

# EXT: TMailform

Extension Key: **pil\_mailform**

Copyright 2007, Sebastian Busch <typo3-dev@billiton.de>

Copyright 2005-2007, <typo3@pil.dk>

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](http://www.typo3.com)

## Table of Contents

<b>EXT: TMailform</b> .....	<b>1</b>	Flexform reference.....	1
<b>Introduction</b> .....	<b>1</b>	TypoScript reference.....	1
What does it do?.....	1	<b>Known problems</b> .....	<b>1</b>
Screenshots.....	1	<b>To-Do list</b> .....	<b>1</b>
<b>Configuration</b> .....	<b>1</b>	<b>Changelog</b> .....	<b>1</b>
Workflow.....	1	<b>Suggestions from users not yet implemented</b> .....	<b>1</b>

## Введение

### Что оно делает?

TMailform обеспечивает простой путь поместить почтовую форму на Ваш вебсайт, без потери контроля над дизайном почтовой формы. Здесь используются шаблоны и легко конфигурируется через flexforms - таким образом необходим не только typoscript.

Это расширение было первоначально написано Tonni Aagesen (pil.dk).

С языком и в специфической поддержке базы данных в версии > 3.0.0, потенциал этого расширения в действительности перерос оригинальную цель. Оно может использоваться, например, для того, чтобы регистрировать/управлять пользователями внешнего интерфейса, подписываться на новости... любая, не слишком сложная манипуляция с данными - используя Ваше воображение :)

Основные особенности:

- Основывается на шаблонах - свобода в дизайне
- Простая конфигурация через flexform
- Множество методов проверки правильности ввода для необходимых полей со стороны сервера
- Сообщение об ошибках в предопределенных полях
- Адреса электронной почты получателя не показаны на странице/источнике
- Значения восстанавливаются/сохраняются в базе данных

## Скриншоты

Plugin:	TMailform
Plugin Options:	
DEF:	
General settings	Field validation   Mail headers   SMTP settings
Testmode	<input checked="" type="checkbox"/>
Static subject	TMailform test
Subject user override	<input checked="" type="checkbox"/>
User subject prefix	From TMailform:
Use recipient	<input checked="" type="radio"/> Static recipient(s) <input type="radio"/> Dynamic recipient
Static recipient(s)	user1@domain.tld user2@domain.tld
Dynamic recipient	Choose recipient Sales division; sales@domain.tld Development division; dev@domain.tld
Copy to user	User select
User copy subject	Copy of Your e-mail to domain.tld
Template	fileadmin/templates/tmai

К сожалению, я был в состоянии вставить только один скриншот, чтобы избежать превышение предела загрузки расширения 100 КБ. Остальные скриншоты скоро будут доступны на <http://t3dev.pil.dk>.

## Конфигурация

Расширение TMailform главным образом использует конфигурацию через шаблон и flexform. Конфигурацией через шаблон я подразумеваю, что подразделение шаблон формы играет роль в установке через обозначение полей формы, как мы увидим позже. Фактически, концепция TMailform в том, что маркеры генерируются динамически, соответственно, из атрибутов имени полей формы.

Если Вы запутались в понятиях, не волнуйтесь. Только помните это эмпирическое правило; Чтобы быть обработанным, все атрибуты имени полей формы должны иметь такую форму "tx\_pilmailform\_pi1 [<тип поля>] [<имя поля>]", где <тип поля> соответствует типу поля формы, и <имя поля> идентификатор имени полей. Пример:

```
<input type="text" name="tx_pilmailform_pi1[tx_pi1][предмет]" />
```

<Имя поля>, в этом случае "предмет", распознан TMailform, и расширение создаст до трех маркеров шаблона, при отправке формы. Первый маркер будет создан как "### SUBJECT\_VAL ###", и будет заменен значением из поля. Если Вы определили это поле, как обязательное, и поле было, будет создан второй маркер "### SUBJECT\_ERR ###". Этот маркер будет заменен на то, что Вы определили во flexform. Кроме того, если Вы определили сообщение об ошибках для определенного необходимого поля, маркер "### SUBJECT\_ERR\_TXT ###" будет создан и заменен указанным сообщением.

Если форма отправлена, и необходимые поля, определенные Вами, не пусты, то значения из полей формы будут доступны как маркеры в MAIL и MAIL\_USERCOPY, как подразделения в "### <имя поля> \_VAL ###" примечание.

Довольно просто? Есть, однако, некоторые особые случаи, где вышеупомянутое описание не применяется точно так. Я объясню их примером в разделе делопроизводства.

## Делопроизводство

При установке TMailform Вы должны будете сначала определить, какую информацию Вы желаете собрать из формы. Далее я опишу пример делопроизводства. Шаблон примера включен в это расширение.

### Создание шаблона

В этом примере я желаю собрать следующую информацию от пользователя:

- имя пользователя
- адрес электронной почты пользователя
- представляет ли пользователь частного человека или компанию
- уровень знания Turbo3 пользователя
- описание задания пользователя
- тема запроса пользователя
- сообщение от пользователя
- резюме как прикрепленный файл

Далее я желаю направить электронную почту на соответствующий адрес электронной почты получателя, и предоставляю пользователю опцию получения копии сообщения.

Теперь, имя, электронная почта и тема являются довольно явными, но мы можем сделать некоторое волшебство с

этимися полями, как, я объясню в последней части этого примера делопроизводства. Я хочу, чтобы эти поля были только стандартными полями ввода текста, и я хочу, чтобы поля были обязательными. Вот как будет выглядеть мой шаблон:

```
<td class="###SUBJECT_ERR###">
    ###SUBJECT_ERR_TXT###
    <input type="text" name="tx_pilmailform_pi1[tx][subject]" value="###SUBJECT_VAL###" />
</td>
<td class="###NAME_ERR###">
    ###NAME_ERR_TXT###
    <input type="text" name="tx_pilmailform_pi1[tx][name]" value="###NAME_VAL###" />
</td>
<td class="###EMAIL_ERR###">
    ###EMAIL_ERR_TXT###
    <input type="text" name="tx_pilmailform_pi1[tx][email]" value="###EMAIL_VAL###" />
</td>
```

Также, мы можем добавить сообщение от пользователя, как поле текстовой области:

```
<td class="###MESSAGE_ERR###">
    ###MESSAGE_ERR_TXT###
    <textarea name="tx_pilmailform_pi1[textarea][message]">###MESSAGE_VAL###</textarea>
</td>
```

Затем, мы хотим знать, представляет ли пользователь частного человека или компанию. С этой целью я буду использовать радио-тип поля:

```
<td class="###USER_TYPE_ERR###">
    ###USER_TYPE_ERR_TXT###
    <input type="radio" name="tx_pilmailform_pi1[radio][user_type]" value="private" ###USER_TYPE_PRIVATE_VAL### /> Private
    <br />
    <input type="radio" name="tx_pilmailform_pi1[radio][user_type]" value="buisness" ###USER_TYPE_BUISNESS_VAL### />
    Buisness
</td>
```

Вы обращали внимание на различие в значениях маркеров? Поскольку у радио-типа поля могут быть выбнаны несколько значений, мы должны определить, какое поле было выбрано. TMailform делает это, называя эти маркеры значением поля, таким образом если у выбранного поля есть значение "private", будет создан маркер "###USER\_TYPE\_PRIVATE\_VAL###". Мы можем обобщить это так, чтобы множественные выбранные типы поля генерировали маркеры в форме "###<имя поля>\_<значение поля>\_VAL ###".

В нашем примере, если пользователь выбирает "Private" тогда, маркер "### USER\_TYPE\_PRIVATE\_VAL ###" создан и заменен на 'отмечен = "отмечен"'.

```
<td class="###LEVEL_ERR###">
    ###LEVEL_ERR_TXT###
    <select name="tx_pilmailform_pi1[select][level]">
        <option value="">Choose best fit</option>
        <option value="Beginner" ###LEVEL_BEGINNER_VAL###>Beginner</option>
        <option value="Advanced" ###LEVEL_ADVANCED_VAL###>Advanced</option>
        <option value="Expert" ###LEVEL_EXPERT_VAL###>Expert</option>
    </select>
</td>
```

Поскольку это также поле со множественным выбором, действует тот же самый принцип замены значения маркера, как и для радио кнопок. Если пользователь не делает выбор или выбирает опцию "Choose best fit", TMailform распознает это как ошибку проверки правильности ввода поля, и маркером заменит на маркер ошибки. Это действие предпринято, потому что у этой опции нет никакого значения (value="").

Теперь, мы также хотели собрать информацию, о том, работа какого типа работа соответствует нашему пользователю. Мы хотим учесть, что пользователь мог бы соответствовать больше чем одному из описаний, которые мы приводим. С этой целью мы будем использовать блок выбора, отличие которого в том, что мы позволяем пользователю выбрать больше одного значения.

```
<td class="###JOBDESCRIPTION_ERR###">
    ###JOBDESCRIPTION_ERR_TXT###
    <select name="tx_pilmailform_pi1[select][job_description]" multiple="multiple">
        <option value="Developer" ###JOB_DESCRIPTION_DEVELOPER_VAL###>Developer</option>
        <option value="Sales person" ###JOB_DESCRIPTION_SALES_PERSON_VAL###>Sales person</option>
        <option value="Manager" ###JOB_DESCRIPTION_MANAGER_VAL###>Manager</option>
    </select>
</td>
```

Есть только небольшое различие между этим кодом и кодом для пользовательского уровня блока выбора. Конечно, мы добавляем 'multiple="multiple"', чтобы сделать доступным выбор более одного значения. Но мы также помещаем дополнительное пустое множество "[]" в название поля блока выбора. Мы должны сделать это, чтобы заставить TMailform распознать все выбранные пользователем варианты. Также обратите внимание, что пробелы позволены в value="" поля, мы лишь должны заменить пробелами с символами подчеркивания в маркерах значения.

Итак, шаблон сделан! Я слышу, как вы бубните: "Хм... что относительно копии получателю и пользователю сообщения. И, где волшебство, о котором говорилось?". Хорошо, начнем с получателя и "копии пользователю" и еще

кое-что о волшебстве... Я объединю эти темы в следующем разделе.

Пока Вы можете поместить это в свой шаблон и сохранить файл где-нибудь в fileadmin/:

```
<td class="###FILE1_ERR###">
  ###FILE1_ERR_TXT###
  <input type="file" name="tx_pilmailform_pi1[file][file1]" />
</td>
<td class="###FILE2_ERR###">
  ###FILE2_ERR_TXT###
  <input type="file" name="tx_pilmailform_pi1[file][file2]" />
</td>
<td class="###MULTI_RECIPIENT_ERR###">
  <select name="tx_pilmailform_pi1[select][multi_recipient]">###MULTI_RECIPIENT_OPTIONS###</select>
</td>
<td>
  <input type="checkbox" name="tx_pilmailform_pi1[checkbox][user_copy]" value="yes" ###USER_COPY_YES_VAL### />
</td>
<td colspan="2">
  <input type="submit" name="tx_pilmailform_pi1[submit][submit]" value="Send!" />
</td>
```

Примечание: Если Вы используете javascript, чтобы отправить форму, должно быть установлено "tx\_pilmailform\_pi1[submit][submit]".

## Волшебство раскрывается - настройка расширения

Хорошо, волшебство может быть сильное слово, но однако мы можем сделать все довольно привлекательным и аккуратным, что я сейчас и покажу. Я предполагаю, что Вы загрузили и установили расширение TMailform из репозитория и вставили его на некоторой странице. Этот раздел лучше читать со скриншотами.

Ниже "волшебные поля" будут объяснены:

### Поле: [тема] ([subject])

TMailform дает Вам опцию определения статической темы или позволять пользователю определить тему. Если Вы выберете "Тема переопределяется пользователем" ("Subject user override") во flexform, то значение поля [тема] ([subject]) будет установлено как тема почты.

### Поля: [имя]([name]), [фамилия]([surname]) и [email]

Вы можете установить заголовки почты "От", и "Ответ" определяя информацию пользователя. Если Вы установите ""От" и "Ответ", как переопределенный пользователем заголовок, то в качестве почтовых заголовков будут использоваться значения полей [имя], [фамилия] и [email]. Кроме того, поле [email], если заполнено, будет проверено на соответствие адресу электронной почты.

### Поле: [multi\_recipient]

Во flexform Вы можете выбрать или статического получателя или позволить пользователю выбирать получателя. Как показано в примере шаблона, Вы можете поместить "### MULTI\_RECIPIENT\_OPTIONS ###" маркер в Вашем шаблоне. Этот маркер будет заменен опциями, определенными во flexform.

Любая строка в текстовой области flexform, содержащая точку с запятой, будет распознана как "<текст>; <электронная почта>" опции будут сгенерированы соответственно. Строки без точки с запятой будут расценены как "опция без значения" (value="").

Кроме того, когда форма подтверждена, создается дополнительный маркер "### MULTI\_RECIPIENT\_OPTVAL\_VAL ###", у которого будет значение текста в тегах "<option>text</option>".

### Поле: [Копия пользователю] ([user\_copy])

Во flexform Вы можете установить, должен ли пользователь получить копию почты, или Вы можете позволить пользователю самому решать. Если TMailform будет конфигурирован как "Пользователь выбирает" ("User select"), то расширение будет искать значение поля [user\_copy] и пошлет копию пользователю, если пользователем будет определен правильный адрес электронной почты.

Итак, все, что нужно сделать, определяет шаблон, который мы создали ранее, сейчас же мы начнем проверять TMailform. С целью тестирования mailform Вы можете установить расширение в "Режиме тестирования", и почта не будет посылаться. Почта, вместо этого, будет показана в Вашем браузере наряду с небольшим количеством информации о заголовках и т.д.

## Дополнительные маркеры, которые могли бы пригодиться для подраздела TMAIL\_FORM

Любое поле в fe\_user таблице можно установить как маркеры ("### FEUSER\_\* ###") в подразделении шаблона, Какая информация о пользователе будет доступна, зависит от Вашей fe\_user таблицы. Некоторые примеры:

```
###_FEUSER_USERNAME###
```

```
###_FEUSER_EMAIL###
```

```
###_FEUSER_WWW###
```

## Дополнительные маркеры, которые могли бы пригодиться для подразделений TMAIL\_MAIL\*

###\_DATE### Date of mail sent

###\_TIME### Time of mail sent

###\_SITE### Site name

###\_PATH### Page path

###\_PAGE### Page title

###\_PID### Page ID

###\_IP### Client IP

### Несколько языков и маркеры

Начиная с версии 3.0.0 Вы можете легко использовать несколько языков в TMailform. Все, что Вы должны сделать - создать locallang xml файл и поместить его где-нибудь в директории fileadmin (представлен файл примера). Теперь Вы можете теперь использовать языковые маркеры в Вашем шаблоне.

Понятие языковых маркеров "###\_LL\_<lang key>###", где <lang key> индекс languageKey из Вашего locallang xml файла. Это очень просто.

### Работа с полями и значениями

На вкладке "Проверка полей" ("Field settings") Вы найдете относительно простой, и в то же время сильный способ обработки поля и значения. Впоследствии я объясню синтаксис отображения значений полей в/из функций. Каждая строка в "Необходимых значениях", "Значениях по умолчанию" и "Сохраненных значениях" состоит из трех разделов, отделенных точками с запятой:

```
field_value[,field_value]; method[:parameter]; [processor:parameter[:parameter]]
```

Позвольте нам просмотреть каждый раздел:

- "field\_value[,field\_value]" - Этот раздел представляет любой маркер значений, сгенерированный TMailform. В некоторых случаях, в том числе в сохраненных значениях это может быть отделенный список, разделенных запятой значений маркеров.
- "method[:parameters]" - Этот раздел определяет, какой метод обработки маркера(ов) значения. Некоторые методы требуют параметров.
- "[processor:parameter[:parameter]]" - Этот раздел определяет, как обработать данные относительно раздела метода. В некоторых случаях этот раздел является дополнительным. Если присутствует, процессор берет один или несколько параметров.

Пожалуйста, ищите примеры во flexform и справочной информации по typoscript.

**Примечание:** Вы можете вставить комментарий, устанавливая первый символ как # (т.е. "#, Это - комментарий"), и эта строка не будет анализироваться.

Версия 3.0.0 TMailform предлагает методы для простых взаимодействий с базой данных. Во время установки TMailform Вы получите возможность установить, как и к каким таблицам и полям может обратиться TMailform. Невозможно, из-за ограничения в НИХ, показать, как Вы можете сконфигурировать доступ для множественных таблиц, но синтаксис таков:

```
<table1>:<field1>,...,<fieldN>;...; <tableN>:<field1>,...,<fieldN>
```

Примечание: Недопустимые запросы регистрируются, используя t3lib\_div:: sysLog и Вы можете установить поведение sysLog, используя Установки.

## Справочная информация синтаксического анализатора поля/значения

Необходимые значения				
Метод	Параметры	Процессор	Параметры	Примечания

notEmpty	None	text	string	"string" может быть любым текстом, который Вы хотите отобразить в случае ошибки проверки правильности. "lang key" является клавишей массива из locallang_template.php файла. Результат регулярного выражения может быть инвертирован, добавлением префикса "!". Некоторые Примеры: - "Требуется целое число в значении поля": regex: / [0-9]/; text: Нужно целое число - "Требуется, чтобы значение в поле не содержало целое число": regex:! / [0-9]/; text: Пожалуйста, все, кроме целых чисел
		lang	lang key	
notEquals	marker Useful if submitting a password, and you want to confirm it. Example: "NAME_VAL"	text	string	
		lang	lang key	
inList	Comma separated list of values (item1,item2,...,itemN)	text	string	
		lang	lang key	
regex	Regular expression (perl compatible).	text	string	
		lang	lang key	

#### Значения по умолчанию

Метод	Параметры	Процессор	Параметры	Примечания
plain	None	text	string	"string" может быть любым текстом, который Вы хотите отобразить в случае ошибки проверки правильности.
		mark	predefined marker	Любой предопределенный маркер. Например: _FEUSER_USERNAME
db	None	get	table:field[,field]:where clause	Смотрите раздел "Сохраненные значения" этой таблицы.

#### Сохраненные значения

Метод	Параметры	Процессор	Параметры	Примечания
db	None	get	table:field[,field]:where clause	Сгенерированные запросы SQL сгруппированы по операторам, и начиная с SQL-оператора "INSERT" берутся не по операторам, мы нуждаемся в уникальном идентификаторе пр. Запрос. Идентификатор <b>должен быть</b> предустановлен со звездочкой (*)  Вполне возможно, одним запросом охватить множество строк, если он воздействуют на ту же самую таблицу базы данных и имеют то же самое выражение. Пример:  NAME_VAL; db; new:tx_pilmailform_log:name:*1 EMAIL_VAL; db; new:tx_pilmailform_log:email:*1  эквивалентно:  NAME_VAL,EMAIL_VAL; db; new:tx_pilmailform_log:name,email:*1  Оба будут генеровать отдельный запрос:  INSERT INTO tx_pilmailform_log (name,email) VALUES (NAME_VAL, EMAIL_VAL);  <b>Примечание:</b> синтаксический анализатор может обработать только <b>одно выражение</b> с пр. оператором. <b>Примечание:</b> сгенерированные запросы показываются, когда TMailform находится в режиме тестирования.
		new	table:field[,field]:unique identifier	
		add	table:field[,field]:where clause	
		update	table:field[,field]:where clause	

## Справочная информация по Flexform

Основное			
Поле:	Тип данных:	Описание:	По умолчанию:
Режим тестирования	Флажок	Если включено, информация о почте будет показана в браузере, и НИКАКАЯ почта не посылается.	Не включен
Статическая тема	Строка	Определяет статическую тему для почты. Вы можете использовать маркеры в этом поле.	Пустая
Тема переопределяется пользователем	Флажок	Если включено, статическая тема будет заменена полем ['text'] ['subject']	Не включен
Префикс темы пользователя	Строка	Если установлено, пользовательская тема будет начинаться с этого. Вы можете использовать маркеры в этом поле.	Пустая

Основное			
Использовать получателя	Выбор	Используйте или статического или динамического получателя почты	"static recipient"
Статические получатель(и)	Строка	Почта посылается на этот/эти адрес(а) электронной почты. Каждая строка, как ожидается, - правильный адрес электронной почты.	Пустая
Динамические получатель(и)	Строка	Получатель определен как "<имя в блоке выбора HTML>; <правильный адрес электронной почты>". Любая строка, не содержащая точку с запятой, обрабатывается как "только строка" в блоке выбора.	Пустая
Копия пользователю	Выбор	Нет: Пользователь не будет получать копию почты. Да: Пользователь получит копию. Пользовательский выбор: Если поле [checkbox] [user_copy] будет обнаружен как выбранное, то копия будет послана пользователю.	"Нет"
Тема копии пользователя	Строка	Если не пустая, у пользовательской копии почты будет эта строка как тема. Если оставлена пустой, тема будет как в оригинальной почте.	Пустая
Шаблон	Строка	Путь к файлу шаблона	Пустая
Многоязыковая поддержка	Флажок	Включите, чтобы обеспечить многоязыковую поддержку. Требуется locallang файла.	
Locallang file	Строка	Путь к locallang файлу	
Проверка полей			
Поле:	Тип данных:	Описание:	По умолчанию:
Заменить маркеры ошибок этим	Строка	Любые ###<field name>_ERR### маркеры ошибок будут заменены этой строкой.	Пустая
Требующиеся значения	Строка	<p>Проверяет правильность полей определяемым пользователем методом. Каждая строка состоит из трех разделов "<b>поле</b>; <b>проверка правильности</b>; <b>ответ</b>" (отделенные точкой с запятой). "проверка правильности" и "ответ" могут каждый быть разделены непосредственно на две части: "<b>поле</b>; <b>method:expression</b>; <b>error_key:error_msg</b>&gt;" (отделенные двоеточием).</p> <p>Подробнее:</p> <ul style="list-style-type: none"> <li>● <b>field</b> название поля.</li> <li>● <b>Method</b> может быть одним из <b>notEmpty</b> (выражения не применяются), <b>inList</b> (выражение: список строк, разделенных запятой, напр. "str1, strN") или <b>regex</b> (выражение: регулярное выражение perl, напр. ("/^test.*"/)).</li> <li>● <b>error_key</b> может быть только <b>text</b> в этой точке.</li> </ul> <p><b>error_msg</b> строка, которой заменяется маркер ###&lt;field name&gt;_ERR_TXT###, если поле не проверяется на правильность.</p> <p>Пример: SUBJECT_VAL; notEmpty; text: Пожалуйста, заполните поле темы SUBJECT_VAL; regex:/^test.*;/ text: Объект должен начинаться с "тест" MESSAGE_VAL; notEmpty; text: Пожалуйста, заполните поле сообщения FILE1_VAL; notEmpty; text: Пожалуйста, пришлите файл FILE1_VAL; inList: image/png, text/plain; text: Недопустимый тип файла USER_TYPE_VAL; notEmpty; LEVEL_VAL; notEmpty; text: Пожалуйста, выберите уровень</p> <p><b>Примечание: Для типа поля "файл" проверка выполняется на mime-типе файла.</b></p>	Пустая
Значения по умолчанию	Строка	<p>Позволяет установить некоторые значения по умолчанию для полей в шаблоне. Для полей в шаблоне примера, который идет с TMailform, Вы можете сделать:</p> <p>NAME_VAL; plain; text: Tonni Agesen EMAIL_VAL; plain; mark: FEUSER_EMAIL JOB_DESCRIPTION_DEVELOPER_VAL; plain; text: selected="selected" USER_COPY_YES_VAL; plain; text: checked="checked" SUBJECT_VAL; db; get: pages: title: uid= _PID</p>	Пустая

Основное			
Сохраненные значения	Строка	Позволяет сохранить значения, когда форма отправлена. Вот некоторые примеры:  # INSERT NAME_VAL,EMAIL_VAL; db; new:tx_pilmailform_log:name,email:*1 # UPDATE/INSERT EMAIL_VAL,NAME_VAL; db; add:tt_address:email,name=email=EMAIL_VAL # UPDATE NAME_VAL; db; update:fe_users:name:uid=_FEUSER_UID	Пустая
Формат маркеров данных	Строка	Представление данных - смотрите <a href="http://php.net/manual/en/function.strftime.php">http://php.net/manual/en/function.strftime.php</a>	%Y-%m-%d
Формат маркера времени	Строка	Также, как dateMarker	%T
Почтовые заголовки			
Поле:	Тип данных:	Описание:	По умолчанию:
"From" имя	Строка	Имя в заголовке "From:".	Пустая
"From" email	Строка	Email в заголовке "From:".	Пустая
"From" переопределенный пользователем заголовок	Флажок	Если включено, поля ["text"]["name"] и ["text"]["email"] будут использоваться, чтобы генерировать заголовок "От".	Не включен
"Reply-To" имя	Строка	Имя в заголовке "Reply-To:".	Пустая
"Reply-To" email	Строка	Email в заголовке "Reply-To:".	Пустая
"Reply-To" переопределенный пользователем заголовок	Флажок	Если включено, поля ["text"]["name"] и ["text"]["email"] будут использоваться, чтобы генерировать заголовок "Reply-To:".	Не включен
Формат переопределения имени	Выбор	Если поле ["text"]["surname"] существует, устанавливает соответствующий формат замены имени.	"name surname"
Раскрыть email получателя в пользовательской копии	Флажок	Если включено, заголовки "From:" и "Reply-To:" будут установлены на первого статического или динамического получателя.	Не включен
"Cc" адреса	Строка	Посылает почту как "Cc" на этот/этим адрес(а) email. Каждая строка, как ожидается, должна быть правильным адресом email.	Пустая
"Bcc" адреса	Строка	Посылает почту как "Bcc" на этот/этим адрес(а) email. Каждая строка, как ожидается, должна быть правильным адресом email.	Пустая
"Кодирование передачи информационного наполнения"	Выбор	Устанавливает заголовок почты "Кодирование передачи информационного наполнения"	"8bit"
Тип информационного наполнения	Выбор	Определяет формат почты HTML / обычный текст. Они определяются как различные подразделения в шаблоне - <b>Примечание:</b> , Выбрав почту HTML, еще требуется подразделения шаблона для почты с обычным текстом, иначе Вы оставите пользователей с почтовым клиентом на основе обычного текст без информации.	"text/plain"
Кодировка	Строка	Устанавливает параметры кодировки для заголовков.	\$GLOBALS['TSFE']->renderCharset
Настройки SMTP			
Способ отправки почты	Выбор	Выбрать метод для отправки почты. PHP mail(): Стандартная функция php mail() . sendmail: Использовать непосредственно демона sendmail. smtp: Использовать определяемый пользователем smtp сервер.	"PHP mail()"
Путь к sendmail	Строка	Путь к программе sendmail.	"/usr/sbin/sendmail"
SMTP хост	Строка	Хост для соединения по SMTP.	"localhost"
SMTP порт	Целое	Порт для соединения по SMTP.	25
Использовать SMTP аутентификацию	Флажок	Если включено, будут использоваться имя пользователя и пароль для подключения SMTP.	Не включен
Имя пользователя SMTP	Строка	Имя пользователя SMTP	Пустая
Пароль SMTP	Строка	Пароль SMTP	Пустая

## Справочная информация по TYPOScript



Примечание: Значения TS переопределяют значения во flexform.

Основное			
TS	Тип данных:	Описание:	По умолчанию:
testmode	Целое	Если равно 1, информация о письме будет отображена в браузере и никакие письма посылаться НЕ БУДУТ.	
staticSubject	Строка	Определяет статическую тему для почты. Вы можете использовать маркеры в этом поле.	
overrideSubject	Целое	Если равно 1, статическая тема будет заменена полем ['text'] ['subject']	
userSubjectPrefix	Строка	Если установлено, пользовательская тема будет начинаться с этого. Вы можете использовать маркеры в этом поле.	
typeofRecipient	Целое	0 = статический получатель(и) 1 = динамический получатель	
staticRecipient	Строка	Почта посылается на этот/эти адрес(а) электронной почты. Каждая строка, как ожидается, - правильный адрес электронной почты. Например: plugin.tx_pilmailform_pi1.staticRecipient ( user@mail.domain.tld )	
dynamicRecipient	Строка	Получатель определен как "<имя в блоке выбора HTML>; <правильный адрес электронной почты>". Любая строка, не содержащая точку с запятой, обрабатывается как "только строка" в блоке выбора. Например: plugin.tx_pilmailform_pi1.dynamicRecipient ( Please select recipient user@mail.domain.tld user@mail.domain.tld )	
copyToUser	Целое	0 = Нет: Пользователь не получает копию письма. 1 = Да: Пользователь получит копию письма. 2 = По выбору: Если поле ['checkbox']['user_copy'] обнаружено как установленное, копия письма будет послана пользователю.	
userCopySubject	Строка	Если не пустая, у пользовательской копии почты будет эта строка как тема. Если оставлена пустой, тема будет как в оригинальной почте.	
template	Строка	Путь к файлу шаблона	
useLL	Целое	0 = многоязыковая поддержка включена 1 = многоязыковая поддержка выключена	
LLFile	Строка	Путь к locallang файлу	
Проверка полей			
TS:	Тип данных:	Описание:	По умолчанию:
errorSubstitution	Строка	Любые ###<field name>_ERR### маркеры ошибок будут заменены этой строкой.	

Основное			
requiredValues	Строка	<p>Проверяет правильность полей определяемым пользователем методом. Каждая строка состоит из трех разделов "<b>поле</b>; <b>проверка правильности</b>; <b>ответ</b>" (отделенные точкой с запятой). "проверка правильности" и "ответ" могут каждый быть разделены непосредственно на две части: "<b>поле</b>; <b>method:expression</b>; <b>error_key:error_msg</b>&gt;" (отделенные двоеточием).</p> <p>Подробнее:</p> <ul style="list-style-type: none"> <li>● <b>field</b> название поля.</li> <li>● <b>Method</b> может быть одним из <b>notEmpty</b> (выражения не применяются), <b>inList</b> (выражение: список строк, разделенных запятой, напр. "str1, strN") или <b>regex</b> (выражение: регулярное выражение perl, напр. ("/^test.*"/)).</li> <li>● <b>error_key</b> может быть только <b>text</b> в этой точке. <b>error_msg</b> строка, которой заменяется маркер <b>###&lt;field name&gt;_ERR_TXT###</b>, если поле не проверяется на правильность.</li> </ul> <p>Пример:  plugin.tx_pilmailform_pi1.requiredValues (  SUBJECT_VAL; notEmpty; text: Пожалуйста, заполните поле темы  SUBJECT_VAL; regex:/^test.*;/ text: Объект должен начинаться с "тест"  MESSAGE_VAL; notEmpty; text: Пожалуйста, заполните поле сообщения  FILE1_VAL; notEmpty; text: Пожалуйста, пришлите файл  FILE1_VAL; inList: image/png, text/plain; text: Недопустимый тип файла  USER_TYPE_VAL; notEmpty;  LEVEL_VAL; notEmpty; text: Пожалуйста, выберите уровень  )  Примечание: Для типа поля "файл" проверка выполняется на mime-типе файле.</p>	
defaultValues	Строка	<p>Позволяет установить некоторые значения по умолчанию для полей в шаблоне. Для полей в шаблоне примера, который идет с TMailform, Вы можете сделать:</p> <pre>plugin.tx_pilmailform_pi1.defaultValues ( NAME_VAL; plain; text:Tonni Aagesen EMAIL_VAL; plain; mark:FEUSER_EMAIL JOB_DESCRIPTION_DEVELOPER_VAL; plain; text:selected="selected" USER_COPY_YES_VAL; plain; text:checked="checked" SUBJECT_VAL; db; get:pages:title:uid=_PID )</pre>	
storeValues	Строка	<p>Позволяет сохранить значения, когда форма отправлена. Вот некоторые примеры:</p> <pre>plugin.tx_pilmailform_pi1.storeValues ( # INSERT NAME_VAL,EMAIL_VAL; db; new:tx_pilmailform_log:name,email:*1 # UPDATE/INSERT EMAIL_VAL,NAME_VAL; db; add:tt_address:email,name=email=EMAIL_VAL # UPDATE NAME_VAL; db; update:fe_users:name:uid=_FEUSER_UID )</pre>	
dateMarkerFormat	Строка	<p>Представление данных - смотрите <a href="http://php.net/manual/en/function.strftime.php">http://php.net/manual/en/function.strftime.php</a></p>	%Y-%m-%d
timeMarkerFormat	Строка	<p>Также, как dateMarker</p>	%T
Почтовые заголовки			
TS:	Тип данных:	Описание:	По умолчанию:
fromName	Строка	Имя в заголовке "From:".	
fromMail	Строка	Email в заголовке "From:".	
overrideFromHeader	Целое	0 = использовать самоопределяющийся заголовок "From:" 1 = использовать заголовок "From:", определяемый пользователем	
replyToName	Строка	Имя в заголовке "Reply-To:".	

Основное			
replyToMail	Строка	Email в заголовке "Reply-To:"	
overrideReplyToHeader	Целое	0 = использовать самоопределяющийся заголовок "Reply-To:" 1 = использовать заголовок "Reply-To:", определяемый пользователем	
overrideHeaderNameFormat	Выбор	Если определены поля ['text']['surname'] устанавливает формат их вывода: 0 = "имя фамилия" 1 = "фамилия, имя"	"имя фамилия"
discloseRecipient	Check	Если включено, заголовки "From:" и "Reply-To:" будут установлены на первого статического или динамического получателя.	Не включен
Cc	Строка	Посылает почту как "Cc" на этот/этим адрес(а) email. Каждая строка, как ожидается, должна быть правильным адресом email.	
Bcc	Строка	Посылает почту как "Bcc" на этот/этим адрес(а) email. Каждая строка, как ожидается, должна быть правильным адресом email.	
contentTransferEncoding	Целое	Устанавливает заголовок почты "Кодирование передачи информационного наполнения"  0 = 8bit 1 = 7bit 2 = base64 3 = binary 4 = quoted-printable	
contentType	Целое	Определяет формат почты HTML / обычный текст. Они определяются как различные подразделения в шаблоне - <b>Примечание:</b> , Выбрав почту HTML, еще требуется подразделения шаблона для почты с обычным текстом, иначе Вы оставите пользователей с почтовым клиентом на основе обычного текст без информации.  0 = простой текст 1 = текст html	
charset	Строка	Устанавливаете параметры кодировки заголовков. Изменение Charset НЕ конвертирует данные в выбранную кодировку. (возможно будет сделано в будущем)	
Установки SMTP			
useMailer	Целое	Метод отправки почты.  0 = "PHP mail()" 1 = "sendmail" 2 = "smtp" -1 = "disable mail" (useful if you only want to store values)	
sendmailPath	Строка	Путь к программе sendmail.	
smtpHost	Строка	Хост для соединения по SMTP.	
smtpPort	Целое	Порт для соединения по SMTP.	
smtpAuth	Целое	0 = Без авторизации по SMTP 1 = Использовать авторизацию по SMTP	
smtpUser	Строка	Имя пользователя SMTP	
smtpPasswd	Строка	Пароль SMTP	

## Известные проблемы

В текущем релизе не выявлены :)

## Список To-Do

- Расширенная проверка правильности email (проверяют правильность адреса, используя почтовый сервер).
- Поддержка разбиения на страницы так, чтобы TMailform мог бы использоваться для многостраничных опросов.
- Поддержка Captcha.

# Список изменений

## Version 3.0.4

- **IMPORTANT** Fixed security bug in included PHPMailer class which allowed remote command execution

## Version 3.0.3

- Fixed bug <http://bugs.typo3.org/view.php?id=4584>
- Fixed bug <http://bugs.typo3.org/view.php?id=4583>.
- Fixed bug <http://bugs.typo3.org/view.php?id=4582>.
- Fixed bug <http://bugs.typo3.org/view.php?id=4522>.

## Version 3.0.2

- Re-enabled parsing default values by url/pivars (Thanks to Florian for pointing out its disappearance).

## Version 3.0.1

- Added option to disclose recipient email address in user copy mail (<http://bugs.typo3.org/view.php?id=3620>). See documentation.
- Added possibility to use surname in mail headers (<http://bugs.typo3.org/view.php?id=4153>). See documentation.
- Fixed bug where user\_copy field was always set (<http://bugs.typo3.org/view.php?id=4152>).
- Fixed bug where a semi colon in regex parameters broke validation (<http://bugs.typo3.org/view.php?id=4291>).
- Fixed bug causing a fatal error in some cases due to an invalid function call (<http://bugs.typo3.org/view.php?id=4268>).
- Added an auto generated marker for multi\_recipient option text. See documentation.
- Added negation to regex in required values. See documentation.
- Added “notEquals” function for required values. See documentation.
- Added substitution of markers in user\_copy subject.
- Added the possibility to disable sending mails – useful if you only want to store data. See documentation.
- Some minor refactoring.

## Version 3.0.0

- WARNING: Configuration changed for field “default values”. See documentation for details.
- Made fe\_user markers available for config field “Default values”. WARNING: marker format changed so the marker is prefixed with an underscore (eg. `###_FEUSER_USERNAME###`). See documentation.
- Made it possible to load or store values in database. You really need to read the documentation! The code and concept is partly based on ideas and examples from Martin Kutschker – Thanks Martin.
- Added a preliminary logging facility using `t3lib_div::sysLog()`
- Added support for multiple languages. See documentation.
- Added options to change format of `_DATE` and `_TIME` markers. See documentation.

## Version 2.0.5

- Fixes bug causing the invalid email message to show when email is valid (`###HEADER_EMAIL_INVALID###` marker).

## Version 2.0.4

- checkbox/radio field types are now validated as empty when not set.
- Minor bugfixes/tweaks

## Version 2.0.3

- multi\_recipient is now validated against required values.
- Disabled UTF-8 encoding detection as it seems to break some environments.

## Version 2.0.2

- Fixed typo in code causing “Invalid argument supplied for foreach()...” error.
- Fixed typo in locallang.php file

## Version 2.0.1

- Fixed incorrect “no template defined” error.
- Added utf-8 encoding detection of template file to avoid double encoding.

## Version 2.0.0

- WARNING: The configuration of required fields has changed – See documentation.
- Changed field validation to make it more flexible using regular expressions, error message pr. field etc. See documentation.
- Added possibility to attach files to the mail.
- Added possibility to use TS/overwrite flexform values with TS. See pil\_mailform/ext\_conf\_example.txt and TS reference in documentation.
- Fixed incorrect charset problem.
- Changed the way testmode shows data to mimic mail source.

#### Version 1.1.0

- WARNING: Changed subpart marker “###HEADER\_MAIL\_NOVALID###” to “###HEADER\_MAIL\_INVALID###” - as this is more correct English. Please remember to update your template!
- Most of the changes in this release are suggestions or issues reported by Martin Kutschker, who did an extensive review of TMailform. Thanks to Martin for taking the time to do this extensive review of TMailform and provide a German translation.
- Throw error if radio buttons and/or checkboxes are required and not set.
- Validate [email] field when the field has some value – even if the field is not required.
- Fixed issue with HTML linebreaks in text/plain emails.
- Added a flexform field for setting default values for fields in template.
- Added FE-user markers for eg. setting default [name] to “###FEUSER\_USERNAME###” (Thanks to “pst”).
- Made extra markers available for mail body and subject. See the last part of the “Workflow” chapter of this manual.
- Made “<field>\_VAL” markers available for “static subject” and “user subject prefix” flexform fields.
- Flexform field “Charset” now defaults to “\$GLOBALS['TSFE']->metaCharset”.
- Utf8-encode template if “\$GLOBALS['TSFE']->renderCharset” is utf-8 (Thanks to Vladimir I. Umnov).
- Added Russian translation (Thanks to Vladimir I. Umnov).
- Added German translation.

#### Version 1.0.1

- Specified category in ext\_emconf.php, so that TMailform is listed correctly in EM.

#### Version 1.0.0

- Realizing that 500+ people have downloaded TMailform and only a few bugs have been reported (and fixed), I have come to the conclusion that TMailform is ready to “become” stable. Thus, version is bumped to 1.0.0 – thanks to all that have send bug report and/or feature-request.
- Fixed email validation issue.

#### Version 0.2.2

- Fixed required [text][email] issue.

#### Version 0.2.1

- Fixed Charset bug.

#### Version 0.2.0

- Provided alternatives to PHP mail().
- Provided options for additional mail headers.
- Provide options for additional mail headers (Cc, Bcc, Content-Type, Charset, Content-Transfer-Encoding).
- Made it possible to have more than one static recipient.
- Added a field in configuration to prefix user subject.
- Made submitted values (###<field>\_VAL###) available for TMAIL\_THANKS and TMAIL\_ERROR subparts.
- Update example template to new usage.

#### Version 0.1.3

- Documentation updates

#### Version 0.1.2

- Fixed cache issue.
- Some refactoring. Fixes “wrong parameter count for array\_keys()” issue.

### **Version 0.1.1**

- Removed redundant and unnecessary files from extension.
- Fixed a few typos in documentation.
- Added missing screenshots to documentation.