

EXT: Logical Form

Extension Key: **logical_form**

Copyright 2007

Dirk Wildt, <dirk.wildt@think-visually.com>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.com

Table of Contents

| | | | |
|---------------------------------|-----------|--|-----------|
| EXT: Logical Form | 1 | Settings for an Input Field | 20 |
| Introduction | 1 | Settings for a Textarea | 21 |
| What does it do? | 1 | Settings for a Radio Button | 21 |
| Screenshots | 2 | Settings for a Checkbox | 21 |
| Users manual | 3 | Settings for a Selectbox | 21 |
| Quick start | 3 | Mail Settings for a Report Mail | 22 |
| Installation | 3 | Mail Settings for the Client Mail | 22 |
| Extension Manager | 3 | Language and translation | 23 |
| Tutorial | 3 | Files | 23 |
| Default mode | 3 | Helpful suggestions | 23 |
| Disclaimer with two steps | 5 | FAQ | 23 |
| Confirmation mode | 10 | SimulateStaticDocuments and RealUrl | 23 |
| Mail | 16 | Xhtml 1.0 Strict | 23 |
| Troubleshooting | 18 | Further Information | 24 |
| Reference | 19 | About the plugin icon | 24 |
| General Settings | 19 | Other extensions published by think visually | 24 |
| Workflow and Pids | 19 | To-Do list | 24 |
| General Field Settings | 20 | Changelog | 24 |

Introduction

What does it do?

This extension is helpful, if you have to evaluate the values of a form.

The form will be evaluated after sending and it will differ between a success and a failure case. The form handles the result.

There are two alternate workflows:

- Disclaiming: In case of unsuccessfulness there will be a failure page.
- Confirmation: In case of unsuccessfulness there will be an error message in the form.

The form will be configured in typoscript. So you don't have to know anything about php but you need a little experience in configuring typoscript. If you want, the form generates mails to providers and visitors.

Features

- A plugin with a form with an internal logic, which is evaluating values and has two possible workflows.

- In the default mode there are a select field for countries and an input field for a city. In the default mode you have to configure nothing but the pids of the success page and the failure page.
- If you like, the web master and the provider or the visitor of your website will get a mail with values and the result.
- It's possible to check mail addresses if they have a wrong syntax like a missing @.
- Full multi language support with TYPO3, configuration with typoscript.
- The form is in Xhtml.

Requirements

- Nothing.

Screenshots

Form and Workflow

Illustration 1: Logical Form in the default mode

```
18.10: 1
Linux 1:
18.10: 1
Hack.lu
20.10: 1
Ubuntu
Download
Software
Kostenl:
für all:
Aufgabe:
hier ..
Aussch
Hochsch
Die Hool
Rektore:
(HRK) u:
Wochen:
Zeit" s:
von der
Typo3-
Majord:
```

Illustration 2: Obligate fields will be evaluated

```
18.10: 1
Linux 1:
18.10: 1
Hack.lu
20.10: 1
Ubuntu
Download
Software
Kostenl:
für all:
Aufgabe:
hier ..
Aussch
Hochsch
Die Hool
Rektore:
(HRK) u:
Wochen:
Zeit" s:
von der
Typo3-
Majord:
```

Illustration 3: Case 1 - visitor is member of EU country

```
18.10: 1
Linux 1:
18.10: 1
Hack.lu
20.10: 1
Ubuntu
Download
Software
Kostenl:
für all:
Aufgabe:
hier ..
Aussch
Hochsch
Die Hool
Rektore:
(HRK) u:
Wochen:
Zeit" s:
von der
Typo3-
Majord:
```

Illustration 4: Case 1 - content for EU members

```
Messen
09.10: 1
Asteris!
13.10: 1
Electro:
18.10: 1
Linux 1:
18.10: 1
Hack.lu
20.10: 1
Ubuntu
```

Illustration 5: Case 2 - visitor isn't a member of EU

```
18.10: 1
Linux 1:
18.10: 1
Hack.lu
20.10: 1
Ubuntu
Download
Software
Kostenl:
für all:
Aufgabe:
hier ..
Aussch
Hochsch
Die Hool
Rektore:
(HRK) u:
Wochen:
Zeit" s:
von der
Typo3-
Majord:
```

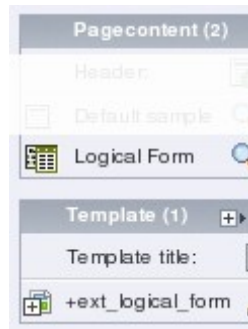
Illustration 6: Case 2 - content for non EU members

```
Messen
09.10: 1
Asteris!
13.10: 1
Electro:
18.10: 1
Linux 1:
18.10: 1
Hack.lu
20.10: 1
Ubuntu
```

Plugin



Illustration 7: Logical Form in the typo3 backend



Users manual

Quick start

- Install the extension with the extension manager.
- Create the pages
 - Disclaiming
 - Success
 - Failure
- Insert the plugin in the page content.
- Include static extension template
- Configure the extension logical_form in the template setup.

Installation

Extension Manager

Open the extension manager download the Logical Form extension (key logical_form) and install it:



Illustration 8: Extension Logical Form

Tutorial

Default mode

You can use the logical form immediately. In the default mode the logical form has

- a country field and
- a city field.

The form evaluates

- if obligated fields are filled up and
- if the visitor is from the European Union or not.

Create pages with content

Add three pages with the following structure to your typo3 backend:

- Default sample
 - Success
 - Failure

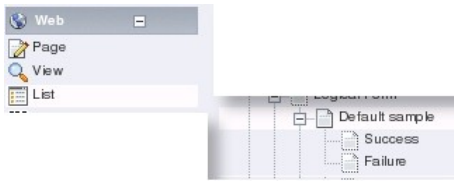


Illustration 9: Default example

In the page “Success” you add a text element with this content e.g.:

You are European Union!

The evaluation of the data was successful.

You can publish here everything for visitors who are from a country in the European Union like

- Germany or
- Spain

In the page “Failure” you add a text element with this content e.g.:

Not a member of EU

You can publish here everything for visitors who are not from the European Union because they are from

- Switzerland or
- USA

Add the plugin “logical form” to the page “Default sample”

Configure the extension template

Next, please add an extension template with the name “+ext_logical_form”.

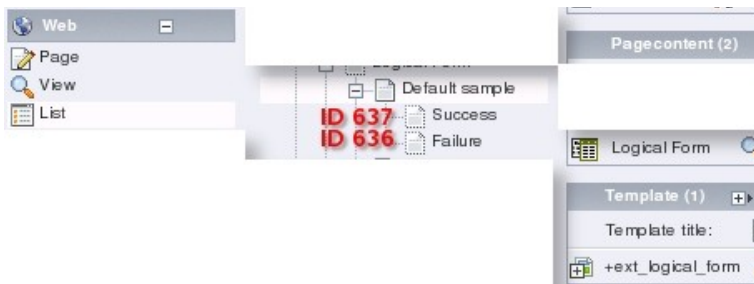


Illustration 10: Default sample with the plugin and the template

Configure the template:

1. Insert in the field “setup” the following lines (the pids are the pids of your pages “Success” and “Failure”):

```
plugin.tx_logicalform_pi1.pids.success = 637
plugin.tx_logicalform_pi1.pids.failure = 636
```

2. Insert in the field “include static template” the template “static (logical_form)” like in the illustration below.

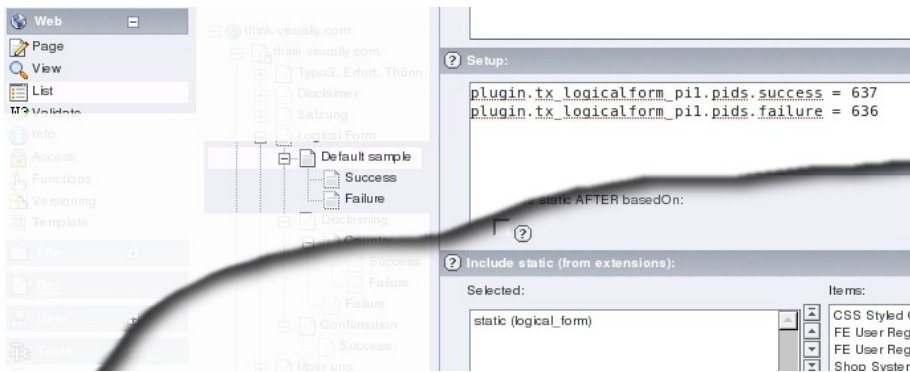


Illustration 11: Configuration of the extension template

The result in the frontend

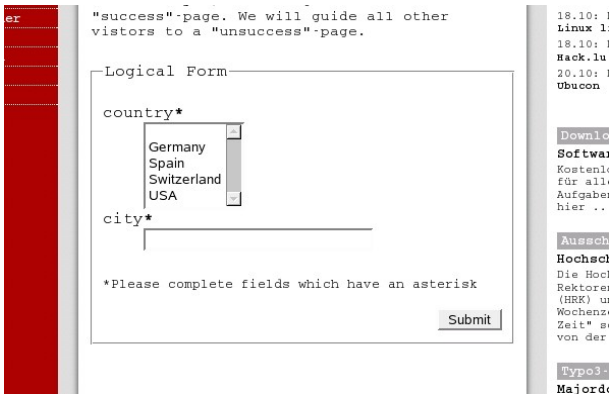


Illustration 12: The logical form in the frontend

The result in the frontend have to be the logical form like in the illustration.

Please test your logical form (look the examples above in the chapter “screenshots” on page 2):

1. Send the form without any value: There will be two error messages.
2. Send it with the values “Germany” and a city of your choice: There will be the message, that you are a member of the European Union.
3. Send it with the values “Switzerland” and a city of your choice: There will be the message, that you aren't a member of the European Union.

If you want to know, how to configure the logical form plugin

- give the two further examples a trial in the tutorial below or
- take a look in the reference on page 19 or
- check the setup of the tx_logicalform_pi1 with the typoscript object browser.

Disclaimer with two steps

In this exercise we use the logical form in a cascading way:

1. We ask, if the user understand our disclaimer.
2. If yes we will guide our visitor to the next page, where we will ask him, if he is a member of the European Union. We will add some country fields.

We need following pages and structure:

- Disclaiming
 - Country
 - Success
 - Failure
 - Failure



Illustration 13: Page tree in the backend

Feel free to add the following content:

- Disclaiming:
Disclaimer
This is the place where you can explain what to disclaim. In the first step you can ask your visitor if he understands your explanation.
- Country:
Are you a member of EU?
We want to know, if a visitor is coming from a country which is a member from the european union. If yes, we will guide him to a "success"-page. We will guide all other vistor to a "unsucces"-page.
 - Success
You are European Union!
The evaluation of the data was successful. You can publish here everything for visitors who are from a country in the European Union like
 - Germany or
 - Spain
 - Failure:
Not a member of EU
You can publish here everything for visitors who are not from the European Union because they are from
 - Switzerland or
 - USA
- Failure
Sorry, you are to stupid
We can't help you anymore.

Now we build our first own logical form.

Add the plugin logical form

Add the plugin logical form in the pages

- Disclaiming and
- Country

Configure the extension template of the page Disclaiming

Next, please add an extension template with the name "+ext logical_form" like in the example of "configure the extension template" on page 4.

1. Insert in the field "setup" the following lines (the pids are the pids of your pages "Country" (case success) and "Failure" (like the illustration on page 5):

```
#
# WORKFLOW
# =====
#
plugin.tx_logicalform_pil.workflow = failure_page
plugin.tx_logicalform_pil.pids.failure = 493
plugin.tx_logicalform_pil.pids.success = 494
#
# FIELDS, WE DON'T NEED
# =====
#
plugin.tx_logicalform_pil.fields.country >
plugin.tx_logicalform_pil.fields.city >
#
# FIELDS
# =====
#
plugin.tx_logicalform_pil.fields.confirm.radio.br = 0
plugin.tx_logicalform_pil.fields.confirm.radio.required = 1
plugin.tx_logicalform_pil.fields.confirm.radio.value.0 = no
plugin.tx_logicalform_pil.fields.confirm.radio.value.1 = yes
plugin.tx_logicalform_pil.fields.confirm.radio.failure.0 = no
#
# LANGUAGE EN (DEFAULT)
# =====
#
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_confirm = I have understand the disclaimer
plugin.tx_logicalform_pil._LOCAL_LANG.en.legend = Confirmation
plugin.tx_logicalform_pil._LOCAL_LANG.en.value_yes = Yes
plugin.tx_logicalform_pil._LOCAL_LANG.en.value_no = No
plugin.tx_logicalform_pil._LOCAL_LANG.en.submit_button_label = Send
```

```

#
# LANGUAGE DE
# =====
#
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_confirm = Ich habe den Disclaimer verstanden
plugin.tx_logicalform_pil._LOCAL_LANG.de.legend = Bestätigung
plugin.tx_logicalform_pil._LOCAL_LANG.de.value_yes = Ja
plugin.tx_logicalform_pil._LOCAL_LANG.de.value_no = Nein
plugin.tx_logicalform_pil._LOCAL_LANG.de.submit_button_label = Weiter

```

The whole setup is written in the file pi1/samples/setup_disclaim_1st.txt. You can read the file with the extension manager:

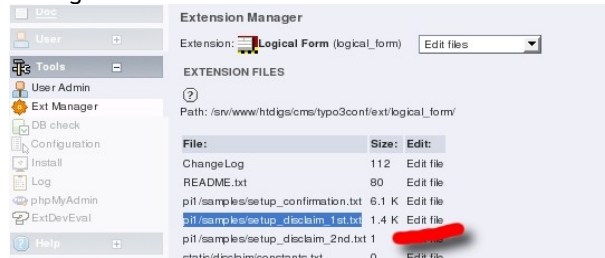


Illustration 14: The setup code

Or copy the code above and paste it into your setup.

2. Insert in the field “include static template” the template “static (logical_form)” like in the illustration in the example of “configure the extension template” on page 4.

The Logical Form should look like this in the frontend now:

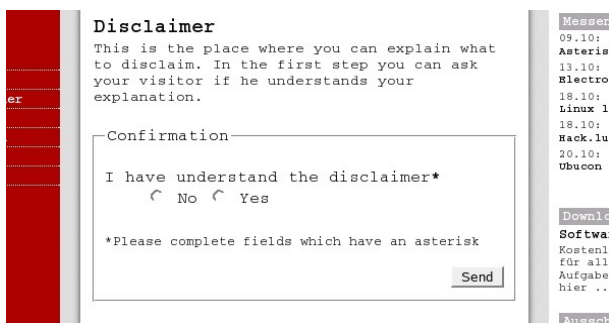


Illustration 15: Your first logical form

If you choose “No” the logical form will guide you to the failure page.

If you choose “Yes” the logical form will guide you to the success page.

Now we have to configure the success page. But before we explain the used setup code.

Explanation of the setup code

The “workflow” has two possibilities:

1. failure_page
2. form

The first (“failure_page”) means, that if the values aren't the values we want (the visitor doesn't understand the disclaimer), then the visitor will be guided to a page with a message.

The second (“form”) means, that the visitor will get an error message on the same page – the page with the logical form. Example for this message: Sorry, there is a mistake in your mail address.

```

#
# WORKFLOW
# =====
#
plugin.tx_logicalform_pil.workflow = failure_page

```

The pids are the pids of your pages for success (in this example the page “Country”) and failure (like the illustration on page 5).

```

plugin.tx_logicalform_pil.pids.failure = 493
plugin.tx_logicalform_pil.pids.success = 494

```

In the default mode the logical form has two fields – country and city. We don't need it, we delete it.

```
#
# FIELDS, WE DON'T NEED
# =====
#
plugin.tx_logicalform_pil.fields.country >
plugin.tx_logicalform_pil.fields.city >
```

But we need a radio button with the two values “no” and “yes”.

“fields.confirm” is the name of the field.

“fields.confirm.radio” is the Html type of the field in the form.

“fields.confirm.radio.br” has the following effect:

- 0: There will be no line break between the values
- 1: There will be a line break between the values

“fields.confirm.radio.required” has the following effect:

- 0: The visitor doesn't have to fill up the field.
- 1: The visitor has to fill up the field. There will be added an asterisk to the field name in the form automatically. There will be added the comment “Please complete fields which have an asterisk” at the bottom of the form automatically. You can edit every language expression. See Chapter “How to configure language for fields and values“ on page 15.

“fields.confirm.radio.value” is an array with the values of the radio button.

“fields.confirm.radio.failure” is an array with the values, we don't want to accept for the success case.

```
#
# FIELDS
# =====
#
plugin.tx_logicalform_pil.fields.confirm.radio.br = 0
plugin.tx_logicalform_pil.fields.confirm.radio.required = 1
plugin.tx_logicalform_pil.fields.confirm.radio.value.0 = no
plugin.tx_logicalform_pil.fields.confirm.radio.value.1 = yes
plugin.tx_logicalform_pil.fields.confirm.radio.failure.0 = no
```

With the `_LOCAL_LANG` array you can configure the display of your field names and values.

If you want to determine the displayed name of the field, you have to set it in this format:

- `language.field_name = value`

“language” has the value of the language like “en” for English or “de” for german.

“field_name” has the value “field_” and than the name of the field. In the example above we defined the displayed name of the field “confirm”:

```
_LOCAL_LANG.en.field_confirm = I have understand the disclaimer.
```

If you want to determine the displayed values of a field, you have to set it in this format:

- `language.value_name = value`

“language” has the value of the language like “en” for English or “de” for german.

“value_name” has the value “value_” and than the name of the value. In the example above we defined the displayed name of the value “yes”:

```
_LOCAL_LANG.en.value_yes = Yes
```

Here you have the value for the submitbutton:

```
_LOCAL_LANG.en.submit_button_label = Yes
```

```
#
# LANGUAGE EN (DEFAULT)
# =====
#
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_confirm = I have understand the disclaimer
plugin.tx_logicalform_pil._LOCAL_LANG.en.legend = Confirmation
plugin.tx_logicalform_pil._LOCAL_LANG.en.value_yes = Yes
plugin.tx_logicalform_pil._LOCAL_LANG.en.value_no = No
plugin.tx_logicalform_pil._LOCAL_LANG.en.submit_button_label = Send
```

Below there is the same example for german language:

```
#
# LANGUAGE DE
# =====
#
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_confirm = Ich habe den Disclaimer verstanden
plugin.tx_logicalform_pil._LOCAL_LANG.de.legend = Bestätigung
plugin.tx_logicalform_pil._LOCAL_LANG.de.value_yes = Ja
plugin.tx_logicalform_pil._LOCAL_LANG.de.value_no = Nein
plugin.tx_logicalform_pil._LOCAL_LANG.de.submit_button_label = Weiter
```

Configuring the success page

The success page “Country” is showing the same logical form like the page “Disclaiming”.

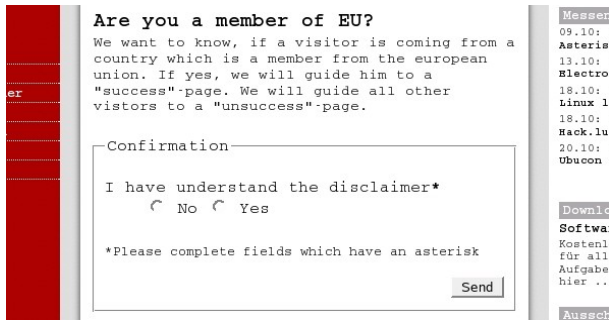


Illustration 16: Page "Country" without configuration

This isn't a surprise really, because we didn't configure the extension template of the page “Country”.

Our page “Country” should look like this:



Illustration 17: Page "Country" with configuration

We have to complete the following orders:

1. We want to display the default logical form with the fields “country” and “city”.
2. We have to configure the pids of the success and failure page.
3. We have to delete the field confirm.

Configure the extension template

Next, please add an extension template with the name “+ext_logical_form”.

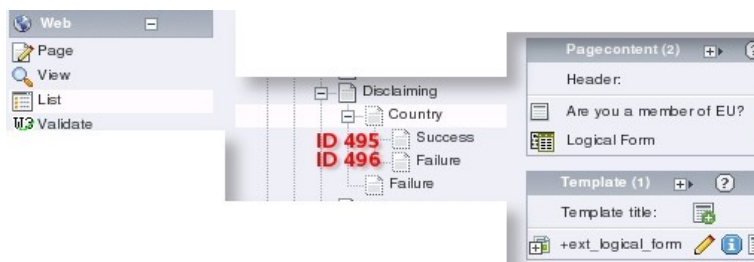


Illustration 18: Default sample with the plugin and the template

Configure the template:

1. Insert in the field “setup” the following lines (the pids are the pids of your pages “Success” and “Failure”):

```

#
# WORKFLOW
# =====
#
plugin.tx_logicalform_pil.workflow = failure_page
plugin.tx_logicalform_pil.pids.failure = 496
plugin.tx_logicalform_pil.pids.success = 495
#
# FIELDS, WE DON'T WANT AND WE DELETE
# =====
#
plugin.tx_logicalform_pil.fields.confirm >

```

2. Insert in the field “include static template” the template “static (logical_form)” like in the illustration below.

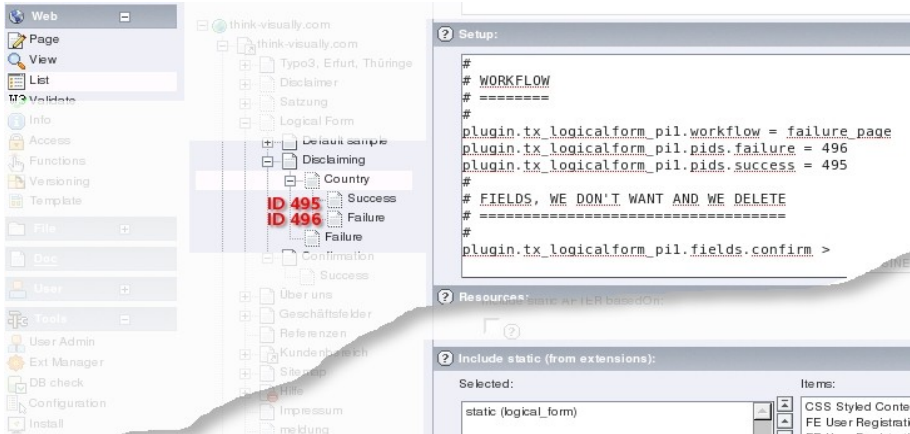


Illustration 19: Configuration of the extension template

The result in the frontend should look like this:



Illustration 20: Page "Country" with configuration

Confirmation mode

This exercise demonstrate, how to create a form for ordering something:

- We need only a success page (for confirmation) and we don't need a failure page.
- We have to delete the default fields.
- We have to create some fields.
- We have to configure the mails to the provider and the visitor.

Create pages with the content and the plugin

Add two pages in the following structure to your typo3 backend:

- Confirmation
 - Success

Please add to the page “Confirmation”

- this content:
 - Confirmation
 - This is an example for the confirmation case. E.g. a visitor likes to get a document by postal delivery. He can order the document with the form above. He will get a confirmation if he filled up every field and if there is no mistake in the mail.
- the plugin “Logical Form”
- and an extension template “+ext_logical_form”



Illustration 21: Confirmation example

Please add to the page “Success” this content:

- Thank you
- Thanks for ordering our document.
- If you have any question please call us.
- * Contact (Link to your contact page)

Configure the extension template of the page Confirmation

Next, please add an extension template with the name “+ext_logical_form” like in the example of “configure the extension template” on page 4.

1. Insert in the field “include static template” the template “static (logical_form)” like in the illustration in the example of “configure the extension template” on page 4.
2. Insert in the field “setup” the following lines (the pid is the pid of your page “Success”):

```
#
# WORKFLOW
# =====
#
plugin.tx_logicalform_pil.workflow = form
plugin.tx_logicalform_pil.pids.failure >
plugin.tx_logicalform_pil.pids.success = 634
#
# MAIL TO THE CLIENT, THE VISITOR OF OUR WEBSITE
# =====
#
plugin.tx_logicalform_pil.mail.to_client = 1
plugin.tx_logicalform_pil.mail.to_client.from.mail = name@my.provider.com
plugin.tx_logicalform_pil.mail.to_client.from.name = my provider company
plugin.tx_logicalform_pil.mail.to_client.bcc = dirk.wildt@think-visually.com
plugin.tx_logicalform_pil.mail.to_client.field = mail
plugin.tx_logicalform_pil.mail.to_client.mail.to_client.body = mail_client
plugin.tx_logicalform_pil.mail.to_client.signature.0 = my provider company
plugin.tx_logicalform_pil.mail.to_client.signature.1 = my provider contact
plugin.tx_logicalform_pil.mail.to_client.signature.2 = my provider street
plugin.tx_logicalform_pil.mail.to_client.signature.3 = my provider zip and city
plugin.tx_logicalform_pil.mail.to_client.signature.4 = my provider phone
plugin.tx_logicalform_pil.mail.to_client.signature.5 = my provider mail address
#
# FIELDS, WE DON'T NEED
# =====
#
plugin.tx_logicalform_pil.fields.country >
plugin.tx_logicalform_pil.fields.city >
#
# NEW FIELDS
# =====
#
plugin.tx_logicalform_pil.fields.mail.input.failure.0 = my.name@example.com
plugin.tx_logicalform_pil.fields.mail.input.failure.1 = dirk.wildt@think-visually.com
plugin.tx_logicalform_pil.fields.mail.input.default = my.name@example.com
plugin.tx_logicalform_pil.fields.mail.input.size = 30
plugin.tx_logicalform_pil.fields.mail.input.maxlength = 60
plugin.tx_logicalform_pil.fields.mail.input.required = 1
```

```

plugin.tx_logicalform_pil.fields.mail.input.mail = 1
plugin.tx_logicalform_pil.fields.sexus.radio.br = 0
plugin.tx_logicalform_pil.fields.sexus.radio.required = 1
plugin.tx_logicalform_pil.fields.sexus.radio.value.0 = male
plugin.tx_logicalform_pil.fields.sexus.radio.value.1 = female
plugin.tx_logicalform_pil.fields.name.input.size = 30
plugin.tx_logicalform_pil.fields.name.input.maxlength = 60
plugin.tx_logicalform_pil.fields.name.input.required = 1
plugin.tx_logicalform_pil.fields.street.input.size = 30
plugin.tx_logicalform_pil.fields.street.input.maxlength = 60
plugin.tx_logicalform_pil.fields.street.input.required = 1
plugin.tx_logicalform_pil.fields.zip_city.input.size = 30
plugin.tx_logicalform_pil.fields.zip_city.input.maxlength = 60
plugin.tx_logicalform_pil.fields.zip_city.input.required = 1
plugin.tx_logicalform_pil.fields.phone.input.size = 30
plugin.tx_logicalform_pil.fields.phone.input.maxlength = 60
plugin.tx_logicalform_pil.fields.phone.input.required = 1
plugin.tx_logicalform_pil.fields.note.textarea.cols = 30
plugin.tx_logicalform_pil.fields.note.textarea.rows = 5
plugin.tx_logicalform_pil.fields.note.textarea.required = 0
#
# LANGUAGE EN (DEFAULT)
# =====
#
plugin.tx_logicalform_pil._LOCAL_LANG.en.legend = I want this document
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_mail = Mail
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_name = First and last Name
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_note = Message to the provider
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_phone = Phone
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_sexus = Gender
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_street = Street
plugin.tx_logicalform_pil._LOCAL_LANG.en.field_zip_city = ZIP and City
plugin.tx_logicalform_pil._LOCAL_LANG.en.mail_client_subject = Delivery of the ordered document
plugin.tx_logicalform_pil._LOCAL_LANG.en.mail_client_text (
Dear ###VALUE_NAME###,
Thanks for ordering our document.
We will send it with postal delivery to your address:

    ###VALUE_NAME###
    ###VALUE_STREET###
    ###VALUE_ZIP_CITY###

You have send the following data:

###VALUES###

If there is any mistake please send the correct data.
Please visit this page and fill up the form:

###URL###

This is a mechanical message.

Best Regards

###SIGNATURE###
)
plugin.tx_logicalform_pil._LOCAL_LANG.en.value_female = Female
plugin.tx_logicalform_pil._LOCAL_LANG.en.value_male = Male
#
# LANGUAGE DE
# =====
#
plugin.tx_logicalform_pil._LOCAL_LANG.de.legend = Ich möchte das Dokument
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_mail = eMail
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_name = Vor- und Nachname
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_note = Nachricht an den Betreiber
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_phone = Telefon
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_sexus = Anrede
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_street = Straße
plugin.tx_logicalform_pil._LOCAL_LANG.de.field_zip_city = PLZ und Ort
plugin.tx_logicalform_pil._LOCAL_LANG.de.mail_client_subject = Versand des bestellten Dokuments
plugin.tx_logicalform_pil._LOCAL_LANG.de.mail_client_text (
Hallo ###VALUE_NAME###,
herzlichen Dank für die Bestellung unseres Dokuments.
Wir senden Ihnen das Dokument per Post an Ihre Adresse:

    ###VALUE_NAME###

```

```
###VALUE_STREET###
###VALUE_ZIP_CITY###
```

Sie haben uns folgende Daten gesendet:

```
###VALUES###
```

Sollte in den Daten ein Fehler stecken, senden Sie uns Ihre richtigen Daten bitte ein zweites Mal. Besuchen Sie dafür bitte folgende Seite:

```
###URL###
```

Dies ist eine automatische generierte Mitteilung.

Mit freundlichen Grüßen

```
###SIGNATURE###
```

```
)
plugin.tx_logicalform_pil._LOCAL_LANG.de.value_female = Frau
plugin.tx_logicalform_pil._LOCAL_LANG.de.value_male = Herr
```

The whole setup is written in the file pi1/samples/setup_disclaim_2nd.txt. You can read the file with the extension manager:

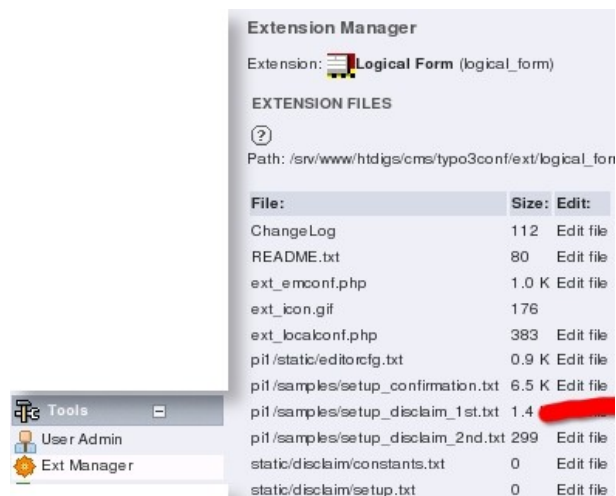


Illustration 22: The setup code

Or copy the code above and paste it into your setup.

Explanation of the setup code

We don't repeat the explanation of the example above "Explanation of the setup code" on page 7. If you didn't read it, please read and understand it now.

The workflow value is "form", because there is only the failure case of missing data or wrong mail addresses. In this case every error should be displayed with a message in the form and not in a failure page:

- We set workflow to: form
- We delete the value pids.failure

```
#
# WORKFLOW
# =====
#
plugin.tx_logicalform_pil.workflow = form
plugin.tx_logicalform_pil.pids.failure >
plugin.tx_logicalform_pil.pids.success = 634
```

The array mail.to_client is explained in the chapter "mail" on page 16.

How to configure fields

In our form we need the following fields:

- mail (input)
- sexus (radio)
- name (input)
- street (input)
- zip_city (input)
- note (input)

It is easier to understand the controlling of the Logical Form, if we don't use the language configuration.

Therefore we commented out the lines with the language configuration as shown below:

Illustration 23: The form without language configuration

```
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.legend = I want this document
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_mail = Mail
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_name = First and last Name
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_note = Message to the provider
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_phone = Phone
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_sexus = Gender
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_street = Street
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_zip_city = ZIP and City
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.value_female = Female
#plugin.tx_logicalform_pi1._LOCAL_LANG.en.value_male = Male
```

The following images show, how to control fields.

Field name, field type and field values

Illustration 24:
 plugin.tx_logicalform_pi1.fields.mail...
 plugin.tx_logicalform_pi1.fields.sexus...

Illustration 25:
 plugin.tx_logicalform_pi1.fields.mail.input...
 plugin.tx_logicalform_pi1.fields.sexus.radio...

Illustration 26:
 plugin.tx_logicalform_pi1.fields.sexus.radio.value.0 = male
 plugin.tx_logicalform_pi1.fields.sexus.radio.value.1 = female

Required fields

if there is no mistake in the mail.

Order

fields.mail.input.required = 1

mail*

sexus* male female

fields.sexus.radio.required = 1

name*

street*

18.10: Hack.lu
20.10: Ubucon

Software:
Kostenl:
für all:
Aufgabe:
hier ..

Auswahl:
Hochsch:
Die Hoch:
Rektore:

Illustration 27:
 plugin.tx_logicalform_pi1.fields.mail.input.required = 1
 plugin.tx_logicalform_pi1.fields.sexus.radio.required = 1

Default values and evaluating of a mail address

if there is no mistake in the mail.

Order

fields.mail.input.default = my.name...

mail*

sexus* male female

name*

street*

18.10: Hack.lu
20.10: Ubucon

Software:
Kostenl:
für all:
Aufgabe:
hier ..

Auswahl:
Hochsch:
Die Hoch:
Rektore:

Illustration 28:
 plugin.tx_logicalform_pi1.fields.mail.input.default = my.name...

if there is no mistake in the mail.

Order

fields.mail.input.mail = 1

mail*

sex*

Check, if value is a valid mail address

name*

street*

18.10: Hack.lu
20.10: Ubucon

Software:
Kostenl:
für all:
Aufgabe:
hier ..

Auswahl:
Hochsch:
Die Hoch:
Rektore:

Illustration 29:
 plugin.tx_logicalform_pi1.fields.mail.input.mail = 1

How to configure language for fields and values

In our example we want that

- the field “mail” will be displayed as “Mail”,
- “sexus” as “Gender” and
- the values “male” as “Male” and “female” as “Female”

```
#
# LANGUAGE EN (DEFAULT)
# =====
#
...
plugin.tx_logicalform_pi1._LOCAL_LANG.en.field_mail = Mail
...
```

if there is no mistake in the mail.

Order

Logical Form

mail*

sexus* male female

name*

street*

18.10: Hack.lu
20.10: Ubucon

Software:
Kostenl:
für all:
Aufgabe:
hier ..

Auswahl:
Hochsch:
Die Hoch:
Rektore:

Illustration 30: Our form without any translation

if there is no mistake in the mail.

Order

_LOCAL_LANG.en.field_mail = Mail

Mail*

Gender* Male Female

First and last Name*

Street*

18.10: Hack.lu
20.10: Ubucon

Software:
Kostenl:
für all:
Aufgabe:
hier ..

Auswahl:
Hochsch:
Die Hoch:
Rektore:

Illustration 31:
 ..._LOCAL_LANG.en.field_mail = Mail
 ..._LOCAL_LANG.en.field_sexus = Gender

if there is no mistake in the mail.

Order

I want this document

Mail*

Gender* Male Female

First and last Name*

Street*

18.10: Hack.lu
20.10: Ubucon

Software:
Kostenl:
für all:
Aufgabe:
hier ..

Auswahl:
Hochsch:
Die Hoch:
Rektore:

Illustration 32:
 ..._LOCAL_LANG.en.value_female = Female'
 ..._LOCAL_LANG.en.value_male = Male

Mail

Configuring the mail to the client – the visitor of our website

You have to configure the array “mail.to_client...”.

You have to set the flag “mail.to_client” to 1, otherwise the server won't send any mail to a client.

```
#
# MAIL TO THE CLIENT, THE VISITOR OF OUR WEBSITE
# =====
#
plugin.tx_logicalform_pil.mail.to_client = 1
plugin.tx_logicalform_pil.mail.to_client.from.mail = name@my.provider.com
plugin.tx_logicalform_pil.mail.to_client.from.name = my provider company
plugin.tx_logicalform_pil.mail.to_client.bcc = dirk.wildt@think-visually.com
plugin.tx_logicalform_pil.mail.to_client.field = mail
plugin.tx_logicalform_pil.mail.to_client.body = mail_client
plugin.tx_logicalform_pil.mail.to_client.signature.0 = my provider company
plugin.tx_logicalform_pil.mail.to_client.signature.1 = my provider contact
plugin.tx_logicalform_pil.mail.to_client.signature.2 = my provider street
plugin.tx_logicalform_pil.mail.to_client.signature.3 = my provider zip and city
plugin.tx_logicalform_pil.mail.to_client.signature.4 = my provider phone
plugin.tx_logicalform_pil.mail.to_client.signature.5 = my provider mail address
```

We hope, that the illustration below shows the relation between the mail and the controlling with the setup understandably.

With the value

```
plugin.tx_logicalform_pil.mail.to_client.field = mail
```

you determine, in which field of the form we found the mail address of the client.

We explain in step II, how to configure the mail subject and the mail message.

Illustration 33: The field name of the mail address is “mail”. In the setup it's the line: `plugin.tx_logicalform_pi1.fields.mail...`

Illustration 34: The sent mail and the controlling with the setup

1. mail.to_client.from.mail = name@my.provider.com
2. mail.to_client.from.name = my provider company
3. mail.to_client.field = mail
4. mail.to_client.signature.0 = my provider company
mail.to_client.signature.1 = ...

Configuring the subject and the message of the mail to the client

You have to define

- the subject and
- the message text

in the language array “_LOCAL_LANG”. You are free, to define the string, which is marked yellow. But it has to be the same string like in the setup variable “mail_client.body”.

```
plugin.tx_logicalform_pil.mail.to_client.body = mail_client
...
plugin.tx_logicalform_pil._LOCAL_LANG.en.mail_client_subject = ...
plugin.tx_logicalform_pil._LOCAL_LANG.en.mail_client_text = ...
```

In the message – the mail body – you may work with markers.

If you want use values, which the client have written in the form, there is this main principle:

- `###VALUE_fieldname###`

For example:

In the form is a field called “street”, than the marker in the mail is `###VALUE_STREET###`.

If you want a list of all form values, than you use the marker `###VALUES###`.

There a two markers additionally:

- `###URL###`: The Url of the form
- `###SIGNATURE###`: The setup array “.mail.to_client.signature...”

Our Example

```

plugin.tx_logicalform_pil._LOCAL_LANG.en.mail_client_subject = Delivery of the ordered document
plugin.tx_logicalform_pil._LOCAL_LANG.en.mail_client_text (
Dear ###VALUE_NAME###,
Thanks for ordering our document.
We will send it with postal delivery to your address:

```

```

###VALUE_NAME###
###VALUE_STREET###
###VALUE_ZIP_CITY###

```

You have send the following data:

```
###VALUES###
```

If there is any mistake please send the correct data.
Please visit this page and fill up the form:

```
###URL###
```

This is a mechanical message.

Best Regards

```
###SIGNATURE###
)
```

And the result:

if there is no mistake in the mail.

Order
I want this document

Mail*
dirk.wildt@puppenspiel-portal.de

Gender*
 Male Female

First and last Name*
Dirk Wildt

Street*
Andreasstraße 37

ZIP and City*
99084 Erfurt

Phone*
0361 / 6025631

Message to the provider

*Please complete fields which have an asterisk

Submit

Illustration 35: Fields and values in the form

Betreff: Delivery of the ordered document
Von: my provider company <name@my.provider.com>
Datum: 12:39
An: dirk.wildt@puppenspiel-portal.de

Dear Dirk Wildt, **###VALUE_NAME###**
Thanks for ordering our document.
We will send it with postal delivery to your address:

Dirk Wildt **###VALUE_NAME###**
Andreasstraße 37 **###VALUE_STREET###**
99084 Erfurt **###VALUE_ZIP_CITY###**

You have send the followong data:

Mail: dirk.wildt@puppenspiel-portal.de
Gender: Male
First and last Name: Dirk Wildt
Street: Andreasstraße **###VALUES###**
ZIP and City: 99084 Erfurt
Phone: 0361 / 6025631
Message to the provider: No message text

If there is any msitake please send the correct data twice.
Please visit this page and fill up the form:
<http://www.think-visually.com/###URL###/>

This is a mechanical message.

Best Regards

my provider company
my provider contact
my provider phone
my provider zip and city
my provider phone
my provider mail address
###SIGNATURE###

Illustration 36: Mail with values of the form and of the setup

Configuring a report mail

With Logical Form you can configure a report mail.

This mail send all datas of the form to a mail address of your choice.

Example for a configuration:

```
#
# MAIL REPORT I.E. TO THE ADMIN OR/AND THE PROVIDER OF THE WEBSITE
# =====
#
plugin.tx_logicalform_pil.mail.report = 1
plugin.tx_logicalform_pil.mail.report.from.mail = webmaster@my-domain.com
plugin.tx_logicalform_pil.mail.report.from.name = Typo3 Logical Form Report
plugin.tx_logicalform_pil.mail.report.to.0 = henry@statistic.com
plugin.tx_logicalform_pil.mail.report.to.1 = mary@statistic.com
plugin.tx_logicalform_pil.mail.report.signature.0 = My company
plugin.tx_logicalform_pil.mail.report.signature.1 = My name
plugin.tx_logicalform_pil.mail.report.signature.2 = My street
plugin.tx_logicalform_pil.mail.report.signature.3 = My zipcode and my city
plugin.tx_logicalform_pil.mail.report.signature.4 = My phone
plugin.tx_logicalform_pil.mail.report.signature.5 = my.name@my-domain.com
```

If you want to edit the mail content, please edit

- pi1/locallang.xml (see chapter “Language” on page 23)

Troubleshooting

If the logical_form extension doesn't display anything (not even an error message) check the following:

- Did you set a “static template (from Extensions)” in your TS-template?
- Clear all TYPO3 caches, including the temp files in typo3conf, clear your Browser cache.

Debug Mode

The Logical Form has a debug mode. In this mode the extension will print messages in your Html code. The debug mode makes sense, if you have any problem with mailing.

Turn the debug mode on:

```
plugin.tx_logicalform_pil.mail.debug = 1
```

You will get the following output e.g.:

Try to send the following mail

```
Header: MIME-Version: 1.0
Content-type: text/plain; charset=iso-8859-15
From: my provider company <name@my.provider.com>
Bcc: dirk.wildt@think-visually.com
```

```
To: dirk.wildt@puppenspiel-portal.de
Subject: Delivery of the ordered document
Body: Dear Dirk Wildt,
Thanks for ordering our document.
```

Illustration 37: Example for Html output in the debug mode

Reference

plugin.tx_logicalform_pi1 properties: TS configuration.

General Settings

| Property: | Data type: | Description: | Default: |
|---------------|------------|--|--|
| templateFile | string | The template file for the form Example: <pre>plugin.tx_logicalform_pi1 { templateFile = typo3conf/ext/logical_form/ \ pi1/template.html }</pre> | typo3conf/ext/logical_form/pi1/template.html |
| code | string | The marker for the form subpart in the template file Example: <pre>plugin.tx_logicalform_pi1 { code = FORM }</pre> | FORM |
| error.wrap | string | The wrap for an error message in the form. The message will be displayed, if a field is required but the value of this field is an empty string. Example: <pre>plugin.tx_logicalform_pi1.error { wrap = <dd><div style="font-weight:bold; \ font-size:small;color:red;"> </div></dd> }</pre> | <div style="font-weight:bold;font-size:small;color:red;"> </div> |
| required.wrap | string | The wrap for the line required in the form. The required line will be displayed, if there is any field which is required. Example: <pre>plugin.tx_logicalform_pi1.required { wrap = <dd><div style="font-size:small; \ padding-top:2em;"> </div></dd> }</pre> | <div style="font-size:small;padding-top:2em;"> </div> |
| date | string | Output format for a timestamp. A timestamp is used in a report mail. <pre>plugin.tx_logicalform_pi1{ date = Y-m-d H:i:s }</pre> | Y-m-d H:i:s |

Workflow and Pids

| Property: | Data type: | Description: | Default: |
|------------------------------|-------------------------------------|--|--------------|
| workflow | string <i>failure_page, form</i> | The workflow of the logical form. failure_page: In case of unsuccessfulness the visitor will be guided to a failure page. You have to configure the setup value pids.failure! form: In case of unsuccessfulness there will be an error message in the form. Example: <pre>plugin.tx_logicalform_pi1 { workflow = failure_page }</pre> | failure_page |
| pids.failure pids.success | integer | The ids of the pages with the failure and success message. You have to configure the value pids.success in every case! You have to configure the value pids.failure, if your workflow (see above) is failure_page (this is the default mode). Example: <pre>plugin.tx_logicalform_pi1.pids { failure = 417 success = 418 }</pre> | |

General Field Settings

Settings for all field in the extension logical_form

| Property: | Data type: | Description: | Default: |
|--|---|--|-------------------------------|
| fields.[field_name]. [field_type] | array [field_type]: <i>input</i> <i>textarea</i> <i>radio</i> <i>checkbox</i> <i>select</i> | Create a field Example for a select field: <pre>plugin.tx_logicalform_pi1.fields.country.select { ... }</pre> Example for an input field: <pre>plugin.tx_logicalform_pi1.fields.city.input { ... }</pre> See tables below | country.select, city.input |
| fields.[field_name]. [field_type].required | boolean | If the field is required the extension won't accept an empty value The field name will be completed with an asterisk. The last line in the form will be an information, that required fields have to be completed. Example: <pre>plugin.tx_logicalform_pi1.fields.country.select { required = 1 }</pre> | |
| fields.[field_name]. [field_type].failure.x | string | Value, which shouldn't be accepted Example: <pre>plugin.tx_logicalform_pi1.fields.country.select { failure.0 = Switzerland failure.1 = USA }</pre> | |
| fields.[field_name]. [field_type].default | string | Default value of a field Example: <pre>plugin.tx_logicalform_pi1.fields.city.input { default = Berlin }</pre> | |

Settings for an Input Field

| Property: | Data type: | Description: | Default: |
|---|------------|--|----------|
| fields.[field_name]. input.size | integer | Html length of the input field Example: <pre>plugin.tx_logicalform_pi1.fields.city.input { size = 30 }</pre> | |
| fields.[field_name]. input.maxlength | integer | Allowed length of the value in the input field Example: <pre>plugin.tx_logicalform_pi1.fields.city.input { maxlength = 60 }</pre> | |
| fields.[field_name]. input.mail | boolean | If set the value of the input field have to be a valid mail address m(e.g. with an @). Otherwise there will be an error message Example: <pre>plugin.tx_logicalform_pi1.fields.email.input { mail = 1 }</pre> | |

Settings for a Textarea

| Property: | Data type: | Description: | Default: |
|---------------------------------------|------------|--|----------|
| fields.[field_name]. textarea.cols | integer | Html cols of the texarea Example: <code>plugin.tx_logicalform_pi1.fields.note.textarea { cols = 30 }</code> | |
| fields.[field_name]. textarea.rows | integer | Html rows of the texarea Example: <code>plugin.tx_logicalform_pi1.fields.note.textarea { rows = 5 }</code> | |

Settings for a Radio Button

| Property: | Data type: | Description: | Default: |
|---------------------------------------|------------|--|----------|
| fields.[field_name]. radio.value.x | string | Option field of a radio button Example: <code>plugin.tx_logicalform_pi1.fields.sexus.radio { value.0 = male value.1 = female }</code> | |
| fields.[field_name]. radio.br | boolean | If set, then there will be a -tag between the option fields Example: <code>plugin.tx_logicalform_pi1.fields.sexus { br = 1 }</code> | |

Settings for a Checkbox

| Property: | Data type: | Description: | Default: |
|--|------------|--|----------|
| fields.[field_name]. checkbox.value.x | string | Option field of a checkbox Example: <code>plugin.tx_logicalform_pi1.fields.profile.checkbox { value.0 = sport value.1 = culture value.2 = vacation }</code> | |

Settings for a Selectbox

| Property: | Data type: | Description: | Default: |
|---|------------|--|----------|
| fields.[field_name]. select.size | integer | Amount of the displayed items in the option list Example: <code>plugin.tx_logicalform_pi1.fields.country.select { size = 5 }</code> | |
| fields.[field_name]. select.value.x | string | Option value of a selectbox Example: <code>plugin.tx_logicalform_pi1.fields.country.select { value.0 = value.1 = Germany value.2 = Spain value.3 = Switzerland value.4 = USA }</code> | |
| fields.[field_name]. select.multiple | boolean | Enable the multiple choice of a selectbox Example: <code>plugin.tx_logicalform_pi1.fields.country.select { multiple = 1 }</code> | |

Mail Settings for a Report Mail

| Property: | Data type: | Description: | Default: |
|---|----------------|--|-------------------------|
| mail.from | string | The sender of the mail, which will send in cases of failure and success to the provider of the website and others Example: <pre>plugin.tx_logicalform_pi1.mail { from = webmaster@my-client.com }</pre> | webmaster@my-client.com |
| mail.debug | boolean | If set there is a output in Html Example: <pre>plugin.tx_logicalform_pi1.mail { debug = 1 }</pre> | 0 |
| mail.report | boolean, array | Activating a report mail (you have to configure the report array). Example: <pre>plugin.tx_logicalform_pi1.mail { report = 1 }</pre> | 0 |
| mail.report.from.mail, mail.report.from.name | strings | Sender of the report mail Example: <pre>plugin.tx_logicalform_pi1.mail.report { from.mail = webmaster@my-domain.com from.name = Typo3 Logical Form Report }</pre> | |
| mail.report.to.x | string | Receiver of the report mail Example: <pre>plugin.tx_logicalform_pi1.mail.report { to.0 = webmaster@my-domain.com to.1 = henry@statistics.com }</pre> | |
| mail.report.signature.x | string | Signature of the report mail Example: <pre>plugin.tx_logicalform_pi1.mail.report { signature.0 = My company signature.1 = My name signature.2 = webmaster@my-domain.com }</pre> | |

Mail Settings for the Client Mail

| Property: | Data type: | Description: | Default: |
|---|----------------|--|----------|
| mail.client | boolean, array | Activating a mail to the client, the visitor, who filled up the values in the form (you have to configure the mail client array). Example: <pre>plugin.tx_logicalform_pi1.mail { client = 1 }</pre> | 0 |
| mail.client.from.mail, mail.client.from.name | strings | Sender of the report mail Example: <pre>plugin.tx_logicalform_pi1.mail.client { from.mail = name@my.provider.com from.name = My provider company }</pre> | |
| mail.client.bcc | string | Comma separated list of mail addresses for a blind copy Example: <pre>plugin.tx_logicalform_pi1.mail.client { bcc = name@my.provider.com, \ webmaster@my-domain.com }</pre> | |
| mail.client.field | string | The name of the field in the form, which contained the mail address of the visitor (see the mail tutorial on page 16) Example: <pre>plugin.tx_logicalform_pi1.mail.client { field = mail }</pre> | |

| Property: | Data type: | Description: | Default: |
|-------------------------|------------|---|----------|
| mail.client.body | string | The name of the field for mail subject and mail message in the language configuration (see the mail tutorial on page 16) Example: <pre>plugin.tx_logicalform_pi1.mail.client { body = mailclient }</pre> | |
| mail.client.signature.x | string | Signature of the report mail Example: <pre>plugin.tx_logicalform_pi1.mail.client { signature.0 = My provider company signature.1 = My provider name signature.2 = name@my-provider.com }</pre> | |

Language and translation

There are translations for some words and phrases in the file

- pi1/locallang.xml

especially for

- error_mail: "There is a mistake in your mail address ###MAIL###"
- error_not_accepted: "The system doesn't accept this value!"
- error_not_accepted_checkbox "The system doesn't accept one of the values!"
- "error_not_accepted_select: "The system doesn't accept one of the values!"
- legend: "Logical Form"
- mail_report_subject: "Logical Form Report"
- mail_report_text: A text with markers.

If you want to change the values,

- you can edit locallang.xml (but maybe you will get trouble in case of updating the extension)
- or edit the array “_Local_Lang” (recommended).

If you want to edit the array “_LOCAL_LANG” please take a look in the tutorial “How to configure language for fields and values” on page 15.

Files

Files for the configuration of language items and for documentation.

| File: | Description: |
|----------------------------------|---|
| pi1/class.tx_logicalform_pi1.php | Main PHP-class used to display the form and the confirmation. The class validates mail addresses and sends mails to logical_form and the subscriber or unsubsubscriber. |
| pi1/locallang.xml | Language file for class.tx_logicalform_pi1.php |
| pi1//template.html | Template file for the form. |
| ext_php_api.dat | API documentation of the logical_form extension. |
| pi1/samples/* | Setup code for the tutorial |

Helpful suggestions

If you have helpful suggestions, feel free to contact me: Dirk Wildt, dirk.wildt@think-visually.com.

FAQ

SimulateStaticDocuments and RealUrl

There isn't any problem known.

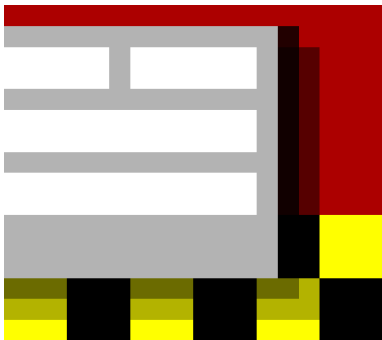
Xhtml 1.0 Strict



The notation of field arrays like `<label for="tx_logicalform_pi1[mail_id] ...` is not accepted in Xhtml 1.0 Strict

Further Information

About the plugin icon



The concept of the icon:

- The grey rectangle is symbolizing a form.
- The stripe "follow me" is symbolizing content management as an powerful help.
- The color red is symbolizing the color of think visually, the company which created the logical_form extension.

Other extensions published by think visually

-  majordomo: A web frontend for subscribing in majordomo lists and for unsubscribing from majordomo lists.

To-Do list

- Convert to Xhtml 1.0 Strict: The notation of field arrays like `<label for="tx_logicalform_pi1[mail_id] ...` is not accepted in Xhtml 1.0 Strict

Changelog

- 0.0.1: Initial release