(netto)}{e} * 100$$

- *Relative Klickrate (Indikator für Kundenaktivität)* Anteil der öffnenden Empfänger, die mindestens einmal auf einen beliebigen Link im eMailing geklickt haben

$$\frac{k(netto)}{o(netto)} * 100$$

- *Multiple click rate (indicator for quality)* Share of a recipient in the sum of all clicks

$$\frac{k(brutto)}{e}$$

- *Relative multiple click rate* Percentage of an opener in the sum of all clicks

$$\frac{k(brutto)}{o(netto)}$$

6.4 Profiles

- *Addressees* Total number of recipients of the target group, **a**
- *Redundant entries* Number of redundant entries (differentiating criterion: e-mail address), **d**
- *Blacklist* List of excluded recipients on customer level, **b**

- *Robinson lists* List of excluded recipients on country level, RTR list for Austria, Robinson list DDV, **r**
- *Current hardbounces* Emails that could not be delivered, e.g. due to an incorrect email address (permanent failure cause), **hb**
- *Current softbounces* Emails that could not be delivered, e.g. due to an inbox that has exceeded its capacity (temporary failure cause), **sb**
- *Recipients* Actual group of recipients

$$e = a - (d + b + r + hb + sb)$$

- *Unsubscriptions* Recipients that have canceled their subscription in the distribution list, e.g. via e-mailing (Unsubscribing the newsletter via {$REVOKE_URL}), **ab**

6.5 Domains

List of domains where the eMailing has been sent

- *Rate of addressees* Percentage of recipients of a domain relative to the total number of all recipients

$$\frac{dom(n)}{e} * 100$$

- *Successful* Percentage of e-mails successfully delivered to the recipients of a domain

$$\frac{dom(n)}{n(erfolgreich)} * 100$$

- *Current softbounces* Percentage of softbounces for recipients of a domain

$$\frac{dom(n)}{n(aktuelle\ SB)} * 100$$

- *Current hardbounces* Percentage of hardbounces for recipients of a domain

$$\frac{dom(n)}{n(aktuelle\ HB)} * 100$$

- *n* Numeric value displayed in the statistics, e.g. (10)

6.6 Media

Tracking the display formats clicked on in the eMailing, e.g. Web, PDF or RSS Version, Text-to-Speech of the eMailing

- *Clicks* = number of times a display format was clicked by the recipients in eMailing, **k**, gross (1st value) = sum of all clicks, net (2nd value) = uniquely identified user
- *Klickrate* = Percentage of recipients who have clicked at least once on a display format in eMailing, **k(netto)/e * 100**
- *Relative click rate* = Percentage of recipients who have opened eMailing at least once and clicked on a display format in eMailing at least once, **((netto)/e)/(o(netto)/e)) * 100**

6.7 Article

Click tracking of articles and display formats that have been clicked on within the context of the article, e.g. article detail page (landing page), PDF version of the article etc.

- *Clicks* = Number of times an article was clicked on by the recipients in eMailing, **k**, gross (1st value) = sum of all clicks, net (2nd value) = uniquely identified users
- *Click activity* = Number of times an article and its different display formats have been clicked on by the recipients in eMailing, **Sum(k(formats);(k(gross) / k(netto)=unique))**
- *Klickrate* = Percentage of recipients who clicked at least once on the article in the e-mail, **k(netto)/e * 100**
- *Relative click rate* = **Click rate (article)/Opening rate**, percentage of recipients who have opened the eMailing at least once and clicked the article at least once in the eMailing, **((k(netto)/e)/(o(netto)/e)) * 100**
- *Multiple click rate* = one recipient's share of the total of all clicks of this article in this mailing, **k(gross)/e**
- *Relative multiple click rate* = **Clicks(articles)/openings**, a recipient's share of the sum of all clicks of this article in this mailing, **((k(gross)/e)/(o(net)/e))**

6.8 Links

Click tracking of all external links that have been clicked on in the eMailing, e.g. link to homepage

- *Clicks* Number of clicks that precisely this external link has received from the recipients in the e-mailing, **k**, gross (1st value) = total number of all clicks, net (2nd value) = clearly identified users
- *Click rate* Percentage of recipients that clicked exactly this external link in the e-mailing at least once, **k(net)/e * 100**
- *Relative click rate* Percentage of recipients that opened the e-mailing at least once and that clicked precisely this external link in the e-mailing at least once, **((k(net)/e)/(o(net)/e)) * 100**

6.9 Map

Geographical visualization of the profiles. Current limitation: Europe. The geographical information is generated from the address data if available. If no address data is available, the IP address of the subscription is used provided that the profile was entered via a form.

6.10 Cleared statistics for tracking revocation

- *tracking contradiction* = number of recipients with active tracking contradiction
- *Revocation rate* = Tracking contradiction / addressees
- *Opening Rate Adjusted* = Openings-Unequivocal / (RECIPIENT*(1-REFUAL RATE)
- *Opening rate POR Cleaned* = (((Openings-Only / Clicks-of-Openers) * (Clicks-of-Not-Openers)) + Openings-Only) / (RECIPIENT*(1-REFERRED RATE)
- *Click Rate Adjusted* = Click Unique / (RECIPIENT*(1-REFERRED RATE)
- *Relative click rate adjusted* = Clicks unique / Opener unique

CHAPTER 7

Glossary

7.1 A

ASP Application Service Provider - The possibility of renting software applications temporarily via the Internet. The software runs on servers in the computer center of the service provider and can be used via the Internet using a browser or special user interface.

Attachment File attached to the email.

Autoresponder A program that answers enquiries automatically by email. Frequent applications are the transmission of product and service information, confirmations of receipt or absence notices.

7.2 B

Blacklist List with email addresses which are excluded from future mailing campaigns.

Body The body is the actual content of an email.

Browser Software for the retrieval and display of HTML documents in the World Wide Web (WWW).

Bounces Mails which are returned to the sender due to transmission errors or for other reasons. A distinction is made here between soft bounces and hard bounces.

7.3 C

Click-Through-Rate The Click-Through-Rate (CTR) is a performance indicator in Internet Marketing, which depicts the percentage of clicks on an advertising banner in relation to its overall impressions. If an advertisement is displayed a hundred times and is clicked on once during that time, the click-through-rate is 1%.

Closed Loop Marketing This term describes the closed circuit that the marketing communication can have with a user in the best-case scenario. Based on the measurement of campaign data such as openings, clicks and purchasing decisions a profile of the user is created automatically. This can be accessed again when developing the next campaign in order to show contents to the user only relevant to him, for example.

CMS Content Management-System - Software for creating, managing and publishing content. Not only can websites be supplied with current content by a CMS, but the content of email newsletters can likewise be created, too.

Confirmed Opt-In With this method, the newsletter subscriber receives a written confirmation of his subscription before sending the first newsletter which also contains an immediate cancellation option.

Conversion The term Conversion-Tracking describes the part of the website analytics that tracks the effectiveness with which an addressed group of people are motivated to perform desired actions. Types of Conversion Rates:

- Order conversion rate - the ratio of orders to visitors
- Basket conversion rate - the ratio of orders to baskets
- Order abandonment rate - the ratio of orders to commenced order processes
- Visitor conversion rate - the ratio of visitors of an explicit website (webpage) to visitors of the website
- The ratio of newsletter subscribers to readers as well as (because of the newsletter) to visitors (click-through-rate) and to buyers (order conversion rate of the newsletter)

CRM Customer Relationship Management - "Customer Relationship Management". Customer Relationship Management refers to handling of customers from the time of acquisition to keeping the customer loyal to your own company as a regular customer with various marketing measures.

7.4 D

Database Marketing Marketing measure based on information from special address databases.

Direct marketing Marketing measures by means of direct communication and personal conversations of the target group.

Double Opt-In Special application procedure whereby the subscriber after his application must agree explicitly to the subscription in a second step in order to receive emails.

Duplicates Addresses that can occur twice or even multiple times in databases.

7.5 E

eMarketing The term eMarketing comprises marketing via various channels such as Internet and mobile phone. Besides the most well-known variety of email marketing, eMarketing also includes advertising via SMS, WAP messaging as well as Voice/Text-to-Speech Marketing.

eMail Format Technical format in which an e-mail is sent. The most common formats are text, HTML and Multipart.

eMail Marketing Communication with active or potential customers via email. The most rapidly expanding segment in direct marketing.

7.6 F

Feedback analysis The transmission statistics are also described as feedback analysis

Firewall Software ad/or hardware for protection against unauthorized access to hosts.

Forwarding Describes the forwarding of received emails to other persons.

7.7 H

Ham The opposite of Spam. This means desired electronic post.

Hard Bounce Hard bounces are all permanently undeliverable emails. The most frequent reason is because the user in question is no longer registered with the provider that the email is addressed to.

Header The header is the "letterhead" of an email so to speak. This contains, among other things, details about the sender, recipient, transmission date and subject line.

HTML Format An HTML email is written in the page description language HTML known from the web, and therefore closely resembles a website. Consequently, HTML emails can also include graphics - this ensures a greater graphic benefit as compared with text emails.

7.8 I

Customization A personalization technique used to customize text content according to the recipient's profile.

IP address Worldwide unique identification number of a computer in the Internet.

ISP Internet Service Provider - Service provider that provides access to the Internet.

7.9 K

Purchase rate Percentage number of recipients of a mailing that have subsequently made a purchase on the website of the advertising company (or another action, e.g. download of information material). The purchase rate can be measured during the postclick tracking.

Click rate Percentage number of clicks on hyperlinks based on the total number of emails transmitted during a mailing. The clickrate is a good indicator for the attractiveness of the content/offer linked to from a mail.

Conversion rate Percentage ratio between clicks on a link and the purchases made afterwards. The number of prospective customers (clickers) is compared with the number of purchasers "conversions". Also see Conversion Rate.

7.10 L

Link Tracking Analysis and logging of the recipient ratio by automatically tracking and counting the links clicked on within an email.

List broking Within list broking, a service provider (list broker) mediates between companies interested in transmitting advertising emails and companies that offer lists for rental with addresses of users who have expressed interest in receiving advertising emails.

7.11 M

Mail Client A program for reading and writing emails. (e.g. Microsoft Outlook, Eudora, etc.).

Emailing In email marketing, we speak of mailing if a marketing message - e.g. newsletter - is sent by email to a list of recipients.

Mail server Computer in the Internet with main task of receiving and transmitting emails.

MIME Multipurpose Internet Mail Extension - A format description enabling a classification of attachments.

Multipart Format If several email formats (e.g. text and HTML) are sent at the same time in a mail, we call this multipart-mails. The mail client of the user determines which version is shown. Even in the case of old or restrictively set mail clients, the text version at least is always shown even if HTML cannot be represented. This ensures that the user always receives a readable mail.

7.12 O

Offline HTML An email in offline HTML mode contains all necessary information that is needed for displaying the mail, i.e. images and style sheets. The emails can therefore be read without there having to be a connection to the Internet at this time.

One-to-One Marketing Direct, personal communication from one person to another as with email marketing. In contrast to "One-to-Many", whereby one person communicates at random with many, as with poster advertising.

Impression rate Actual reach of an eMailing without taking passive recipients into account who do not load images or click.

POR opening rate Berechnete Reichweite eines eMail-Versands unter Berücksichtigung des Verhältnisses Klicker pro Öffner. Beispiel: 100 Profile haben geöffnet und davon haben 10 auch geklickt. Dies entspricht 10%. Nun gibt es zusätzlich auch noch 10 Klicker, welche nicht geöffnet haben. Daraus ergeben sich statistisch weitere 100 Öffner, welche nicht gemessen werden konnten. Also geht die POR von 200 Öffnern aus. Die POR Öffnungsrate ist nicht explizit und daher nicht exakt, aber vom Ergebnis her genauer als die reine Öffnungsrate oder Klickrate.

7.13 P

Personalization A special technique for sending emails so that the text of the mail can vary per recipient. In this process, the text is only written once and variable data can be inserted from a database per recipient.

Permission Marketing Permission Marketing is based on the idea that the recipient grants the sender his permission for sending him advertising messages.

Postclick Tracking The measurement of actions of a mailing recipient that go beyond regular tracking. The measurement is based on all actions that occur after clicking ("post click") on a link in the mail e.g. online purchases or downloads. This can be achieved by a small modification to the website where the measurement is to take place.

Profiles Profiles are additional information to normal address data.

7.14 R

Response The incoming response to a direct marketing campaign of the sender.

Robinson list List of persons that do not want to receive any more advertising messages (e.g. by mail or email). These lists are mostly maintained by neutral institutions (branch associations or similar).

7.15 S

Soft Bounce Soft bounces are emails that are temporarily undeliverable, which could occur, for example, if a recipient‘s mailbox is too full.

Spam check Unsolicited messages and information by email. The official term is “unsolicited commercial email” (UCE).

Spam filter Mechanism in email clients such as Microsoft Outlook, Eudora and Pegasus Mail, which filter out the receipt of advertising emails.

7.16 T

Template In this context, this a template for an email with integrated customization variables.

Text format An email in text format only contains text. Formatting (bold type, different font sizes etc.) as well as images are not possible. These are reserved for the HTML format.

TRACKING Tracking in email marketing describes the measurement of the users‘ response to a mailing. A variety of success-relevant parameters can be measured, e.g. opening rate, click rate, unsubscribe rate and - within the context of post click tracking - the purchase rate, too. After the tracking data has been measured, it is prepared as a graphic report.

7.17 U

Unsubscribe If the recipient of a newsletter cancels the subscription, we call this an unsubscribe. Every marketing email should contain a clearly visible unsubscribe link to enable the user to stop any further mailings at any time. This is also a principle of permission marketing.

Usability In this context, the user friendliness or usability of an eMailing marketing system. Determining factors of the usability are the information architecture and user guidance.