Manual
for Contao extension "contao-pdf-nc-attachment-bundle"

This Contao extension extends the standard email gateway of the notification center in Contao 4 by the possibility to fill in a PDF template with the SimpleTokens which are sent to the notification center, to save it and to attach it to the mail.

The extension **contao-pdf-nc-attachment-bundle** installs itself as a limited demo. The demo version generates a demo hint in the PDF with full functionality and is limited to 2 output pages.
When you buy a license key, the restrictions are removed. The license allows you to use the extension in the domain specified at the time of purchase. The use in other domains requires an additional license.

In **version 2**, QR codes and image data from file management, upload or from a data stream can also be transferred to the PDF.

Contents

1. System Requirements

The extension is approved for Contao 4 from version 4.4
The Composer or Contao Manager is required for the installation.

If not available, the Notification Center (terminal42/notification_center) will be installed as well, because this is the basis for this extension.

2. Installation

Simply install the extension with the **Contao Manager**, search for

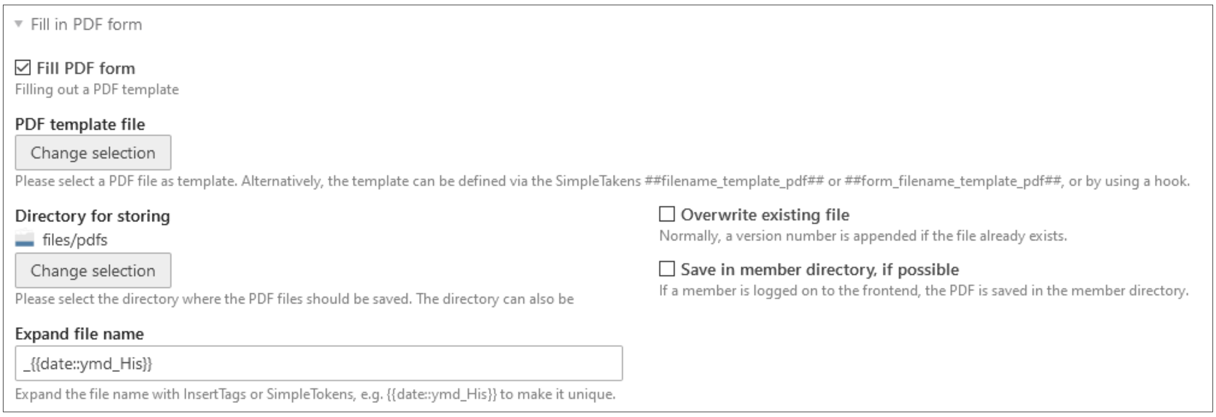
do-while/contao-pdf-nc-attachment-bundle

or on the command line with the **Composer**:

```
composer require do-while/contao-pdf-nc-attachment-bundle
```

3. Enhanced standard email gateway

After the installation has been completed, the Notification Center provides you with additional functions in the properties of the standard email gateway:
In the section *Fill in PDF form*, you can turn on the creation of a PDF file.



▾ Fill in PDF form

☑ Fill PDF form
Filling out a PDF template

PDF template file
Change selection
Please select a PDF file as template. Alternatively, the template can be defined via the SimpleTakens ##filename_template_pdf## or ##form_filename_template_pdf##, or by using a hook.

Directory for storing
files/pdfs
Change selection
Please select the directory where the PDF files should be saved. The directory can also be

☐ Overwrite existing file
Normally, a version number is appended if the file already exists.

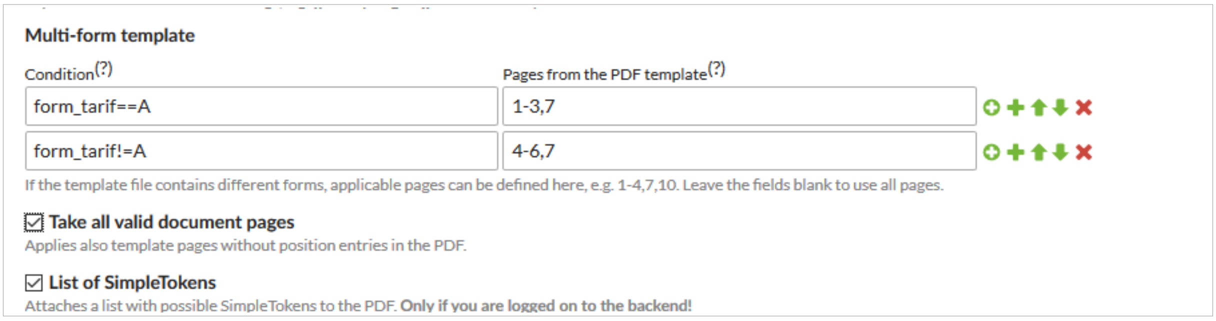
☐ Save in member directory, if possible
If a member is logged on to the frontend, the PDF is saved in the member directory.

Expand file name
_{{date::ymd_His}}
Expand the file name with InsertTags or SimpleTokens, e.g. {{date::ymd_His}} to make it unique.

You specify a **PDF template file** that includes an empty paper form. The form is filled out and saved as a copy with the entries. The page sizes of the template pages are copied.

To save the PDF files, you specify a **Directory for storing**. In this directory the complete generated PDFs are stored. Normally a version number is appended if the file already exists, with **Overwrite existing file** you can prevent this and let the existing file be replaced. With **Save in the member directory if possible**, the file is saved in the member directory for registered Contao members, if a member directory is defined.

The file name consists of the gateway title and a time stamp. You can adjust the structure of the file name in the **Expand file name** field. The default entry appends the current date and time. It is possible to use insert tags.
Only if the extension string starts with _ or - will the gateway title be prepended.



Multi-form template

Condition(?)	Pages from the PDF template(?)
form_tarif==A	1-3,7
form_tarif!=A	4-6,7

If the template file contains different forms, applicable pages can be defined here, e.g. 1-4,7,10. Leave the fields blank to use all pages.

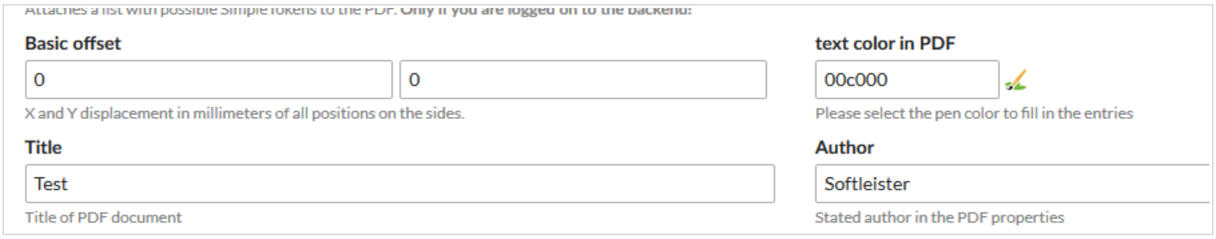
☑ Take all valid document pages
Applies also template pages without position entries in the PDF.

☑ List of SimpleTokens
Attaches a list with possible SimpleTokens to the PDF. Only if you are logged on to the backend!

It is possible to manage **multi-form template**s in one template PDF. To do this, all output pages are listed one after the other in one template PDF. By specifying the controlling fields as a condition, the output can be restricted to certain pages.
In the example there is a checkbox element where you can select tariff A or B. According to the selection in the form a SimpleToken ##form_tarif## is transmitted, which has the value "A" or "B". The first condition checks whether the token value is equal to (==) "A", then pages 1, 2, 3 and 7 are output, for tariff B the second condition applies, since the token value is not equal to (!=) "A", then pages 4, 5, 6 and 7 are output. Further conditions can also be used to control attachments in the PDF.
If you leave the multi-form template input fields completely empty, all pages, i.e. the complete template PDF, are used.

With **Take all valid document pages** you can control whether all pages from the template PDF are to be included into the output PDF, or only those pages on which valid positions are entered. So you have the possibility to create optional pages via the template. If a single empty page (e.g. AGB) is nevertheless contained, enter a position for this that prints a blank character. In the case of multiple templates, "All pages" refers only to the pages of the template that have been approved by the conditions.

To facilitate the creation a additional page with a **List of SimpleTokens** is possible. In the list you can see which SimpleTokens are sent to the PDF at all and can be used. In addition InsertTags in the position entries are also possible.
The list of tokens is only available if you are additionally logged into the backend during sending. For the normal web user this page will never be included.



Attaches a list with possible SimpleTokens to the PDF. Only if you are logged on to the backend!

Basic offset
0 | 0
X and Y displacement in millimeters of all positions on the sides.

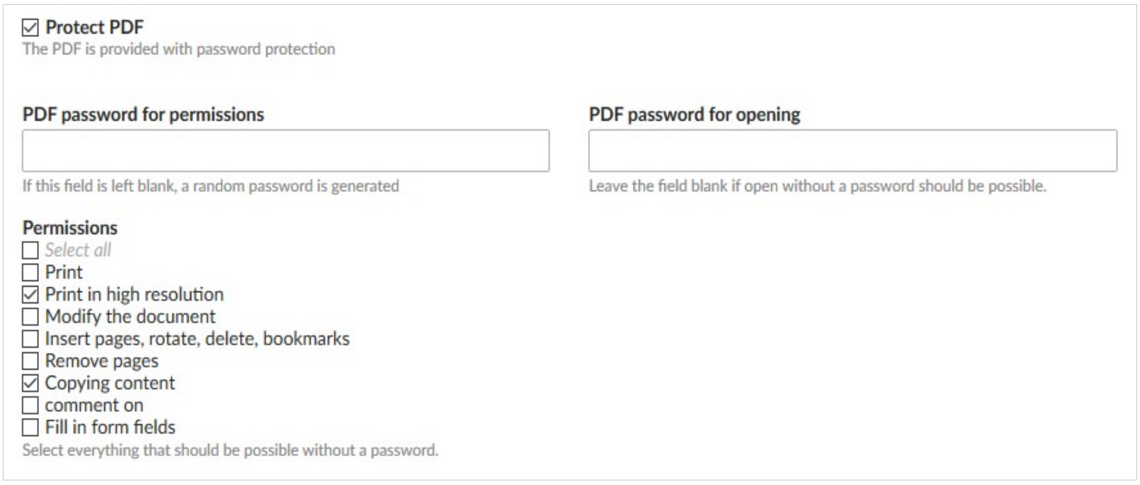
text color in PDF
00c000
Please select the pen color to fill in the entries

Title
Test
Title of PDF document

Author
Softleister
Stated author in the PDF properties

If you determine the positions of the entries, there are often differences to the edge of the paper. This offset is true for all positions of the PDF. To avoid having to adjust each item, you can just define a **Basic offset** by an entry in the gateway properties.

For the presentation, you can define the **text color in PDF** and the **Title** and the **Author** for the file properties.



☑ Protect PDF
The PDF is provided with password protection

PDF password for permissions

If this field is left blank, a random password is generated

PDF password for opening

Leave the field blank if open without a password should be possible.

Permissions
☐ Select all
☐ Print
☑ Print in high resolution
☐ Modify the document
☐ Insert pages, rotate, delete, bookmarks
☐ Remove pages
☑ Copying content
☐ comment on
☐ Fill in form fields
Select everything that should be possible without a password.

It is possible to protect the created PDF with a password. There are 2 ways:

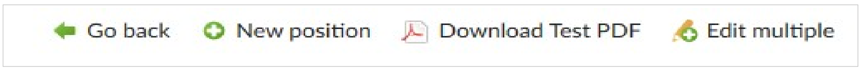
a) Password protection when opening the document

b) Restriction of privileges, select all allowed permissions. If the PDF password for permissions is left blank, a random password is generated. The password is then unknown.

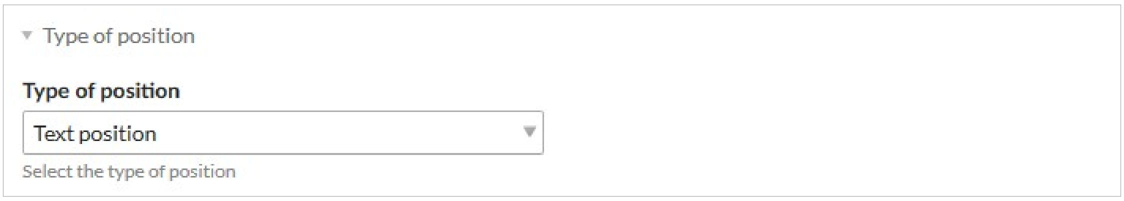
Note: The PDF protection is not completely secure. With appropriate programs, the password can be bypassed!

4. Definition of positions in the PDF

In the overview of the defined gateways there is an additional icon (PDF icon). The following table, the positions of the entries in the template PDF are created. Any number of positions can be created.



It is possible to create a completed test PDF file directly in the backend. This allows you to quickly check that all the positions are on the right places in the form. All published positions are output, form fields are printed with its field name.
Because no input data is available in the test output, the conditions are not checked, all fields and all pages are copied into the test PDF.



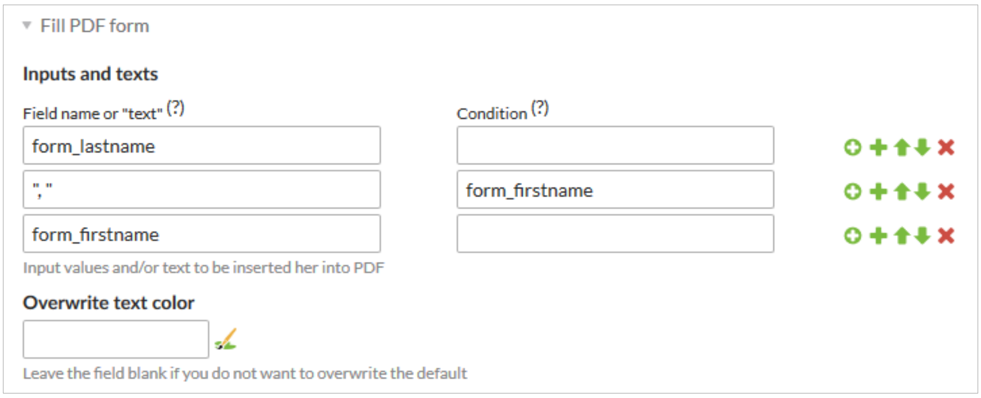
▾ Type of position

Type of position
Text position ▾
Select the type of position

From version 2 there are different types of positions:

• Text position	Output of texts, SimpleTokens and InsertTags
• Picture position	Output of image data
• QR code	Output of a QR code from texts, tokens, InsertTags

Creating the individual positions:
a) Text position



▾ Fill PDF form

Inputs and texts

Field name or "text" (?)	Condition (?)	
form_lastname		⊕ + ↑ ↓ ✕
", "	form_firstname	⊕ + ↑ ↓ ✕
form_firstname		⊕ + ↑ ↓ ✕

Input values and/or text to be inserted her into PDF

Overwrite text color

Leave the field blank if you do not want to overwrite the default

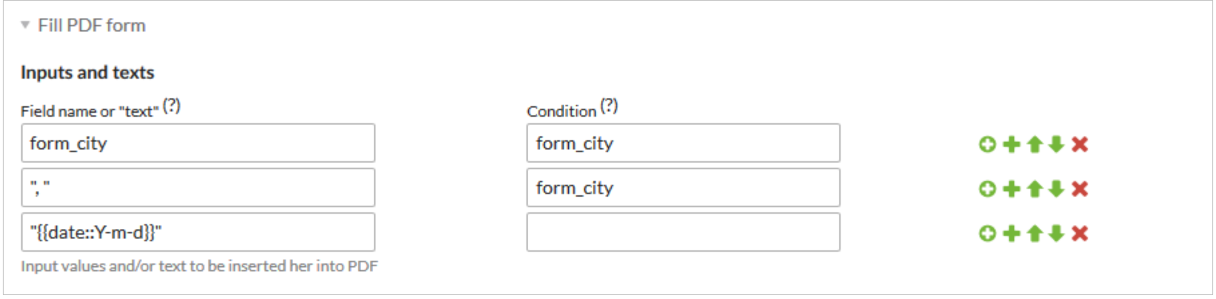
In the first section the text information are set. There are 2 options: A SimpleToken from the transmitted data or a fixed text in quotation marks.
In the example these are the tokens form_firstname, form_lastname and a fixed text with comma/space. This individual line definitions are lined up with spaces (no space before the comma).
In the 2nd column a condition can be programmed for the output. In the example, the comma depends on the fact that the token form_firstname exists and has a value.

Simple comparisons can also be made in the conditions, e.g. form_question==yes, if you want to test for a radio button, for example.

You can check for equality (==), two equal signs
or test for inequality (!=), exclamation mark, and an equals sign

The text is normally output with the color defined in the gateway properties. If you want to use a different color in this element, enter the color in **Overwrite text color**.

The fixed texts can also contain Insert tags:



▾ Fill PDF form

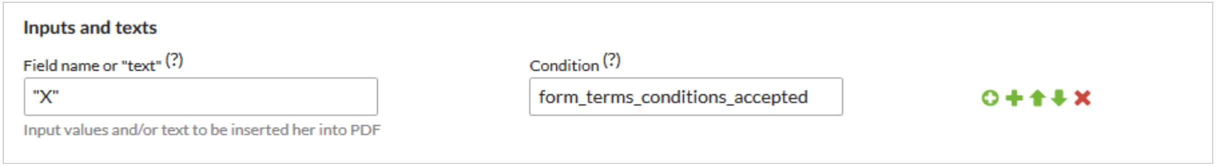
Inputs and texts

Field name or "text" (?)	Condition (?)	
form_city	form_city	⊕ + ↑ ↓ ✕
", "	form_city	⊕ + ↑ ↓ ✕
"{{date::Y-m-d}}"		⊕ + ↑ ↓ ✕

Input values and/or text to be inserted her into PDF

In this example, "Berlin, 2018-09-23" or only "2018-09-23" entered in the PDF, depending on whether **form_city** is specified or not.

To cross check boxes, we use an "X" with the checkbox value in the condition:

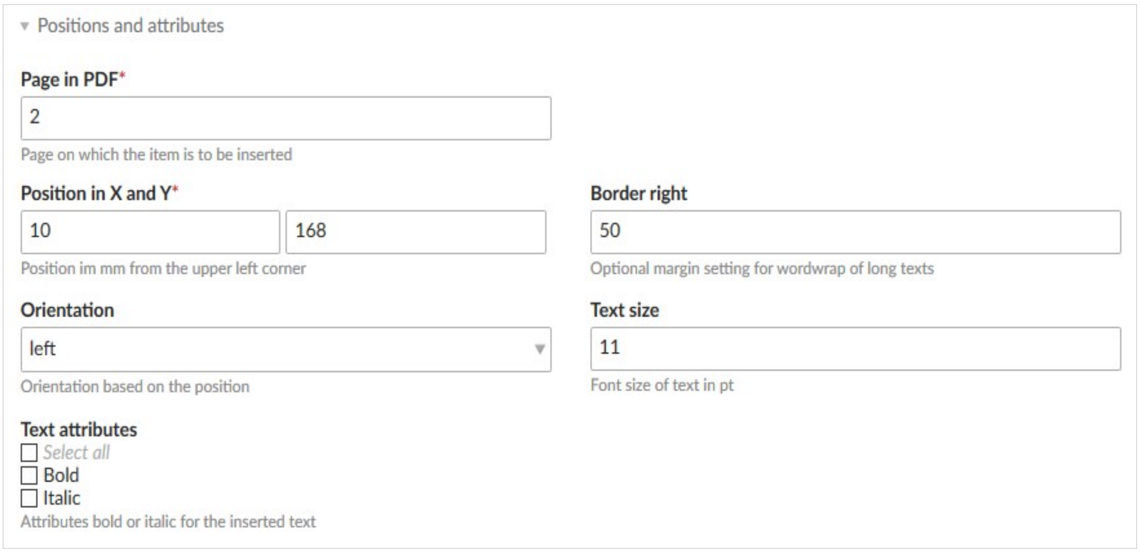


Inputs and texts

Field name or "text" (?)	Condition (?)	
"X"	form_terms_conditions_accepted	⊕ + ↑ ↓ ✕

Input values and/or text to be inserted her into PDF

In the lower part of the backend form it comes to the position in the PDF where the selected information is entered and the font attributes.



▾ Positions and attributes

Page in PDF*
2
Page on which the item is to be inserted

Position in X and Y*
10 | 168
Position im mm from the upper left corner

Border right
50
Optional margin setting for wordwrap of long texts

Orientation
left ▾
Orientation based on the position

Text size
11
Font size of text in pt

Text attributes
☐ *Select all*
☐ Bold
☐ Italic
Attributes bold or italic for the inserted text

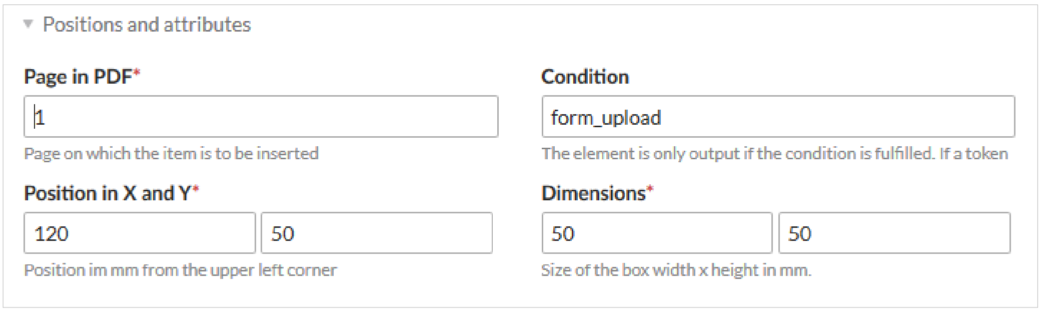
The **position** consists of the PDF page, the horizontal distance from the left edge and the vertical distance from the top edge. Optionally, also a right margin can be entered. This is often useful in form fields of type textarea as a position where the lines of text to wrap. Depending on the font used (defined as PDF_FONT_NAME_ MAIN /system/config/tcpdf.php), there may be a constant offset of over- and under-lengths in this font.

A good way to determine the positions, is a measurement of the PDF page e.g. in Photoshop. With a test position you can easily detect the offset values. Enter these offsets in the form properties and the offset will be included in all other positions.
Note: The positions always refer to the basic offset in the form properties.

b) Picture position

There are 3 possibilities for the integration of pictures into the PDF:

- Select an image from the file manager
- Transfer an image from the upload, i.e. from the SimpleTokens
- Include an image that consists of a data stream



▾ Positions and attributes

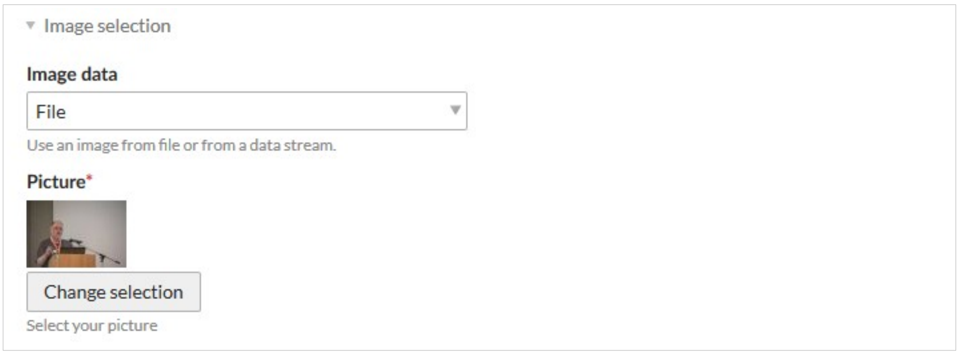
Page in PDF*
1
Page on which the item is to be inserted

Condition
form_upload
The element is only output if the condition is fulfilled. If a token

Position in X and Y*
120 | 50
Position im mm from the upper left corner

Dimensions*
50 | 50
Size of the box width x height in mm.

The page, position and size are specified for all picture positions. In addition, it is possible to specify a **condition** for the inclusion, the image is only included if the condition is fulfilled. For example, you can include images that match a select.



▾ Image selection

Image data

File ▾

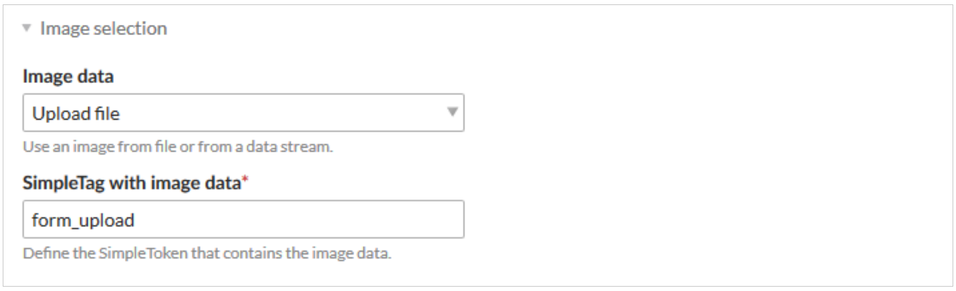
Use an image from file or from a data stream.

Picture*

Change selection

Select your picture

In the **File** option, select an image from the file manager.



▾ Image selection

Image data

Upload file ▾

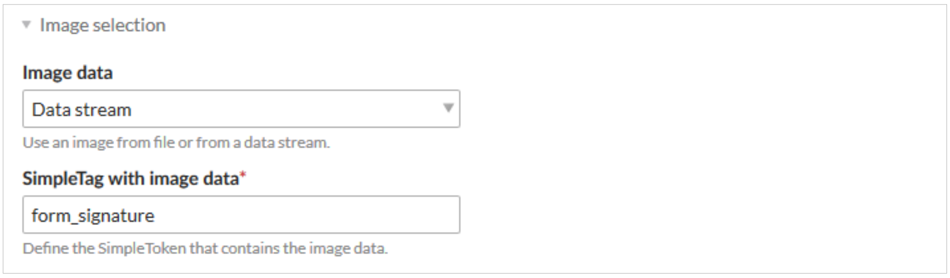
Use an image from file or from a data stream.

SimpleTag with image data*

form_upload

Define the SimpleToken that contains the image data.

In the option **Upload file** enter the SimpleToken which contains the file name of the uploaded file.



▾ Image selection

Image data

Data stream ▾

Use an image from file or from a data stream.

SimpleTag with image data*

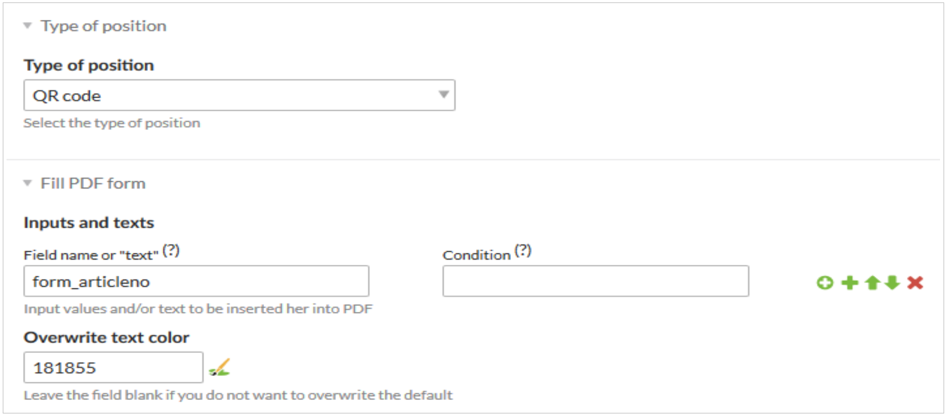
form_signature

Define the SimpleToken that contains the image data.

In the option **Data stream** you specify the SimpleToken which contains the image data stream. Example would be a signature from the extension *do-while/contao-signature-formfield-bundle*.

c) QR code

Include a QR code from your data in the PDF:



▾ Type of position

Type of position

QR code ▾

Select the type of position

▾ Fill PDF form

Inputs and texts

Field name or "text" (?)

form_articleno

Input values and/or text to be inserted her into PDF

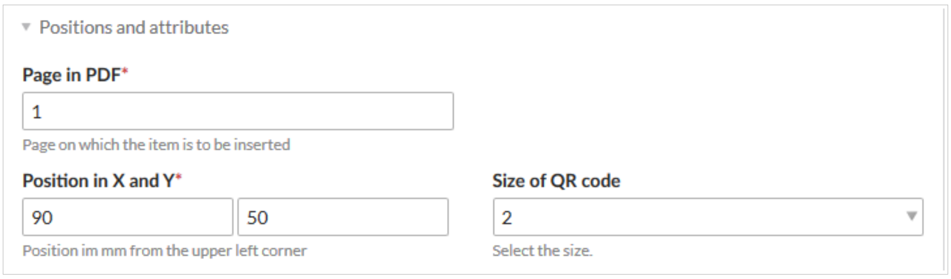
Condition (?)

Overwrite text color

181855

Leave the field blank if you do not want to overwrite the default

As with text positions, different texts, InsertTags and SimpleTokens can be linked here. The resulting text is encoded in the QR code. The QR code is normally output in the standard color, if you want to use a different color, enter it in the field **Overwrite text color**.



▾ Positions and attributes

Page in PDF*

1

Page on which the item is to be inserted

Position in X and Y*

90 | 50

Position im mm from the upper left corner

Size of QR code

2

Select the size.

Below you enter the position in the PDF and the size of the QR code.

5. PDF as email attachment

To send the generated PDF file together with the mail, the SimpleTokens of the Notification Center are extended in the notifications. In the token list you will now find a token **##pdfnc_document##** to use the PDF document name in the texts.

In the "Attachments via tokens" you will find the token **##pdfnc_attachment##** to attach the PDF document.

To prevent the e-mail from being sent, you can create a SimpleToken **##do_not_send_notification##** or **##form_do_not_send_notification###** (for forms), e.g. using a hidden form field or a form checkbox.

6. Directory protection, protected PDF

To prevent direct downloads of PDF files, which often contain personal data, the storage directory can be protected in the file manager of Contao. The files are still fully usable as an e-mail attachment.

In addition, the PDF created can be protected with a password. The protection can prevent the opening or restrict the permissions.

Note: The PDF protection is not completely secure. With appropriate programs, the password can be bypassed!

7. Insert tags

The extension replaces insert tags in the position definitions, so that, if necessary, dynamic data or the data of the registered member may be used.

The extension also provides its own insert tags available:

```
{{pdfnc::pdfdocument}}
```

can be used to insert a file link to the generated document in a download link.

`{{pdfnc::pdfdocument::name}}`

can be used to insert the file name of the PDF document produced in the page.

8. For programmers and developers

This section is intended for programmers who want to take in your own modules influence on these extensions. In the program sequence you can inject at different locations by means of a HOOK-registration. The extension **contao-pdf-nc-attachment-bundle** then calls the registered hooks, provided there are any.

The following hooks are available:

`$GLOBALS['TL_HOOKS']['pdfnc_BeforePdf']`

Called after the data preparation **before** the PDF is created. Here you can add further items to the passed array $arrPDF or modify existing ones, e.g. the filename of the PDF template.
The hook should return the $arrPDF as return value.

Example:

```
// config.php
$GLOBALS['TL_HOOKS']['pdfnc_BeforePdf'][] = array('vendor\MyExtension\MyClass',
                                                     'myPdfNcBeforePdf');

// MyClass.php
namespace vendor\MyExtension;

public function myPdfNcBeforePdf( $arrPDF, $pdfObj )
{
  // any code

  return $arrPDF;
}
```

`$GLOBALS['TL_HOOKS']['pdfnc_Positions']`

Called when creating a PDF for each position item on the form. It is possible to manipulate data. The page number can not be changed, otherwise the output will not work correctly. The return value is the modified array $arrItem.

Example:

```
// config.php
$GLOBALS['TL_HOOKS']['pdfnc_Positions'][] = array('vendor\MyExtension\MyClass',
                                                     'myPdfNcPositions');

// MyClass.php
namespace vendor\MyExtension;

public function myPdfNcPositions( $arrItem )
{
  // any code

  return $arrItem;
}
```

`$GLOBALS['TL_HOOKS']['pdfnc_AfterPdf']`

Called after the generation, storage and shipping. You can add more final activities by this Hook. The hook has no return value.

Example:

```
// config.php
$GLOBALS['TL_HOOKS']['pdfnc_AfterPdf'][] = array('vendor\MyExtension\MyClass',
                                             'myPdfNcAfterPdf');

// MyClass.php
namespace vendor\MyExtension;

public function myPdfNcAfterPdf( $pdfdatei, $arrPDF, $pdfObj )
{
  // any code
}
```

9. Troubleshooting

Collection of some points which could cause errors:

The template PDF file can not be read
The template PDF must be readable for the PDF generator, it must be available in the PDF specification format 1.4 (Acrobat 5.x). Recent PDF versions must be converted to the older format before.
In Adobe Acrobat on "Save As ..." select "Adobe PDF files optimized (*.pdf)" and then adjust the settings to "Acrobat 5.0 and higher."

The text does not have the right position in the form PDF generator
The measured positions usually have a fixed offset, which comes from the over- and under-lengths in the font. The easiest way is to set a text on test position and measure the difference to the desired position. This basic offset can be entered in the form properties and will calculate in the output at each position.

There are fields that do not exist in PDF form, but should be output in the document anyway.
Simply enter an additional page number for a note page, if the target page is not included in the template, a blank page is added where additional information can be placed in the same way.

10. Tips and hints

Specify the default PDF dynamically

1. In the gateway settings, the template PDF can be selected. The filename can be extended or directly specified, insert tags are also possible.

2. The filename (or the UUID) can be passed as SimpleToken: **##filename_template_pdf##** or **##form_filename_template_pdf##** (for forms).

3. The hook `$GLOBALS['TL_HOOKS']['pdfnc_BeforePdf']` allows you to change the filename. The $arrPDF pass parameter $arrPDF['vorlage'] contains the selected filename which can be changed and returned with the hook.

Update-safe use of your own font

When using your own fonts, the problem is that on the one hand the tcpdf.php where you enter the fonts is reset to the standard with every update and on the other hand stored own fonts are deleted from the TCPDF directories.

You can find an update proof solution here:

1. define an own tcpdf.php file
   The extension searches the tcpdf.php in the following order:
   - in the app/config/config.yml (from Contao 4.8 on in config/config.yml) can you have to add an additional line:
     `"pdftemplateTcpdf": "files/myfolder/tcpdf.php"`
     to tell Contao about the file
   - you can create your own tcpdf.php in the files directory
   - The default cpdf (up to Contao 4.6.x) will be searched
   - The default cpdf (from Contao 4.7.0 on) is searched
   - As a fallback, if nothing is found, the tcpdf.php in the extension

2. you can define the path for the fonts in your own update-secure tcpdf.php and store the fonts in the files directory.
   In order to use the font in the PDF, the file tcpdf.php must be adapted:
   `\define('PDF_FONT_NAME_MAIN', 'fontname');`

3. TCPDF does not use TTF or OTF fonts directly, the fonts must be converted first. This can be done online:
   http://fonts.snm-portal.com/
   Copy the received files (3 files per font) into the specified directory of the Contao installation.

Create PDF, but do not send the e-mail

There may be situations where a PDF should be created and saved, but no e-mail should be sent.
Create a SimpleToken **##do_not_send_notification##** or **##form_do_not_send_notification##** (for forms), e.g. using a hidden form field or a checkbox.

You can also add a token in the `$GLOBALS['TL_HOOKS']['pdfnc_BeforePdf']` hook.

___

Status: Version 2.7 - 2020-11-08
Softleister, Dipl. Ing. Hagen Klemp, info@softleister.de, www.softleister.de