EXT: Registrations for calendar events

Extension Key: register4cal
Language: en
Keywords: forAdmins
Copyright 2000-2010, Thomas Ernst, <typo3@thernst.de>

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.org

Table of Contents

EXT: Registrations for calendar events.....1
Introduction.....3
What does it do?.....3
Screenshots.....4
Installation.....7
Prerequisites.....7
Installing the extension.....7
Add static templates.....7
Adapt TypoScript settings to your needs.....7
Recommended: Adapt settings for "calendar base" extension.....7
Users manual.....8
Activate the registration for an event.....8
Linking Calender Event Organizers and Frontend Users/Groups.....9
User-defined fields.....10
Field sets.....11
Set storage folder for fields and fieldsets.....11
FAQ.....12
Administration.....13
Enabling registration in event list view.....13
Enabling fields for frontend editing.....14
Telling users that they need to log in.....15
FAQ.....16
Configuration.....17
Reference.....17
Markers for templates.....20
Changing texts with TypoScript.....21
FAQ.....21
Known problems.....22
To-Do list.....23
ChangeLog.....24

Introduction

What does it do?

General overview

The very popular TYPO3 extension "Calendar Base" ("cal") provides a nice calendar, showing all types of events in several possible calendar displays. The extension "register4cal" adds a registration functionality to the calender events. Users can register and unregister for events. A maximum number of attendees can be set for events and a waitlist can be used if this number has been reached.

Userdefined fields for registrations

Several sets of userdefined fields can be defined and displayed in the registration form. This provides a very flexible way to obtain additional data from the attendees. Possible field types include text fields, multiline text areas as well as selection boxes. Fields are fully configurable from the TYPO3 backend and can be arranged in fieldsets. Every event can either use no fields, a configured default fieldset or an individual fieldset. Changing fields or fieldsets will not affect already stored registrations.

Registration in single event view and event list view

This extension supports the registration from the single event view as well as from the event list view. While the single event view shows the registration form below the event display, the event list view has a box attached to each event, including a checkbox to allow the user to register or unregister for the event.

Email confirmation and notification

Every change in a registration can cause a confirmation being sent to the user which registered for the event. Also, a notification can be send to the organizer(s) of the event if a registration has been changed.

Registration display

Besides from showing registration records in the backend, this extension provides a frontend-plugin which can display registrations in two ways:

- An event view, where frontend-users can see the events, they have registered for.
- An participant view, where event organizers can see the users which have registered for their events

Foreign user registration

Event organizers can be enabled to register/unregister foreign users for their events. This enables the organizers to manage all attendees for their events via this extension, even if the registration has been communicated via other ways than the website.

Frontend editing

This extensions fully supports the frontend editing features of the cal extension. It provides modified cal default templates, showing how to include the additional event fields into an additional tab of the cal frontend editing view.

Supporting extensions

The following extensions can be used together with this extension:

- rlmp_dateselectlib: Provides a date selection for the register4cal fields when using calendar event frontend editing.
- onetimeaccount: Enables users to create a temporary frontend user account to register for events.

Screenshots

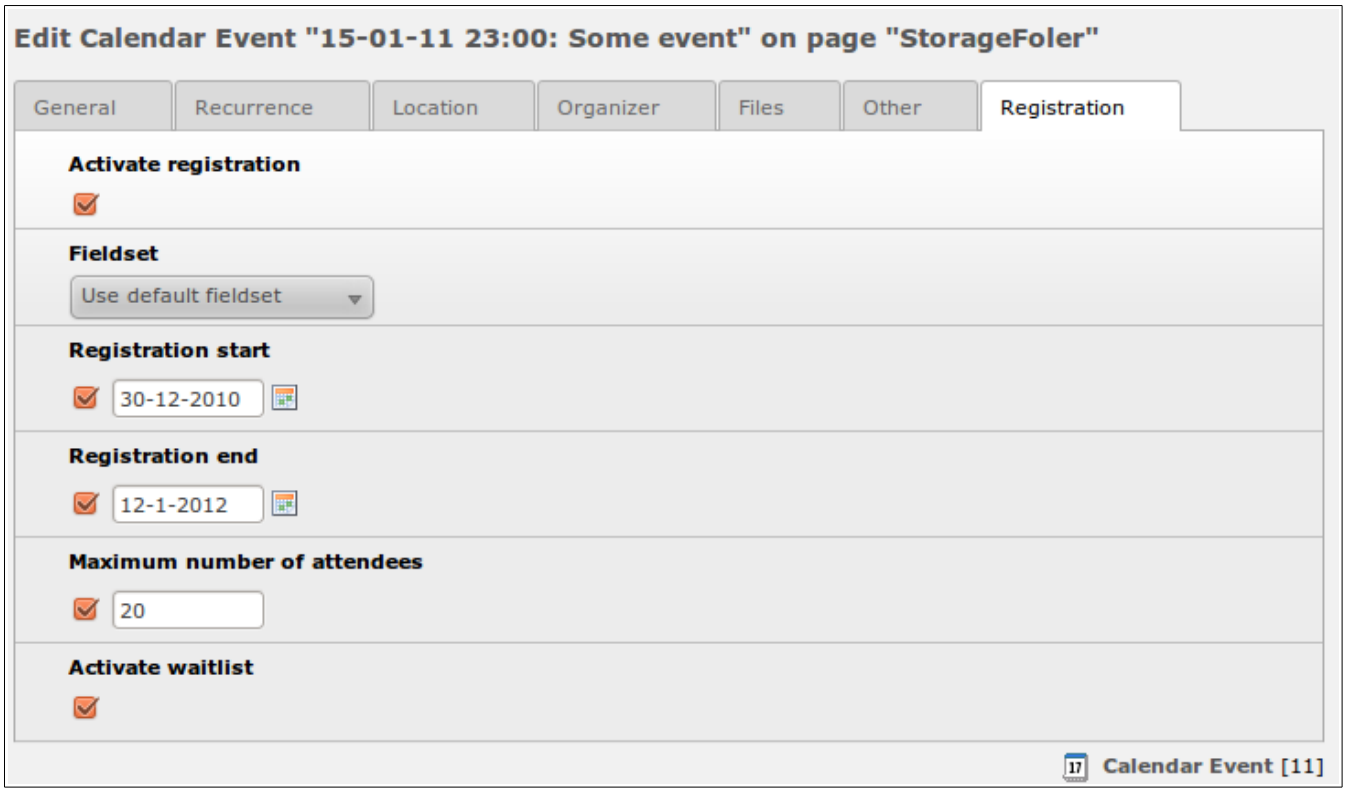

Illustration 1: Activate registration in backend

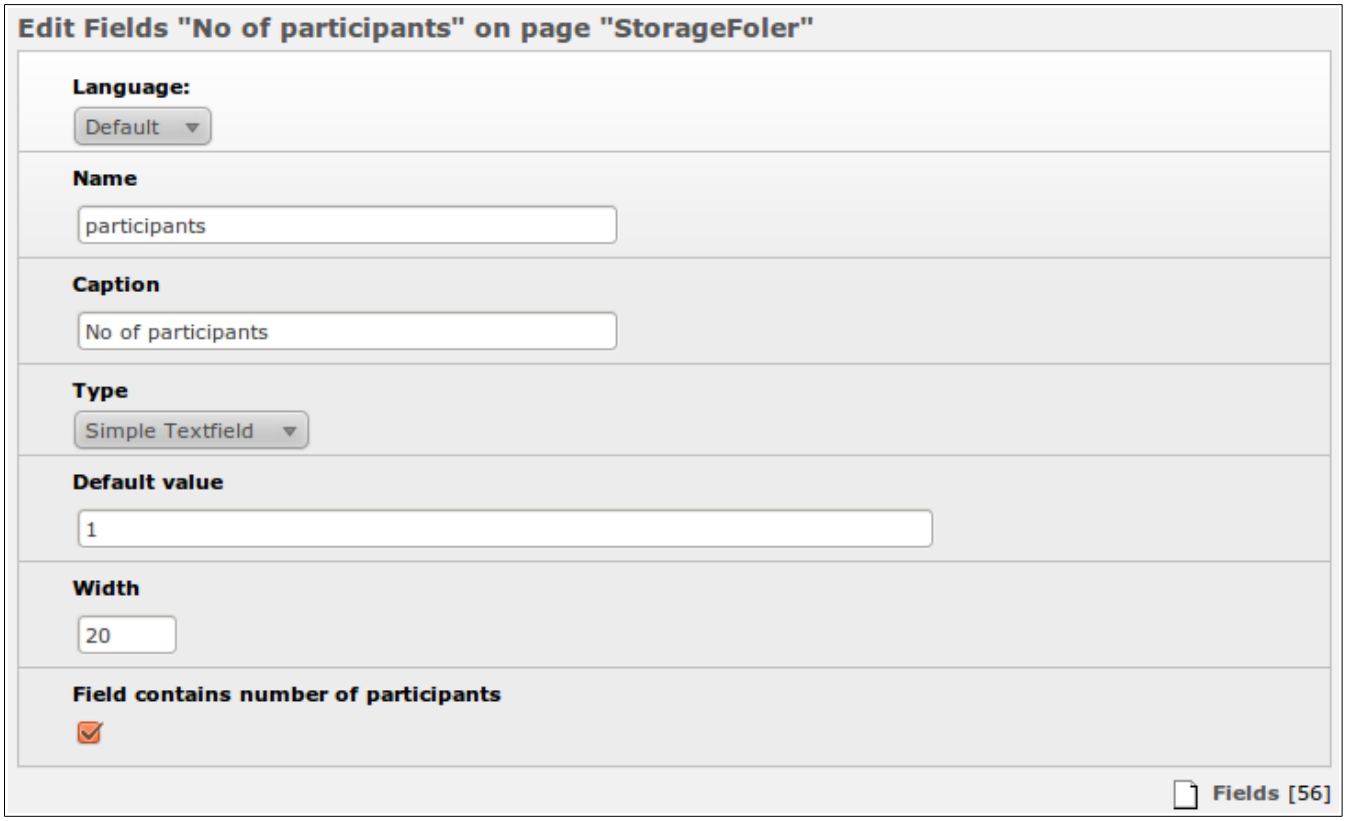

Illustration 2: Settings for a userdefined field

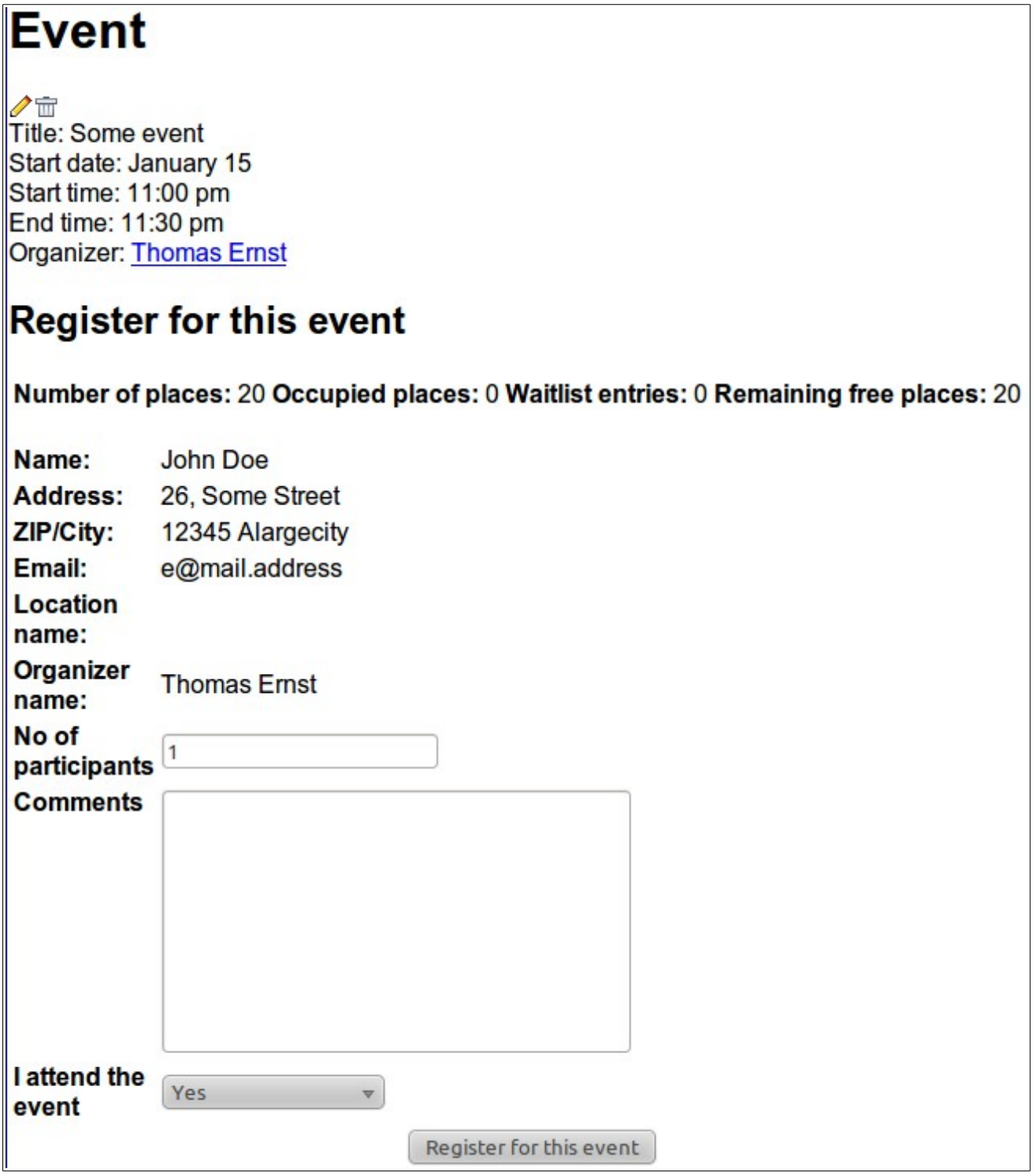

Illustration 3: Single event view with registration form

Title	Start	End	Organizer name	Status of registration
Some event	15.01.2011 23:00	15.01.2011 23:30	Thomas Ernst	Registered

Illustration 4: List of events for which the current frontend user has registered

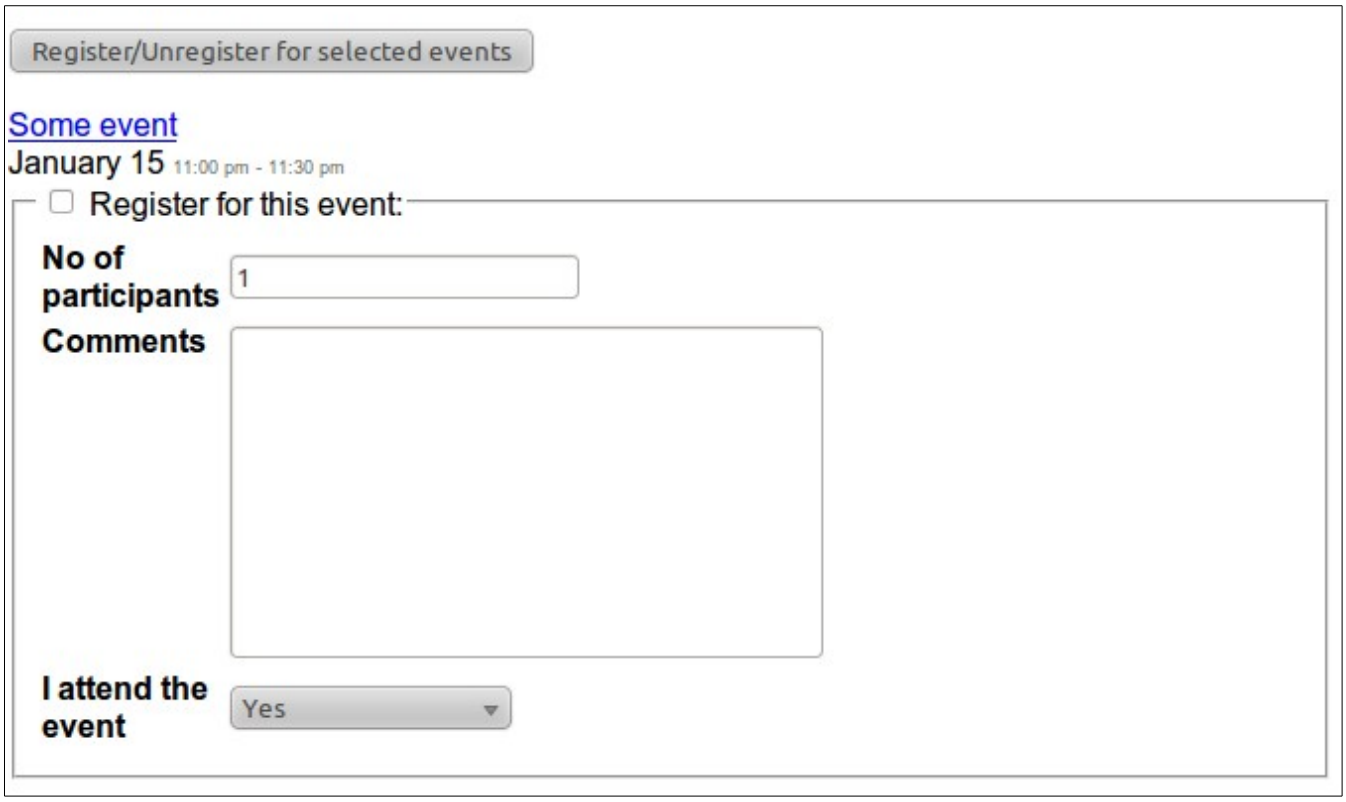

Illustration 5: Event list view with registration box attached to event

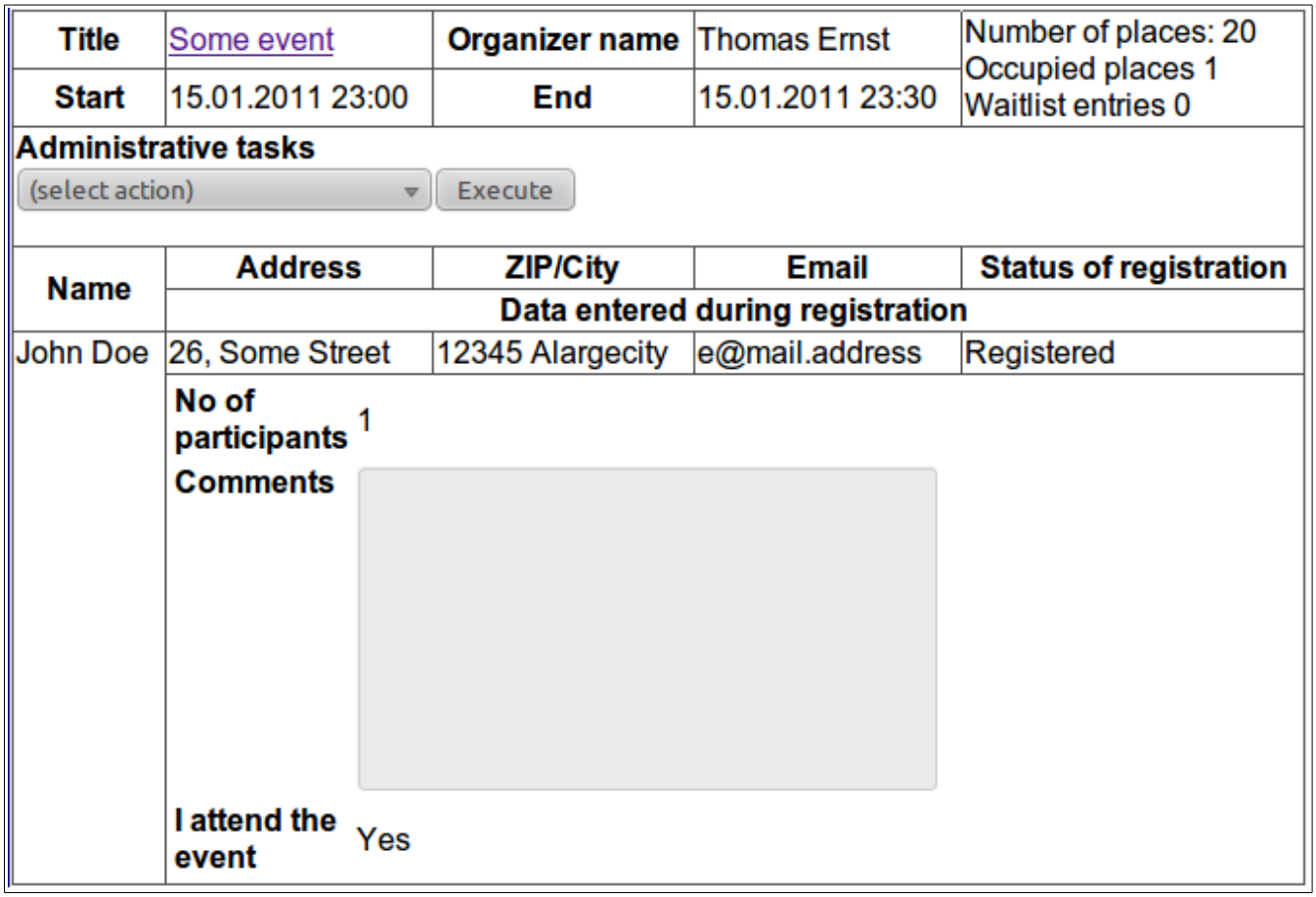

Illustration 6: List of registrations shown to the organizer of an event

Installation

Prerequisites

Before you install this extension, be sure that the following prerequisites are fulfilled:

– You are using at least PHP version 5.1.0 on your server

– You have installed the Calendar Base extension "cal", it is configured properly and working.

Installing the extension

Download the extension from the TYPO3 extension repository and install it through the typical TYPO3 extension installation process using the Extension Manager.

During the installation you will be asked to perform several database changes.

Add static templates

You should now add the required static templates to one of your template records. The following list shows, which templates are provided and when they should be used:

Static Template:	Usage:
Basic settings (register4cal)	General settings for this extension. Should be used, if you are using the current template files from this extension (e.g. for a new installation)
FE-Editing (register4cal)	Settings for frontend-editing. Include this static template, if you are using the frontend editing feature of the Calendar Base extension. The register4cal fields need to be included in the templates, too!
Compatibility to old templates (register4cal)	General settings for this extension, fitting to the old template files used in previous versions of this extension. Use instead of "Basic settings" if you have modified template files from register4cal versions up to 0.6.2. This template should not be used for fresh installations as it might me removed in the future.

Adapt TypoScript settings to your needs

When using the "Basic settings" static template, there is only one setting which needs to be adapted. Add

```
plugin.tx_register4cal_pi1.singleEventPid = [pid]
```

to your TypoScript template and replace [pid] with the page id, that contains the Calendar Base plugin for single event display.

Recommended: Adapt settings for "calendar base" extension

To find out, which frontend users are organizers for events, this extension expands the organizer records of the Calendar Base extension, so that frontend users can be assigned. But the Calendar Base extension also provide the possibility to assign freely typed in organizers for events. For events, which have a freely typed in organizer, this extension can not determine the matching frontend user. Therefore the organizer textfield should not be used for calender events.

The Calendar Base extension provides the possibility to hide the organizer textfield. For this extension it is recommended to activate this option. If you wish to do so, log on to the TYPO3 Backend and go to "Ext. Manager" → "Loaded extension". Click on the Frontend Plugin "Calendar Base". In the "Enable features" section, you will find an option "Hide organizer textfield". Activate it and click on the "Update" button.

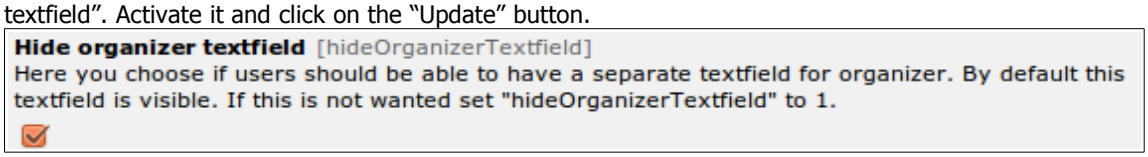

Illustration 7: Calendar Base setting "Hide organizer textfield"

Users manual

Activate the registration for an event

To activate the registration for an event, simply display the event in the TYPO3 Backend and go to the "Registration" tab.

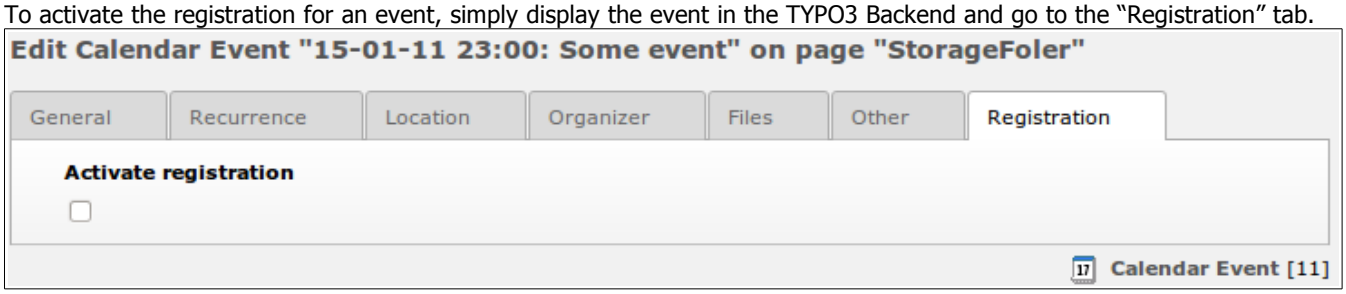

Illustration 8: Activate registration for an event

Registration sepecific fields

After checking the field "Activate registration" TYPO3 asks you to save and refresh the display. Confirm this. The display will be refreshed and show more registration related fields:

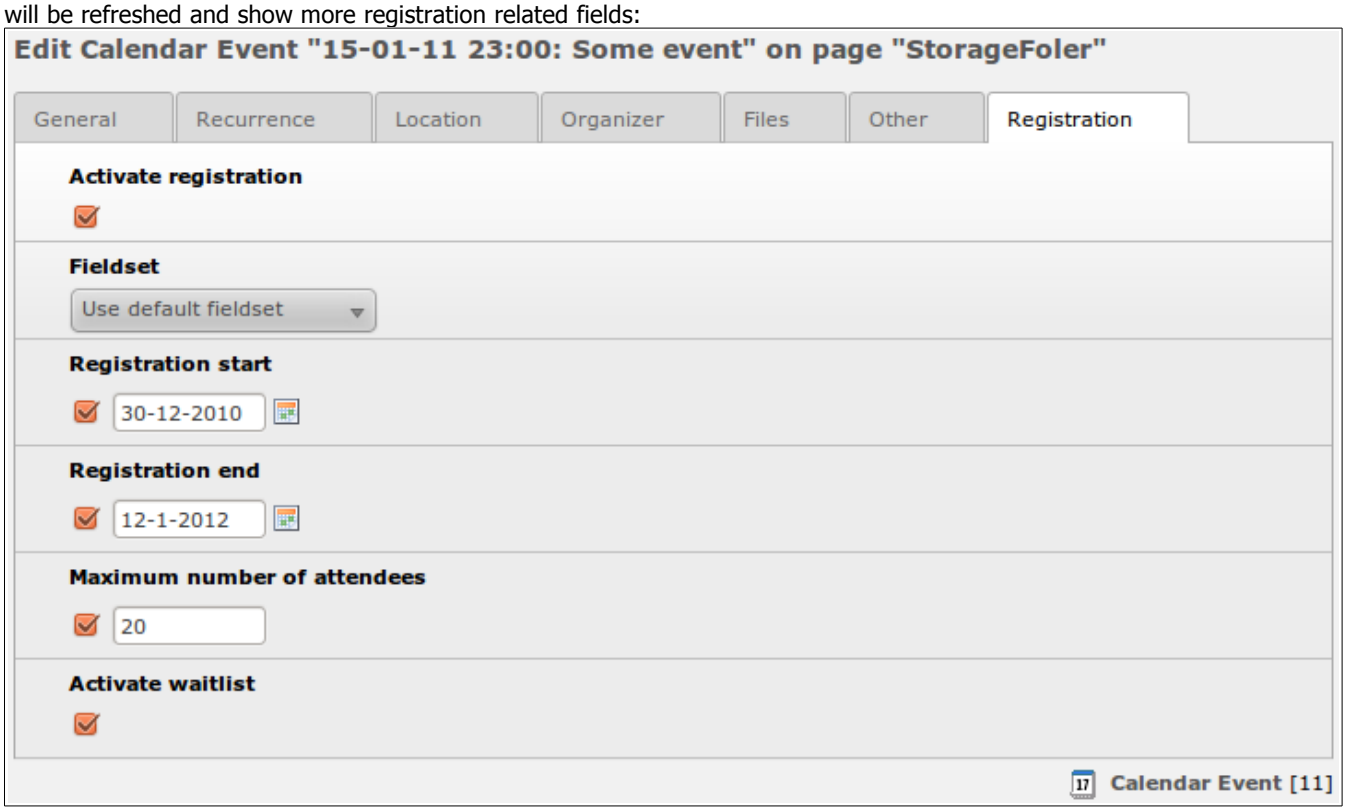

Illustration 9: Registration related fields for an event

Field:	Usage:
Activate registration	Activate or deactivate the registration for this event. For recurring events, registration is activated for all occurrences of the event. If registration is not activated, all other fields are hidden.
Field set	Choose the set of user-defined fields to be shown for the registration for this event. – "Don't use any user fields" will show no user-defined fields – "Use default field set" will use the field set marked as default. If multiple field sets have the "default" option set, the first one found will be used – If you select a defined field set, this field set will be used
Registration start	Date since when registering for this event should be possible. If you leave this field empty, users can register immediately for the event.

Field:	Usage:
Registration end	Date till when registering for this event should be possible. If you leave this field empty, users can register until the start date of the event.
Maximum number of attendees	Maximum number of attendees, which should be able to register for this event. If you leave this field empty, registering is possible for an unlimited number of attendees.
Activate waitlist	If activated, users can enlist to a waitlist, of the maximum number of attendees has been reached. If not activated, no further registrations are possible if the event is fully booked.

Assigning organizers to the event

Besides from the registration specific fields, the organizer setting of the event is important, if the organizer should be able to see registrations for this event in the frontend. On the Organizer tab, there are two possibilities to assign the organizer, if the Organizer Textfield has not been hidden as described in the Installation section. If you assign a Calender Event Organizer record, the fronent users, linked to that record, will be able to see the registrations for this event. Do not use the Organizer Textfield, if you wish to use this functionality.

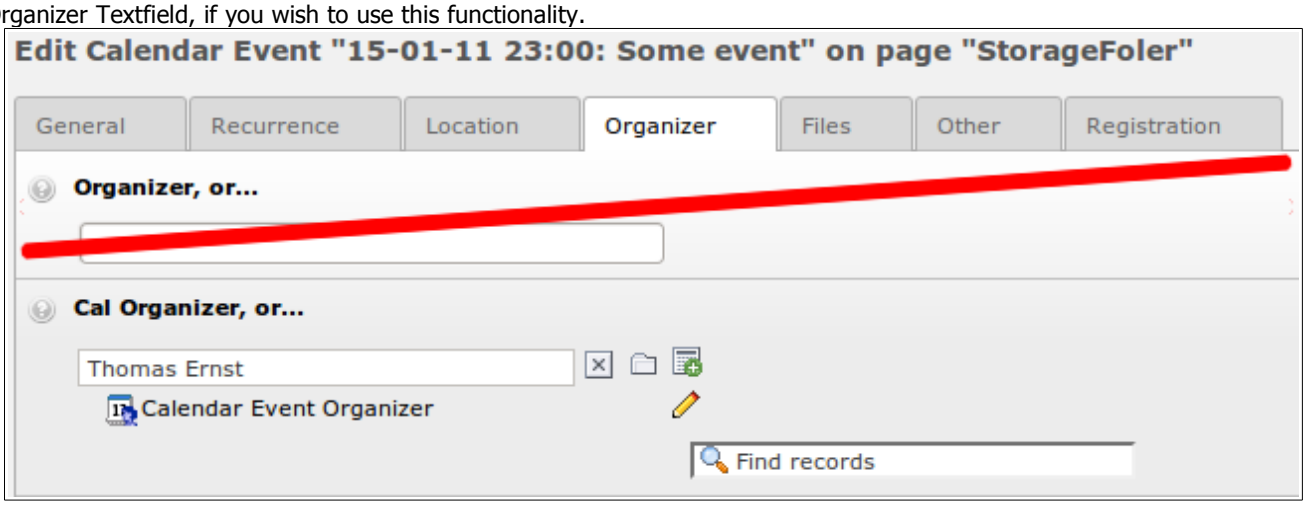

Illustration 10: Assigning an organizer to an event

The Calendar Base extension provides the possibility to hide the Event Organizer Textfield. It is recommended that you do this. Please refer to the description in the Installation section on how to do this.

Linking Calender Event Organizers and Frontend Users/Groups

In order to allow frontend users to see registrations for events for which they are the organizers, it's additionally important to link Calender Event Organizers and Frontend Users. This can be done in the Calender Event Organizer records of the Calendar Base extension. This extension adds an additional field to the record:

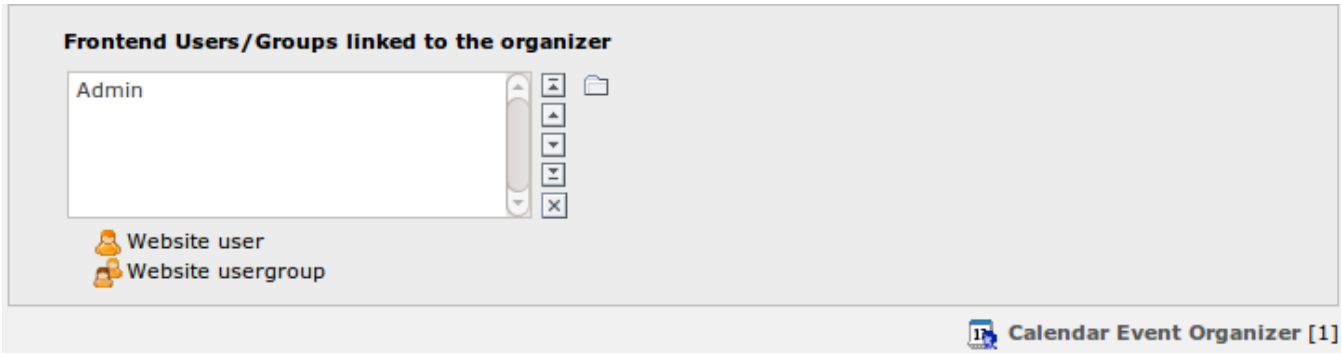

Illustration 11: Link Calendar Event Organizers and Frontend Users/Groups

In this field you can select Website user and Website usergroup records and assign them to the organizer. If such an Calender Event organizer has been asigned to a Calender Event, these users can see who registered for an event and may use additional administrational functions for the event.

User-defined fields

User-defined fields are grouped into field sets. These field sets can be used for registrations.

You can create your own fields by adding "Registration for calender events: Fields" records. User-defined fields need to be stored on the same Page/SysFolder where the events are stored.Alternatively they can be stored in the configured storage folder.

The record for an user-defined field is displayed in the TYPO3 Backend as following:

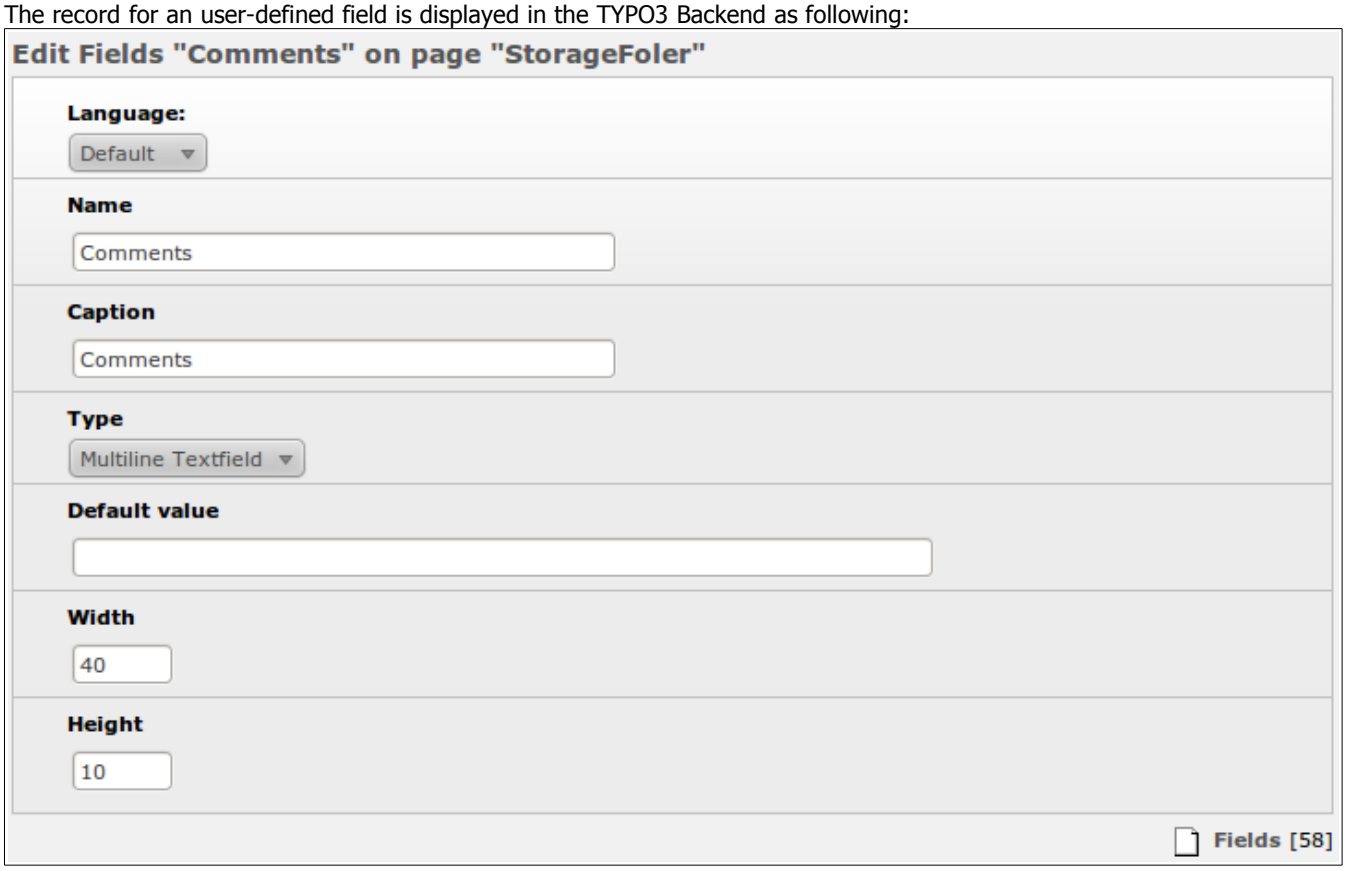

Illustration 12: User-defined field

Field:	Usage:
Language	Language selection. Create a record for language "Default" first. If required, fields can be translated using the normal TYPO3 record translation functionality.
Name	Name of the field. Field names need to be unique.
Caption	Caption of the field. This value is shown in the registration form.
Type	Type of the field. Depending on the type, different input controls are shown in the registration form. Also, different other options will be shown in TYPO3 Backend. Possible types are: – Simple Textfield – Multiline Textfield – Selectionbox
Options (Selectionbox)	Define the options to be shown in the selection box. Separate multiple options by "|"
Default value (all field types)	Default value for the field
Width (all field types)	Width of the field
Height (Mutliline Textfield)	Height for multiline textfield
Height (Selectionbox)	Height for selection box. Set to 1 to display a drop down selection, Set to greater than 1 to display a selection list with the given height.

Field:	Usage:
Field contains number of participants (Simple Textfield)	Indicates, that the field contains the number of participants which should be registered with this registration. If set, the value of this field will be taken as the number of participants for a registration. If no such field is contained in a fieldset for a registration, 1 participant is being counted per registration.

Field sets

User-defined fields are grouped into field sets. These field sets can be used for registrations.

You can create your own field sets by adding "Registration for calendar events: Field sets" records. Field sets need to be stored on the same Page/SysFolder where events and fields are stored. Alternatively they can be stored in the configured storage folder. Field sets can only use fields, which are stored in the same Page/SysFolder.

The record for a field set is displayed in the TYP3 Backend as following:

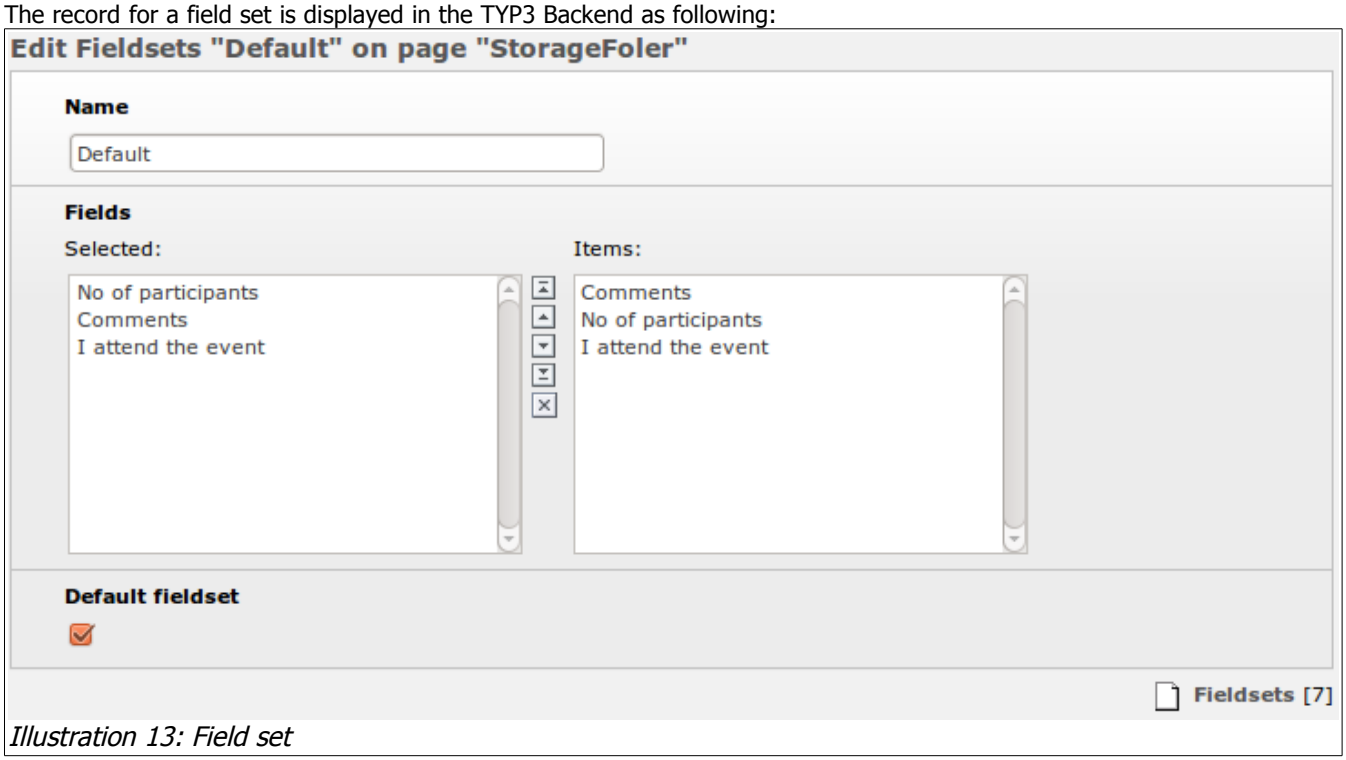

Illustration 13: Field set

Field:	Usage:
Name	Name of the field. Used to select field set in event record.
Fields	Select the user-defined fields to use in the fieldset here
Default field set	If set, this field set is being used as default fieldset. "Use default fieldset" is the default setting for the field set selection in the event record. If multiple field sets with this option set exist, this first one found will be used.

Set storage folder for fields and fieldsets

User-defined fields and fieldsets will first of all be be searched in the folder, containing the event records. In case you have your event records only on a single page/sysfolder, there is nothing else to do than store the field and fieldset records on this page/sysfolder, too.

If you are using multiple pages/sysfolders to store your events, you may want global fields and fieldsets for all events. In this case you can define the page id (pid) of a page/sysfolder from which fields and fieldgroups are used, too. This can be configured via the Extension Manager in the extension configuration of this extension.

After changing this settings it is required to clear the cache so that the changes become effective.

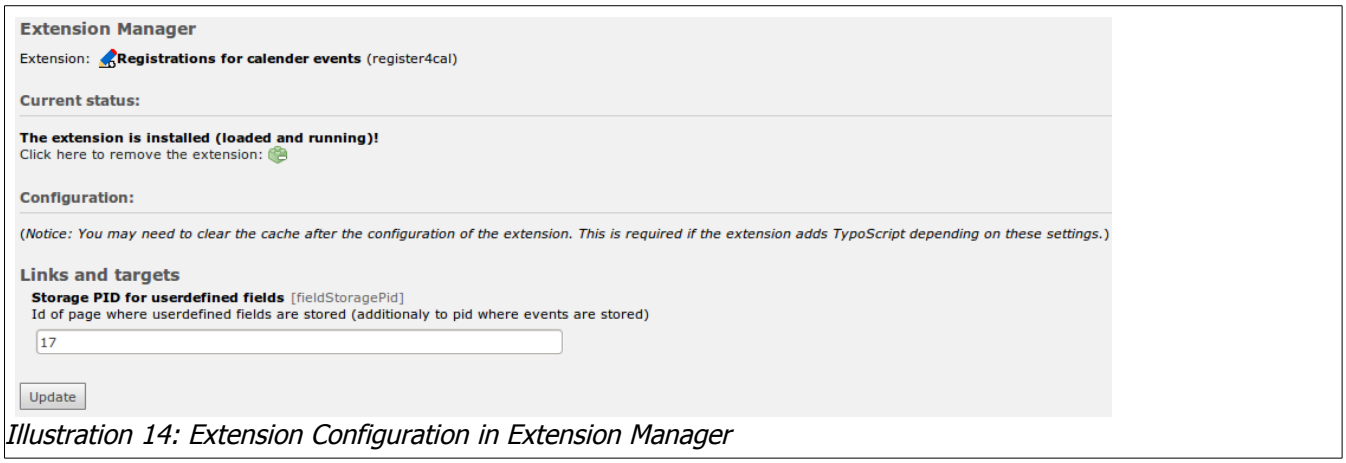

Illustration 14: Extension Configuration in Extension Manager

FAQ

Even though I activated the registration feature, the single event view does not show the registration form.

Check if you are inside the registration period. Check if a frontend user has logged on

The registration form is being displayed well in the single event view, but the event list view does not show the registration possibility.

Read the section on enabling list view registration. Some changes need to be done in the list view templates to activate registration in event list view.

My user-defined field is not being shown in the registration form.

Did you include your user-defined field into a field set? Did you include it in the field set used for the registration for this event?

What happens if I change a field or field set after it has been used for a registration?

Nothing. Existing registrations will not be affected by changes on fields or field sets. For new registrations always the current definition of fields/field sets is being used.

I need an additional field type for my user-defined fields, how can I set it?

Currently it is not possible to define additional field types (except from modifying this extensions coding). If you have a good reason to add another field type, write an email and I'll see what I can do.

Administration

Enabling registration in event list view

To activate the registration feature in the event list view of the Calendar Base Extension, additional markers need to be included in the Calendar Base templates.

Marker for submit button

The marker ###MODULE__tx_register4cal_submit### inserts the submit button for the registration form. It should be added in the list template (default: EXT:cal/template/list/tmpl) in subpart "###LIST_TEMPLATE###". The marker can be added twice to show the button both button on top and below the event list. Please notice the double underscore in the marker name after "MODULE".

Example

```
<!-- ###LIST_TEMPLATE### begin -->

<!-- ###PAGEBROWSER### begin -->
<p class="pagebrowser">
###PAGEOF###  ###PAGES###  ###PREVIOUS### ###NEXT###
</p>
<!-- ###PAGEBROWSER### end -->

<h1>###HEADING###</h1>
###FOUND###
<!-- ###PRE_LIST_TEMPLATE### begin -->
###MODULE__tx_register4cal_submit###
<dl>
<!-- ###PRE_LIST_TEMPLATE### end -->
      <!-- ###LIST### start -->
      <!-- ###LIST### end -->
<!-- ###POST_LIST_TEMPLATE### begin -->
</dl>
###MODULE__tx_register4cal_submit###
<!-- ###POST_LIST_TEMPLATE### end -->

<!-- ###PAGEBROWSER### begin -->
<p class="pagebrowser">
###PAGEOF###  ###PAGES###  ###PREVIOUS### ###NEXT###
</p>
<!-- ###PAGEBROWSER### end -->

###CREATE_EVENT_LINK###
<!-- ###LIST_TEMPLATE### end -->
```

Marker for registration box

The marker "###MODULE_tx_register4cal_listreg" (notice the doule underscore after "MODULE") inserts the registration box for each event. It needs to be added to the event template (default: EXT:cal/template/phpicalendar_event.tmpl, since Calendar Base 1.3.0 EXT:/cal/template/event_model.tmpl). There are several subparts for event display to which the marker should be added:

- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_ODD###
- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_EVEN###
- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_ODD_ALLDAY###
- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_EVEN_ALLDAY###

Example

```
<!-- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_ODD### start -->
      <dt>
            <!-- ###EVENT_LINK### start-->###IMAGE### ###TITLE###<!-- ###EVENT_LINK### end--><br />
            ###STARTDATE### ###ENDDATE### <span class="V9G">###STARTTIME### ###ENDTIME###</span><br />
            ###MODULE__tx_register4cal_listreg###
            <br /><span class="V9G">###CATEGORY_LINK###</span>
      </dt>
      <dd>###DESCRIPTION###</dd>
<!-- ###TEMPLATE_PHPICALENDAR_EVENT_LIST_ODD### end -->
```

Example templates for Calendar Base

This extension contains modified templates for Calendar Base for the event list view. From version 1.4.0 on, cal contains a new set of templates, called "Standard". The previously used templates are still contained and called "Classic". The extension register4cal contains adapted templates for both the standard and the classic templates. These files can be found in EXT:register4cal/templates/cal_standard and EXT:register4cal/templates/cal_classic.

With some lines of TypoScript, you can use the template files from this folder, where the required markers for registration in event list view have been added already.

Typoscript

```
plugin.tx_cal_controller.view {
    list.listTemplate = EXT:register4cal/templates/cal_standard/list.tmpl
    event.eventModelTemplate = EXT:register4cal/templates/cal_standard/event_model.tmpl
}
```

Enabling fields for frontend editing

This extension supports the Calendar Base frontend editing of events. The additional fields related to the registraion feature can be added via markers to the template files of the Calendar Base extension. In addition, some TypoScript settings are required.

Adding static template

All TypoScript settings, related to Calendar Base frontend editing have been put into the static template "FE-Editing (register4cal)". Ensure you have included this static template to your TypoScript template(s), so that the settings will be taken into account.

IMPORTANT: If you have other additional fields defined for Calendar Base frontend editing, you need to combine these settings manually with the settings of this extension.

Template files to modify

There are two templates which need to be modified. Please ensure that you add all markers to both files, as otherwise the fields wont work for frontend editing.

File (Default):	Usage:
EXT:/cal/template/create_event.tmpl	Event creation and modification template
EXT:/cal/template/confirm_event.tmpl	Template for confirmation of event changes

Markers for registration related fields

The following markers can be used on the templates. Please notice the double underscore after "MODULE" and the upper/lower case of the markers

Marker:	Usage:
###MODULE_tx_register4cal_activate###	Field "Activate registration"
###MODULE_tx_register4cal_field###	Field "fieldset"
###MODULE_tx_register4cal_regstart###	Field "Start of registration period"
###MODULE_tx_register4cal_regend###	Field "End of registration period"
###MODULE_tx_register4cal_maxattendees###	Field "Maximum number of attendees"
###MODULE_tx_register4cal_waitlist###	Field "Activate waitlist"

Create an own tab for the registration related fields

To create an own tab for the registration related fields for Calendar Base frontend editing, the fields should not be added to the fields inside an existing tab. Instead a new div section has to be created whose id has to be numbered consecutively to the existing div sections.

Example

```
<div id="sc7" class="tabcontent">
      ###MODULE__tx_register4cal_activate###
      ###MODULE__tx_register4cal_regstart###
      ###MODULE__tx_register4cal_regend###
```

```
      ###MODULE__tx_register4cal_maxattendees###
      ###MODULE__tx_register4cal_waitlist###
      ###MODULE__tx_register4cal_fieldset###
</div>
```

Using the following lines of TypoScript, the caption for the tab is being set:

Example

```
plugin.tx_cal_controller.view {
    create_event.tabbedMenu {
      7 = TEXT
      7.dataWrap = <li><a href="" rel="sc7"><b>Registration</b></a></li>
    }
    edit_event.tabbedMenu {
      7 = TEXT
      7.dataWrap = <li><a href="" rel="sc7"><b>Registration</b></a></li>
    }
}
```

On a multilanguage site you may want to replace the static text "Registration" in the above example with

```
{LLL:EXT:register4cal/locallang_db.xml:tx_cal_event.tx_register4cal_tablabel}
```

This will translate the caption if this extension contains a proper translation for the language used.

Using rlmp_dateselectlib

This extension supports the extension rlmp_dateselectlib for front end editing. With rlmp_dateselectlib a date selector can be added to the "Start/End of registration period" fields. If you want to activate this feature perform the following steps:

- Ensure the rlmp_dateselectlib extension is installed
- Set the TypoScript setting plugin.tx_register4cal.edit.useDateSelector to 1
- Configure the date selector in TypoScript section plugin.tx_register4cal.edit.calConf. The settings in this section are taken from the rlmp_dateselectlib extension, therefore please refer to the manual of rlmp_dateselectlib for further details on these settings

By default, the use of rlmp_dateselectlib is activated in this extension, so that the only task would be to install rlmp_dateselectlib.

Example templates for Calendar Base

This extension contains modified templates for Calendar Base for frontend editing. These files can be found in EXT:register4cal/templates/cal_standard and EXT:register4cal/templates/cal_classic. If you are using a Calendar Base version less than 1.4.0, use cal_classic. For version from 1.4.0 on, use cal_classic if you are using the "Classic" templates from Calendar Base and cal_standard for the "Standard" templates from Calendar Base.

With some lines of TypoScript, you can use the template files from this folder, where the required markers for frontend editing have been added already.

Typoscript

```
plugin.tx_cal_controller.view {
    create_event.template = EXT:register4cal/templates/cal_standard/create_event.tmpl
    edit_event.template = EXT:register4cal/templates/cal_standard/create_event.tmpl
    confirm_event.template = EXT:register4cal/templates/cal_standard/confirm_event.tmpl
    create_event.tabbedMenu {
      7 = TEXT
      7.dataWrap = <li><a href=""
rel="sc7"><b>{LLL:EXT:register4cal/locallang_db.xml:tx_cal_event.tx_register4cal_tablabel}</b></a></li>
    }
    edit_event.tabbedMenu {
      7 = TEXT
      7.dataWrap = <li><a href=""
rel="sc7"><b>{LLL:EXT:register4cal/locallang_db.xml:tx_cal_event.tx_register4cal_tablabel}</b></a></li>
    }
}
```

Telling users that they need to log in

By default no registration forms will be shown if no frontend user has logged in. You may want to change this, so that in this case an information will be shown and links to registration pages will be presented in event single view. It's not possible to use the "need to login" form for the event list view. To enable the "need to login" form, se the following TypoScript parameter:

```
plugin.tx_register4cal_pi1.needLoginForm.disable = 0
```

By default the form now shows two buttons:

- Login using an existing account
- Create an one-time account

Both buttons need some additional customizing.

Login using an existing account

To login with an existing account, usually the TYPO3 system extension felogin is being used. You need to set the following TypoScript parameters, which reside in the "plugin.tx_register4cal_pi1" section:

Parameter:	Usage:
.needLoginForm.loginpid	Page id of page, containing the "Login" plugin from the felogin extension
.needLoginForm.loginreturnparam	URL parameter name, telling the login plugin to which page the user should be forwarded, once he has logged in successfully. For felogin this is "redirect_url".

If these parameters have been set properly, the user can click on the "Login using an existing account" button and will be forwarded to the login form. After sucessfully loggin in, he will be sent back the event single view where he can register.

Create an one-time account

This option requires the extension onetimeaccount to be setup and configured as described in it's manual.

In addition the following TypoScript parameters need to be set, which also reside in the "plugin.tx_register4cal_pi1" section:

Parameter:	Usage:
.needLoginForm.onetimepid	Page id of page, containing the "One-time FE account creator" plugin from the onetimeaccount extension
.needLoginForm.onetimereturnparam	URL parameter name, telling the plugin to which page the user should be forwarded, once he has created an account successfully. For onetimeaccount this is "redirect_url".

If these parameters have been set properly, the user can click on the "Create an one-time account" button and will be forwarded to the account creation form. After sucessfully creating his account, he will be sent back to the event single view where he can register.

Displaying event-related fields

Usually, no event-related fields are contained in the message, telling the user that he needs to log in. It may be required to show such fields (e.g. the number of free places).

In this case you can add the relevant markers (for example ###MAXATTENDEES###, ###NUMATTENDEES###, ###NUMWAITLIST### oder ###NUMFREE###) in template section ###SINGLE_NEEDLOGIN###.

Display of users which have already registered during/after registration

It may be requested that a list of users which have already registered for an event should be displayed in the registration form. This functionality can be activated via TypoScript Parameters in section „plugin.tx_register4cal_pi1.showOtherRegisteredUsersAtRegistration":

Parameter:	Usage:
.enable	Activates the display of other registered users (1=Active, 0=Inactive). Default: Inactive
.onlyAfterRegistration	Configure if the list should only be displayed when an user has registered or if it will be displayed always. (1=Only after registration, 0=always). Default: Only after registration
.includeOwnRegistration	Show the own registration of an user in the list (1=Yes, 0=No). Default: Yes
.includeWaitlist	Show registrations on the waitlist in the list (1=Yes, 0=No). Default: No
.includeCancelled	Show cancelled registrations in the list (1=Yes, 0=No). Default: No

As this function may lead to questions related to data privacy, every user can decide whether his data should be visible for other users or not during registration. For this purpose an additional checkbox will be displayed in the registration form if this functionality is activated. Other users get to see the data only, if the user ticked this checkbox. Differing from this the organizer of the event can see all registered users.

To configure the displayed data the section "OTHER_USERS_LIST" is available in the template, which encloses the complete List. Inside this section "OTHER_USER" encloses a single registration from another user. Inside this sections all markers available in the registration form can be used. It only needs to be considered, that all markers in this sections relate to the registration of the other user.

Though the checkbox to confirm that other users may see the registration is being displayed in the listview registration form, the list of other registrations is not supported there.

Configuration

The whole extension can be configured via TypoScript. All settings can be done in section "plugin.tx_register4cal_pi1". Besides some general settings, these section contains the following subsections:

- foreignUserRegistration: Settings related to the registration of foreign users by event organizers.
- needLoginForm: Settings related to the form which informs about the necessarity to login if no frontend user has logged in.
- email: Settings related to emails to be send from this extension.
- forms: Settings related to forms displayed from this extension

Reference

General settings

Property:	Data type:	Description:	Default:
templateFile	string	Filename of the template file to use	EXT:register4cal/templates/register.tmpl
dateformat	string	Date format to use (pear syntax)	%d.%m.%Y
timeformat	string	Time format to use (pear syntax)	%H:%M
disableWaitlist	boolean	Disable waitlist functionality	0
disableUnregister	boolean	Disable unregistering from events for users	0
useWaitlistIfNotEnoughPlaces	boolean	If an event has a limited number of places and an user tries to register more places than available, you can use this parameter to decide what should happen: 1: Instead of a registration a waitlist entry will be created. The user will be notified of this. 0: The user will receive an error that he can not register this much attendees. This also happens if the parameter is set to 1 and no waitlist is available for the event	1
eventFillMode	integer	Mode to fill up the event. Used in case someone has enlisted to the waitlist, because there are not enough places left for his registration (see parameter useWaitlistIfNotEnoughPlaces) and for updating waitlist entries to registrations if someone unregisters from the event. **1: First come first serve principle** Everyone else can also enlist to the waitlist only. Even if he requests to register less attendees, which would be possible as there are enough free places for the event, he can only create a waitlist entry because someone else came first. If any other user unregisters from the event, only the first waitlist entry will be checked and transferred to a normal registration if there are enough free places. Later entries in the waitlist will not be checked, even if they contain less attendees. **2: Maximum fill principle** Any other user can register for the event if there are enough places left, even if there are already entries on the waitlist. If an user unregisters the whole waitlist will be checked and any entry, having less or equal attendees than number of free places, will be transferred to a registration.	1
keepUnregistered	integer	Keep registrations after unregistering. These registrations will be shown as "cancelled" for the event organizer.	1
adminUsers	integer	id's of frontend users which will be treated as event organizers for every event.	
singleEventPid	integer	page id of page used to show the event single view.	

[tsref:plugin.tx_register4cal_pi1]

Foreign user registrations

Settings related to the registration of foreign users by event organizers can be found in section "plugin.tx_register4cal_pi1.foreignUserRegistration".

Property:	Data type:	Description:	Default:
enable	boolean	Enable or disable the registration of foreign users by event organizers	0
allowOnlyGroups	string	List of groups, which need to be assigned to users that should be selectable as participants for foreign user registration. If set, only users having one of these groups assigned, will be shown for foreign user registration.	
denyGroups	string	List of groups whose users must not be selectable for foreign user registration. If set, users having one of these groups assigned, will not be shown for foreign user registration.	

[tsref:plugin.tx_register4cal_pi1.foreignUserRegistration]

Hint:

allowOnlyGroups overwrites denyGroups. If allowOnlyGroups is set, denyGroups will not be taken into account.

"need to login" form

Settings related to the form, informing visitors that they need to login can be found in section "plugin.tx_register4cal_pi1.needLoginForm".

Property:	Data type:	Description:	Default:
disable	boolean	Enable or disable the display of the "need to login" form. If the form is disabled and no user has logged in, nothing will be displayed	1
loginpid	integer	page id of the page containing the login form for existing users (usually from system extension felogin)	
loginreturnparam	string	Name of parameter used to tell the login form who which page it should return after a successful login	redirect_url
onetimepid	integer	page id of the page containing the login form for onetime accounts (from extension onetimeaccount)	
onetimereturnparam	string	Name of parameter used to tell the onetime-account login form who which page it should return after a successful login	redirect_url

[tsref:plugin.tx_register4cal_pi1.needLoginForm]

Sending emails

Settings related to emails sent from this extension can be found in section "plugin.tx_register4cal_pi1.emails".

Property:	Data type:	Description:	Default:
sendConfirmation	boolean	Enable or disable confirmation emails to be sent to the user registering for an event	0
sendNotification	boolean	Enable or disable notification emails to be sent to the event organizer	0
senderName	string	Plain text name which should be used as sender for the emails	
senderAddress	string	Email address which should be used as sender for the emails	
adminAddress	string	Email address where all registration emails should be send, too (e.g. event supervisor)	

[tsref:plugin.tx_register4cal_pi1.emails]

Attaching vcard files to notification emails

Settings related to the attachments of vcard files containing the participant information to the notification email sent from this extension can be found in section "plugin.tx_register4cal_pi1.notificationVcard".

<table>
<tr><th>Property:</th><th>Data type:</th><th>Description:</th><th>Default:</th></tr>
<tr><td>enable</td><td>boolean</td><td>Enable or disable the attachment of vcard files to notification emails</td><td>1</td></tr>
<tr><td>filename</td><td>string</td><td>Filename for attached vcard file</td><td>Attendee.vcf</td></tr>
<tr><td>fieldmapping.uid
fieldmapping.fullname
fieldmapping.title
fieldmapping.firstname
fieldmapping.lastname
fieldmapping.attitionalnames
fieldmapping.nameprefix
fieldmapping.namesuffix
fieldmapping.nickname
fieldmapping.birthday
fieldmapping.organization
fieldmapping.department
fieldmapping.subdepartment
fieldmapping.role
fieldmapping.revision
fieldmapping.geolocation
fieldmapping.mailer
fieldmapping.timezone</td><td>string</td><td>Simple assignment of table fields to vcard fields.
For each field, you can assign a table field by using the following markers:
<table><tr><th>Table:</th><th>Marker for data:</th></tr><tr><td>Events</td><td>EVENT_[fieldname]</td></tr><tr><td>Event Location</td><td>LOCATION_[fieldname]</td></tr><tr><td>Event Organizer</td><td>ORGANIZER_[fieldname]</td></tr><tr><td>Frontend-User</td><td>USER_[fieldname]</td></tr><tr><td>User-defined fields</td><td>UDEF_[fieldname]</td></tr></table></td><td>uid = USER_uid
fullname = USER_name
title = USER_title
firstname = USER_first_name
lastname = USER_last_name
additionalnames = USER_middle_name
nameprefix =
namesuffix =
nickname = USER_username
birthday =
organization = USER_company
department =
subdepartment =
role =
revision =
geolocation =
mailer =
timezone =</td></tr>
<tr><td>fieldmapping.phonenumbers
fieldmapping.emails
fieldmappung.urls</td><td>array</td><td>Assignment of table fields to vcard fields.
These entries can appear multiple times in the vcard. Also they can have several “type”-classifications. Therefore you can have multiple subsections for these properties. The value of each subsection is the phonenumber/email/url. Use the markers described above to assign table fields. The properties contained in the subsection are flags for the available “type”-classifications.</td><td>phonenumbers {
0 = USER_telephone
0 {
PREF = 0
WORK = 0
HOME = 1
OTHER = 0
VOICE = 1
FAX = 0
MSG = 0
CELL = 0
CAR = 0
}
1 = USER_fax
1 {
PREF = 0
WORK = 0
HOME = 1
OTHER = 0
VOICE = 0
FAX = 1
MSG = 0
CELL = 0
CAR = 0
}
}</td></tr>
<tr><td>fieldmapping.addresses</td><td>array</td><td>Assignment of table fields to vcard fields.
Adresses can appear multiple times, too. They also can have “type”-classifications, but have a couple of fields per address. Use the markers described above to assign table fields. The properties contained in subsection “type” are flags for the available “type”-classifications.</td><td>addresses {
0 {
postofficeaddress =
extendedaddress =
street = USER_address
city = USER_city
zip = USER_zip
country = USER_country
type {
DOM = 0
INTL = 0
POSTAL = 0
PARCEL = 0
HOME = 1
WORK = 0
}
}
}</td></tr>
</table>

[tsref:plugin.tx_register4cal_pi1.notificationVcard]

Configuring output

All views can be configured via TypoScript in the "plugin.tx_register4cal_pi1.forms" section. There is a subsection for every form, allowing the configuration of this form.

Subsection:	Form:
single.default{}	Default settings, used from all "single." subsections
single.noregister{}	Single event view: Notice that the event is fully booked
single.needLogin{}	Single event view: Notice that the user needs to register
single.registration.enter{}	Single event view: Registration form for event
single.registration.alreadyDone{}	Single event view: Notice and data if user has already registered for an event
single.waitlist.enter{}	Single event view: Waitlist entry form for event
single.waitlist.alreadyDone{}	Single event view: Notice and data if user has already enlisted on waitlist for an event
list.default{}	Default settings, used from all "list." subsections
list.submit{}	Event list view: Section with submit button
list.registration.enter{}	Event list view: Registration form for event
list.registration.alreadyDone{}	Event list view: Notice and data if user has already registered for an event
list.waitlist.enter{}	Event list view: Waitlist entry form for event
list.waitlist.alreadyDone{}	Event list view: Notice and data if user has already enlisted on waitlist for an event
email.default{}	Default settings, used from all "email." subsections
email.registration.enter.confirmation{}	Email: Confirmation after registration
email.registration.enter.notification{}	Email: Notification after registration
email.registration.enter.confirmation_rfu{}	Email: Confirmation after registration by event organizer (register foreign user functionality)
email.registration.enter.notification_rfu{}	Email: Notification after registration by event organizer (register foreign user functionality)
email.registration.cancel.confirmation{}	Email: Confirmation after canceled registration
email.registration.cancel.notification{}	Email: Notification after canceled registration
email.registration.cancel.confirmation_rfu{}	Email: Confirmation after registration canceled by event organizer (register foreign user functionality)
email.registration.cancel.notification_rfu{}	Email: Notification after registration canceled by event organizer (register foreign user functionality)
email.waitlist.enter.confirmation{}	Email: Confirmation after enlisting to waitlist
email.waitlist.enter.notification{}	Email: Notification after enlisting to waitlist
email.waitlist.enter.confirmation_rfu{}	Email: Confirmation after enlisting to waitlist by event organizer (register foreign user functionality)
email.waitlist.enter.notification_rfu{}	Email: Notification after enlisting to waitlist by event organizer (register foreign user functionality)
email.waitlist.cancel.confirmation{}	Email: Confirmation after canceled enlisting to waitlist
email.waitlist.cancel.notification{}	Email: Notification after canceled enlisting to waitlist
email.waitlist.cancel.confirmation_rfu{}	Email: Confirmation after enlisting to waitlist canceled by event organizer (register foreign user functionality)
email.waitlist.cancel.notification_rfu{}	Email: Notification after enlisting to waitlist canceled by event organizer (register foreign user functionality)
email.waitlist.upgrade.confirmation{}	Email: Confirmation after waitlist entry upgraded to registration

Subsection:	Form:
email.waitlist.upgrade.notification{}	Email: Notification after waitlist entry upgraded to registration
listOutput.events{}	List of events for which an frontend user registered
listOutput.attendees{}	List of attendees for the event organizers
listOutput.registerForeignUser{}	Registering foreign users for the event organizer

In this sections the following properties are possible:

Property:	Data type:	Description:	Can be used in:
subtemplate	string	Name of the subpart of the template file used to render the form	all forms
messagetype_i	wrap	Possibility to wrap a single message of type "information"	single event and event list registration forms
messagetype_e	wrap	Possibility to wrap a single message of type "error"	single event and event list registration forms
messages	wrap	Possibility to wrap all messages	single event and event list registration forms
userfield	wrap	Possibility to add a wrap to a single user-defined field. You can use "###CAPTION###" to insert the caption if the field	all forms
fields	wrap	Possibility to wrap all user-defined fields	all forms
eventcheckbox	wrap	Possibility to wrap the event selection checkbox	event list registration forms
onetimeaccountlink	wrap	Possibility to wrap the link for onetime-account creation	single event registration forms
loginlink	wrap	Possibility to wrap the link for normal login	single event registration forms
submitbutton	wrap	Possibility to wrap the submit button	single event and event list registration forms
userselection	wrap	Possibility to wrap the user selection	register foreign user form
foreignuserregistration	wrap	Possibility to wrap the single event registration form if used for registering a foreign user	register foreign user form
adminpanel	wrap	Possibility to wrap the admin-panel functions	list of attendees
subject	wrap	Subject line for emails	emails
[any other property]	wrap	Possibility to wrap a marker in the template. Marker names will be converted into lower case and if a matching property exists, if will be used for wrapping.	all forms

Markers for templates

Data from event, location, organizer and user tables

All fields from tables containing event, location, organizer and user data can be used via markers.

Table:	Marker for data:	Marker for label
Events	EVENT_[fieldname]	LABEL_EVENT_[fieldname]
Event Location	LOCATION_[fieldname]	LABEL_LOCATION_[fieldname]
Event Organizer	ORGANIZER_[fieldname]	LABEL_ORGANIZER_[fieldname]
Frontend-User	USER_[fieldname]	LABEL_USER_[fieldname]
User-defined fields	UDEF_[fieldname]	LABEL_UDEF_[fieldname]

When using user-defined fields please notice, that they can only be used if they are contained in the fieldgroup assigned to the event.

Additionaly, the following markers can be used:

Function:	Marker for data:	Marker for label:
Link to event single view	EVENT_link	

Function:	Marker for data:	Marker for label:
Start of event (formated)	EVENT_formated_start	LABEL_EVENT_formated_start
End of event (formated)	EVENT_formated_end	LABEL_EVENT_formated_end
Name of event organizer (independent if he has been entered manually or assigned via an organizer record)	EVENT_organizer_name	LABEL_EVENT_organizer_name
Email of event organizer (independent if he has been entered manually or assigned via an organizer record)	EVENT_organizer_email	LABEL_EVENT_organizer_email
Registration in list view: Checkbox vor selecting events	EVENT_CHECKBOX	
Link to current page	LINK	
Link to page with login form	LOGINLINK	LABEL_login
Link to page with form to create an onetime-account	ONETIMEACCOUNTLINK	LABEL_onetimeaccount
Number of places for an event	MAXATTENDEES	LABEL_maxattendees
Number of currently booked places for an event.	NUMATTENDEES	LABEL_numattendees
Number of waitlist entries for an event	NUMWAITLIST	LABEL_numwaitlist
Number of free places for an event	NUMFREE	LABEL_numfree
Section with user-defined fields	FIELDS	
Submitbutton	SUBMITBUTTON	LABEL_submitbutton
Section with messages	MESSAGES	
Status of registration	STATUS	LABEL_status
Registration in List view: Label for register/unregister box	TITLE	
Unregistration on list view: Message for checkbox	UNREGISTERME	
Admin-Panel for list of registrations for an event	ADMINPANEL	LABEL_admin_title
Foreign user registration: Select field for users	USERSELECTION	
Foreign user registration: Registration form	FOREIGNUSERREGISTRATION	

Any other markers and labels

If for an marker "LABEL_*" no definition can be found, the text for object "label_*" from the language file EXT:register4cal/view/locallang.xml will be used.

If for an marker * no definition could be found, it is checked if a property with the same name (but lowercase) exists in the TypoScript section for the form. If this is the case, this property will be used as wrap for the marker.

Changing texts with TypoScript

If it is required to change texts from the language file EXT:register4cal/view/locallang.xml, you should not simply change this file. Changes would be lost if a new version of this extension is installed.

Instead, TYPO3 offers an easy way to change texts via TypoScript. There are two different types of texts which require different additions in TypoScript to change then.

Texts used from TypoScript

Mosts texts in the forms are directly set in TypoScript using the following form of TypoScript coding:

```
plugin.tx_register4cal_pi1.[something].dataWrap = {LLL:EXT:register4cal/view/locallang.xml:[text_name]}
```

To change such texts, just replace the default value from the TypoScript template your new text:

```
plugin.tx_register4cal_pi1.[something].dataWrap = [new text]
```

If you are familiar with Typo3 language files, and need to change a lot of texts, you may also create your own language file and use it instead of the default file. Using this method eases the changes for multilanguage sites. Conditions, changing texts for each language differently are not required. Only one value needs to be changed in TypoScript:

```
plugin.tx_register4cal_pi1.[something].dataWrap = {LLL:path/to/your/locallang.xml:[text_name]}
```

Texts used from Coding

Other texts, which are used from the coding, can be changed by adding substitution texts in TypoScript setup. To change a text, add the following line to your TypoScript coding:

```
plugin.tx_register4cal_pi1._LOCAL_LANG.[language].[text_name] = [new text]
```

Where [language] is the language, in which the text should be changed (e. g. „default" for english, „de" for german). [text_name] is the name of the text to be changed and [new text] is the new text.

Known problems

Currently, no major problems are known.

For other issues, requested features and so on, please see the project page at forge.typo3.org. If you encounter problems, please do not hesitate to create a bug message there.

To-Do list

- A backend module should be included which allows the display/management of registrations
- A task for the TYPO3 scheduler should enable the deletion of registrations for past events
- Created HTML coding should include css classes/ids to ease formatting of elements in frontend display
- Several hooks could be implemented to enable the use of other types of userdefined fields
- Localize error messages

If you require further features, please do not hesitate to create a feature request on the project page at forge.typo3.org.

ChangeLog

The ChangeLog can be found online in the extension repository via the following link.