

Extension Netcos Renting

by Bernd Warken at Netcos AG

0. Introduction	3
<i>0.1 Preface</i>	3
<i>0.2 License</i>	3
1. Kickstarter	
<i>1.1 Creation of the Extension</i>	4
<i>1.2 New Database Table</i>	4
<i>1.3 Frontend Plugins</i>	4
<i>1.4 Saving und finishing</i>	5
2. Object methods for Termin	6
3. Additions	15
<i>3.1 smarty</i>	15
<i>3.2 jscalender</i>	15
<i>3.2 Multilingualism</i>	16
4. Suit the PHP source code	22
<i>4.1 Database Access</i>	22
<i>4.2 First Plugin (Renting Administration)</i>	22
<i>4.3 Second Plugin (rented)</i>	31
5. HTML-Templates	38
<i>5.1 First Plugin (Renting Administration)</i>	38
<i>5.2 Second Plugin (rented)</i>	42

0. Introduction

0.1 Preface

This document describes the background for the generation of a TYPO3 extension with access to the database. This is an official extension, it is available on typo3.org.

On a first step, the extension is created using the *kickstarter*. Then it is changed manually. As an example, the renting of an apartment is treated. Two frontend plugins are generated, one for the administration of the renting, the other for viewing the bookings for all users.

The dates are stored in the class *Termin* – a German word meaning fixed date - using the format *yyyymmdd*. This approach simplifies the comparison of dates.

For a better representation of the calendar form during the input, the program *jscalender* is used. To make the source code files smaller, the TYPO3 extension *smarty* is used, which allows to position HTML code in external files.

The extension contains multilingualism, the concepts for dealing with several languages. Up to the moment, the 2 languages English and German are used. In the extension source code, the terms in several languages are stored in the files *locallang*.xml*.

The extension was built on Linux, so this document describes all path names in the way used by Unix-like systems. This also applies to the Mac. But if you want to work on Microsoft Windows you need to adjust all path names accordingly. No other changes are necessary.

Note that the extension created here was originally generated in German. In the source code, there are still many elements that reflect the German language.

0.2 License

This document is part of the TYPO3 project on the **Extension netcos_renting**. The actual version of this document is **1.0** of 3 April 2008. The author is **Bernd Warken** at [netcos AG](http://netcos.de).

All elements of the project (this document and the source code) are set under the license **GPL (GNU General Public License) version 3**. The original English text of this license is available at <http://www.gnu.org/licenses/gpl.html>.

Copyright 2008 Bernd Warken

1. Kickstarter

Kickstarter is a TYPO3 extension for generating of own extensions. First the *kickstarter* must be installed using the *Extension Manager*.

1.1 Creation of the Extension

When using *kickstarter*, a new field *Make new extension* arises in the menu of the *Extension Manager*. We choose this field.

In the arising form, you first provide a name in the field *Enter extension key:*, in this case **netcos_renting**. Admit by using the button *Update...*

Next the option list *KICKSTARTER WIZARD* is handled in the form.

First push the plus sign of *General info*. Therein create the *Title* **Netcos Renting**. As *Category*, choose *Frontend Plugins*. Let *State* at *Alpha*. For *Dependencies* provide **cms** and **smarty**. Add *author*. Push the button *Update...*

Next add an additional language. Choose the plus sign of *Setup languages* and therein the language *German*. Affirm by *Update...* The default language of an extension is always English, the chosen German language is additional.

1.2 New Database Table

We will store the bookings in a new database table. For that, the plus sign of *New Database Tables* is pushed. As *Tablename* we provide: **fromto**. With this name, a new table is created within the database of the TYPO3 installation. The complete table name is **tx_netcosrenting_termine**. As *Title of the table* we take **rented** and on German **vermietet**.

Now the fields that are used in the table are provided. A booking element consists of two dates: **from_date** and **to_date**.

In *NEW FIELD*, we hence take **from** as *Field name*, with *Field title* **From** in English and **Von** in German. The *Field type* is **String input**. *Field width* and *Max. characters* are 8 each. *Update...*

In the newly arising *NEW FIELD* we then add **to_date** as *Field name*, with *Field title* **To** in English and **Bis** in German. The *Field type* is again **String input**. *Field width* and *Max. characters* are again 8 each. *Update...*

1.3 Frontend Plugins

Now push the plus sign of *Frontend Plugins* twice. Thereby two new fields for the generation of plugins are generated, one for the renting administration and the other for the viewing of bookings.

In the first of these plugin forms, the English *title* **Renting Administration** is written, in German it is **Vermietungsverwaltung**. Choose the button *Add to 'Insert Plugin' list in Content Elements* and therein *Add icon to 'New Content Element' wizard*. *Update*.

In the other plugin form, the English *title* is **rented** angegeben und in Deutsch **vermietet**. Anklicken *Add to 'Insert Plugin' list in Content Elements* und darin *Add icon to 'New Content Element' wizard*. *Update*.

1.4 Storing and finishing

The input in the *kickstarter* is now done. So choose the field button *View result*. There push *WRITE*. The Extension is then stored to *typo3conf/ext/netcos_renting/*. This extension is now shown in the menu field *Install extensions* of the *Extension Manager* and can be install by pushing the plus symbol.

You can view and edit the source code files of the extension by pushing the extension name in the *Extension Manager*. In the resulting menu, the field *Edit files* must be chosen. This displays the directory structure of the extension source code. As two plugins were installed, the extension source code contains also the two subdirectories *pi1* and *pi2* with the source code for the plugins.

2. Object Methods for the internal Date Format

The date elements for the booking are stored in the database in the format *yyyymmdd*. For example, the date for 23 January 2008 is internally stored as *20080123*. During the construction of the extension, we will calculate with this format in PHP.

To handle the different date formats, it is useful to define a PHP class for doing this. We call this class **Termin**, a German word meaning fixed date. The code for this class is stored in the file *class_termin.php* in the extension main directory.

As this file does not have the access to several languages all error messages are written in English. For the normal usage, this is enough. The still existing German strings, such as **\$Monatsnamen**, are no longer used, but we just keep them in the file for compatibility.

As this file is very long we decided to present it in several parts. If you want to have the whole file you must combine all of these parts or just look at the extension source code.

Start and license:

```
<?php
/*****
* Copyright notice
*
* (c) 2007,2008 Bernd Warken <be.warken@netcos.de>
* All rights reserved
*
* This script is part of the TYPO3 project. The TYPO3 project is
* free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* The GNU General Public License can be found at
* http://www.gnu.org/copyleft
*
* This script is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* This copyright notice MUST APPEAR in all copies of the script!
*****/

class Termin {
```

Private variables:

```
    private $value;
    private $max;
    private $error = "";
    private $timestamp = 0;
    static private $max_Tage = array(0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
    static private $Monatsnamen = array(0, 'Januar', 'Februar', 'März', 'April', 'Mai', 'Juni',
        'Juli', 'August', 'September', 'Oktober', 'November', 'Dezember');
    static private $Monatsnummern = array(
```

```

'Januar' => 1, 'januar' => 1, 'Jan' => 1, 'Jan' => 1, 'JAN' => 1,
'Februar' => 2, 'februar' => 2, 'Feb' => 2, 'feb' => 2, 'FEB' => 2,
'März' => 3, 'märz' => 3, 'Maerz' => 3, 'maerz' => 3,
'Mär' => 3, 'mär' => 3, 'MÄR' => 3, 'Mae' => 3, 'mae' => 3, 'MAE' => 3,
'April' => 4, 'april' => 4, 'Apr' => 4, 'APR' => 4, 'apr' => 4,
'Mai' => 5, 'mai' => 5,
'Juni' => 6, 'juni' => 6, 'Jun' => 6, 'jun' => 6, 'JUN' => 6,
'Juli' => 7, 'juli' => 7, 'jul' => 7, 'Jul' => 7, 'JUL' => 7,
'August' => 8, 'august' => 8, 'Aug' => 8, 'aug' => 8, 'AUG' => 8,
'September' => 9, 'september' => 9, 'Sep' => 9, 'sep' => 9, 'SEP' => 9,
'Oktober' => 10, 'oktober' => 10, 'Okt' => 10, 'okt' => 10, 'OKT' => 10,
'November' => 11, 'november' => 11, 'Nov' => 11, 'nov' => 11, 'NOV' =>
11,
'Dezember' => 12, 'dezember' => 12, 'Dez' => 12, 'dez' => 12, 'DEZ' => 12
);
static private $char2int = array('0' => 0, '1' => 1, '2' => 2, '3' => 3, '4' => 4, '5' => 5,
'6' => 6, '7' => 7, '8' => 8, '9' => 9);

```

Construction of a Termin objects from a date string:

```

public function __construct($value) {
    $str = trim((string)$value);
    $this->value = "";
    if ($str == "") {
        $this->append_error('empty Termin data. ');
        return(FALSE);
    }
    if ($this->_is_Termin($str)) {
        $this->value = $str;
        $m = $this->Monat();
        if ($m <= 0 || $m > 12) {
            $this->append_error('wrong month number: '.$m.'. ');
            $this->value = "";
            return(FALSE);
        }
        $t = $this->Tag();
        $max = $this->max_Tage_des_Monats();
        if ($t <= 0 || $t > $max) {
            $this->append_error('wrong number of days: '.$t.', maximal number of days of the month is '.
            $max.'. ');
            $this->value = "";
            return(FALSE);
        }
        $this->max = $max;
        return(TRUE);
    }
    $len = strlen($str);
    $a = explode("/", $str);
    $c = count($a);
    $english = 'yes';
    switch($c) {
        case 3:

```

```

$j = $a[0];
$m = $a[1];
$t = $a[2];
break;
case 2:
    $j = date('Y');
    $m = $a[0];
    $t = $a[1];
    break;
default:
    $english = 'no';
    break;
}
if ($english == 'yes') {
    $s = "";
    if ($this->_is_int($j)) {
        $j = (int)$j;
        if ($j < 100) $j += 2000;
        if ($j < 2000 || $j >= 3000) {
            $this->append_error('Wrong year number: '.$j.'.');
            return(FALSE);
        }
        $s = (string)$j;
    }
    if (strlen($s) != 4) {
        $this->append_error('wrong year description: '.$j.'.');
        return(FALSE);
    }
    if ($this->_is_int($m)) {
        $m = (int)$m;
        if ($m > 0 && $m < 10) {
            $s .= '0'.(string)$m;
        } elseif ($m <= 12) {
            $s .= (string)$m;
        } else {
            $this->append_error('wrong month number: '.$m.'.');
            return(FALSE);
        }
    } else {
        $this->append_error('month must be a number: '.$m.'.');
        return(FALSE);
    }
    if ($this->_is_int($t)) {
        $t = (int)$t;
        $max = $this->max_Tage_des_Monats($m, $j);
        if ($t <= 0 || $t > $max) {
            $this->append_error('wrong number of days: '.$t.', maximal number of days of the month is
'. $max.'.');
            return(FALSE);
        }
        if ($t > 0 && $t < 10) {
            $s .= '0'.(string)$t;

```

```

    } else {
        $s .= (string)$t;
    }
} else {
    $this->append_error('day must be a number: '.$t.');
    return(FALSE);
}
$this->value = $s;
return(TRUE);
}

$i = 0;
$s = "";
while ( $this->_is_int( $str{$i} ) && $i < $len ) {
    $s .= $str{$i};
    $i++;
}
if ($s) {
    $t = (int)$s;
} else {
    $this->append_error('no day number. ');
    return(FALSE);
}
while ($str{$i} == '' && $i < $len) $i++;
while ($str{$i} == '.' && $i < $len) $i++;
while ($str{$i} == '-' && $i < $len) $i++;
if ($i == $len) {
    $m = date('n');
    $j = date('Y');
} else {
    $s = "";
    while ($this->_is_int($str{$i}) && $i < $len) {
        $s .= $str{$i};
        $i++;
    }
}
if ($s) {
    $m = (int)$s;
    if ($m <= 0 || $m > 12) {
        $this->append_error('wrong month number: '.$m.');
        return(FALSE);
    }
} else {
    while ($str{$i} != '' && $str{$i} != '.' && $i < $len) {
        $s .= $str{$i};
        $i++;
    }
}
if (!$s) {
    $this->append_error('no information for month. ');
    return(FALSE);
}
if (self::$Monatsnummern[$s]) {
    $m = self::$Monatsnummern[$s];
}

```

```

    } else {
        $this->append_error('wrong month name: '.$s.'.');
        return(FALSE);
    }
}
while ($str{$i} == '' && $i < $len) $i++;
while ($str{$i} == '.' && $i < $len) $i++;
while ($str{$i} == '-' && $i < $len) $i++;
if ($i == $len) {
    $j = date('Y');
} else {
    $s = "";
    while ( $this->_is_int( $str{$i} ) && $i < $len ) {
        $s .= $str{$i};
        $i++;
    }
    if ($s) {
        $j = (int)$s;
        if ($j < 100) $j += 2000;
        if ($j < 2000 || $j >= 3000) {
            $this->append_error('wrong year number: '.$j.'.');
            return(FALSE);
        }
    } else {
        $this->append_error('wrong year description: '.$s.'.');
        return(FALSE);
    }
}
}
$smax = $this->max_Tage_des_Monats($m, $j);
if ($st <= 0 || $st > $smax) {
    $this->append_error('wrong number of days: '.$st.', maximal number of days of the month is ' .
    $smax.'.');
    return(FALSE);
}
$this->max = $smax;
$s = (string)$j;
if ($m <= 9) $s .= '0';
$s .= (string)$m;
if ($st <= 9) $s .= '0';
$s .= (string)$st;
$this->value = $s;
return(TRUE);
}

```

Change format from Termin to string:

```

public function __toString() {
    return($this->string());
}

```

Check year number on leap year (*Schaltjahr* is German for *leap year*):

```
private function _is_Schaltjahr($jahr){
    if(($jahr % 400) == 0 || (($jahr % 4) == 0 && ($jahr % 100) != 0))
        return TRUE;
    else
        return FALSE;
}
```

Check whether string is in Termin format:

```
private function _is_Termin($str) {
    $s = (string)(int)$str;
    $len = strlen($s);
    if ($len != 8) return(FALSE);
    if (!$this->_is_int($str)) return(FALSE);
    if ($str{0} != '2') {
        $this->append_error('wrong year. ');
        return(FALSE);
    }
    return(TRUE);
}
```

Further methods:

```
private function _char2int($c) {
    $n = self::$char2int[$c];
    if (isset($n)) return($n); else return(-1);
}

private function _is_int($str) {
    $len = strlen($str);
    for ($i = 0; $i < $len; $i++) {
        if ($this->_char2int($str{$i}) < 0) return(FALSE);
    }
    return(TRUE);
}

public function is_false() {
    return($this->string() == "");
}

public function is_true() {
    return(!$this->is_false());
}

public function int() {
    return (int)$this->value;
}

public function string() {
    return (string)$this->value;
}
```

```

}

private function append_error($str) {
    if ($this->error) $this->error .= ".$str"; else $this->error = $str;
    return($this->error);
}

public function get_error() {
    return($this->error);
}

```

Output of the actual Termin as date string:

```

public function Datum() {
    if ($this->is_false()) return("");
    $str = $this->string();
    $s = "";
    for ($i = 6; $i <= 7 ; $i++) $s .= $str{$i};
    $s .= ':';
    for ($i = 4; $i <= 5 ; $i++) $s .= $str{$i};
    $s .= ':';
    for ($i = 0; $i <= 3 ; $i++) $s .= $str{$i};
    return($s);
}

public function date_en() {
    if ($this->is_false()) return("");
    $str = $this->string();
    $s = "";
    for ($i = 0; $i <= 3 ; $i++) $s .= $str{$i};
    $s .= '/';
    for ($i = 4; $i <= 5 ; $i++) $s .= $str{$i};
    $s .= '/';
    for ($i = 6; $i <= 7 ; $i++) $s .= $str{$i};
    return($s);
}

```

Replace the day of the actual Termin by 01 (*erster des Monats* is German for *first of the month*):

```

public function erster_des_Monats() {
    if ($this->is_false()) return("");
    $str = $this->string();
    $str[6] = 0;
    $str[7] = 1;
    return(new Termin($str));
}

```

Number of days in the actual month:

```

public function max_Tage_des_Monats() {
    if (func_num_args() == 2) {

```

```

    $m = func_get_arg(0);
    $j = func_get_arg(1);
} else {
    $m = $this->Monat();
    $j = $this->Jahr();
}
if ($m == 2 && $this->_is_Schaltjahr($j)) return(29);
return(self::$max_Tage{$m});
}

```

Actual day as number (*Tag* is German for *day*):

```

public function Tag() {
    if ($this->is_false()) return(0);
    $str = $this->string();
    $s = $str{6}.$str{7};
    return((int)$s);
}

```

Actual month as number (*Monat* is German for *month*):

```

public function Monat() {
    if ($this->is_false()) return(0);
    $str = $this->string();
    $s = $str{4}.$str{5};
    return((int)$s);
}

```

Actual year as number (*Jahr* is German for *year*):

```

public function Jahr() {
    if ($this->is_false()) return(0);
    $str = $this->string();
    $s = $str{0}.$str{1}.$str{2}.$str{3};
    return((int)$s);
}

```

Name of the actual month:

```

public function Monatsname() {
    if ($this->is_false()) return(0);
    $n = $this->Monat();
    if ($n <= 0 || $n > 12) return("");
    return(self::$Monatsnamen[$n]);
}

```

Put variable *timestamp* using the actual date and return the variable value:

```

public function timestamp() {
    if ($this->timestamp == 0) $this->timestamp = strtotime($this->Datum());
    return($this->timestamp);
}

```

```
}
```

Output of the actual week day as whole word:

```
public function get_Wochentag() {  
    return(strftime('%A', $this->timestamp()));  
}
```

Output of the abbreviation of the actual week day with 2 letters:

```
public function get_Wochentag_Abk() {  
    return(strftime('%a', $this->timestamp()));  
}
```

Actual week day as number (for German names), 1=Montag, 7=Sonntag:

```
public function get_Wochentag_Zahl() {  
    return(strftime('%u', $this->timestamp()));  
}
```

Actual week of the year as number (e.g. 26), possible values are 00 upto 53, starting with the first Monday of the first week (i.e. 1 as last week of the previous year):

```
public function get_Kalenderwoche() {  
    return(strftime('%W', $this->timestamp()));  
}
```

strftime applied to the actual Termin:

```
public function strftime($str) {  
    return(strftime($str, $this->timestamp()));  
}
```

End of file:

```
}  
?>
```

3. Additions

3.1 smarty

Smarty is a TYPO3 extension. So it must be installed first.

It is often used in both plugins of the extension. There are some few rules to be applied for it. We use *smarty* to pass parts of the PHP code into templates. These templates are HTML codes that have additional special variables; these work simply as abbreviations that are defined in the PHP code.

First some files of the extension *smarty* must be embedded. This happens in the file

pi1/class.tx_netcosrenting_pi1.php:

```
require_once(t3lib_extMgm::extPath('smarty').'class.tx_smarty.php');
require_once(t3lib_extMgm::extPath('smarty').'Smarty/libs/Smarty.class.php');
```

In the PHP code, a local instance of the class **Smarty** is defined. We call this instance **smarty**.

```
$this->smarty = new Smarty;
```

Then the position of the *smarty* templates must be given. They are all kept in subdirectories of the directory **tmpl** in the extension directory; we define this directory by the argument **template_dir**. Moreover, **tmpl_c** is used internally; this is defined by the argument **compile_dir**. In both of these configurations, the extension directory is supplied by the PHP constructions **t3lib_extMgm::extPath(\$this->extKey)**.

```
$this->smarty->compile_dir = t3lib_extMgm::extPath($this->extKey).'tmpl_c';
$this->smarty->template_dir = t3lib_extMgm::extPath($this->extKey).'tmpl';
```

Next we define how the abbreviation variables can be recognized in the HTML templates.

```
$this->smarty->left_delimiter = '{..}';
$this->smarty->right_delimiter = '..}';
```

Here the format of the abbreviation is **{..\$VALUE..}**, where **VALUE** is in PHP. The putting of these abbreviation variables in PHP is done by the command **assign**.

```
$this->smarty->assign('VALUE', 1234);
```

All of these abbreviations are set within a PHP script, mostly by results from a *form* tag (using POST or GET).

The *smarty* template is run by

```
$content = $this->smarty->fetch($this->smarty->template_dir.'/Datei.tpl');
return $this->pi_wrapInBaseClass($content);
```

That's it.

3.2 jscalendar

Jscalendar allows to use comfortable calendars in own extensions. It is available as extension *netcos_jscalendar*. After the installation, the source code is positioned at *typo3conf/ext/netcos_jscalendar/jscalendar/*. This directory is relative to the domain URL, such that the root directory */* is the main directory of the TYPO3 installation. This can be obtained when a domain is provided in the TYPO3 code.

This program is written in JavaScript and allows the representation of comfortable calendars at date input fields. To use it, it must be embedded by *script* tags in the HTML code.

In the header of each HTML *smarty* template file, the following code is provided:

```
<style type="text/css">@import
  url(/typo3conf/ext/netcos_jscalendar/jscalendar/calendar-green.css);</style>
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/calendar.js"></script>
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/calendar-setup.js"></script>
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/lang/calendar-en.js"></script>
```

Therein a CSS file is embedded first and then 3 source files of the calendar system. The configuration file for the English language *lang/calendar-en.js* is really needed, even when one works with different languages.

When the German language is active the script *lang/calendar-de.js* must be called:

```
<script type="text/javascript"
  src="/typo3conf/ext/netcos_jscalendar/jscalendar/lang/calendar-de.js"></script>
```

As we must embed these calls into each template file and the checking of the language is not possible in HTML we use a *smarty* variable for embedding these scripts. This variable is defined in each main class file for each plugin. More details are provided in chapter 4.

In the *body* region of a template file, the calendar is called within an input form, e.g. With the following code:

```
<form method="post">
  <input type="text" id="data_datum" name="data_datum" />
  <button id="trigger_datum">Datum</button>
  <script type="text/javascript">
    Calendar.setup(
      {
        inputField : "data_Datum", // ID of the input field
        ifFormat   : "%d.%m.%Y", // the date format
        button     : "trigger_datum" // ID of the button
      }
    );
  </script>
  <input type="submit" name="anlegen" value="anlegen" />
</form>
```

These are all structures for *jscalendar*.

3.3 Multilingualism

TYPO3 is able to use several languages. The terms for the different languages are defined in the *locallang*.xml* files. In the main directory of the extension, there are the files *locallang.xml* and *locallang-db.xml* for the general definitions of the extension; moreover, there is a file *locallang.xml* in every plugin subdirectory that is special for this extension.

Within the plugin, the multilingualism must be activated by the command *pi_loadLL*, the terms are

gotten by the command *pi_getLL*. Such a term has always its value for the actual language.

locallang.xml:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
  <meta type="array">
    <type>module</type>
    <description>Language labels for extension 'netcos_renting'</description>
  </meta>
  <data type="array">
    <languageKey index="default" type="array">
      <label index="pi1_title">Renting Administration</label>
      <label index="pi2_title">Rented</label>
    </languageKey>
    <languageKey index="de" type="array">
      <label index="pi1_title">Vermietungsverwaltung</label>
      <label index="pi2_title">vermietet</label>
    </languageKey>
  </data>
</T3locallang>
```

Locallang-db.xml

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
  <meta type="array">
    <type>database</type>
    <description>Language labels for database tables/fields belonging to extension
'netcos_renting'</description>
  </meta>
  <data type="array">
    <languageKey index="default" type="array">
      <label index="tx_netcosrenting_fromto">Rented</label>
      <label index="tx_netcosrenting_fromto.from">From</label>
      <label index="tx_netcosrenting_fromto.to">To</label>
      <label index="tt_content.list_type_pi1">Renting Administration</label>
      <label index="tt_content.list_type_pi2">Rented</label>
    </languageKey>
    <languageKey index="de" type="array">
      <label index="tx_netcosrenting_fromto">vermietet</label>
      <label index="tx_netcosrenting_fromto.from">Von</label>
      <label index="tx_netcosrenting_fromto.to">Bis</label>
      <label index="tt_content.list_type_pi1">Vermietungsverwaltung</label>
      <label index="tt_content.list_type_pi2">vermietet</label>
    </languageKey>
  </data>
</T3locallang>
```

pi1/locallang.xml:

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
```

```

<T3locallang>
<meta type="array">
  <type>module</type>
  <description>Language labels for plugin &quot;tx_netcosrenting_pil&quot;</description>
</meta>
<data type="array">
  <languageKey index="default" type="array">
    <label index="actual">actual</label>
    <label index="apply">apply</label>
    <label index="back">Back</label>
    <label index="cancel">cancel</label>
    <label index="change1">change</label>
    <label index="change2">change</label>
    <label index="delete">delete</label>
    <label index="existing">existing</label>
    <label index="from">from</label>
    <label index="language">en</label>
    <label index="more">more</label>
    <label index="new">new</label>
    <label index="to">to</label>
    <label index="year">year</label>
    <label index="administration_bookings_actual">Bookings since today</label>
    <label index="administration_bookings_year">Bookings of Year</label>
    <label index="administration_date_format">Date format: yyyy/mm/dd</label>
    <label index="administration_header">Renting</label>
    <label index="administration_new">New Entry</label>
    <label index="administration_none">No entries available.</label>
    <label index="administration_years">Bookings of whole Years</label>
    <label index="calendar_ifformat">%Y/%m/%d</label>
    <label index="exists_header">The entry was not generated because it exists already.</label>
    <label index="overlap_header">Overlap of Entries</label>
    <label index="pil_comes_after">comes after</label>
    <label index="pil_erroneous">is erroneous</label>
    <label index="pil_has_an_error">has an error</label>
    <label index="pil_has_error">has error</label>
    <label index="to_be_changed_header">Change Entry</label>
    <label index="to_be_deleted_header">The following entry should be deleted</label>
    <label index="was_created_header">The entry was created.</label>
    <label index="was_deleted_header">The entry was deleted.</label>
    <label index="wrong_entry_header">Wrong entry</label>
  </languageKey>
  <languageKey index="de" type="array">
    <label index="actual">aktuell</label>
    <label index="apply">anlegen</label>
    <label index="back">Zurück</label>
    <label index="cancel">Abbrechen</label>
    <label index="change1">ändern</label>
    <label index="change2">wechseln</label>
    <label index="delete">löschen</label>
    <label index="existing">bestehend</label>
    <label index="from">Von</label>
    <label index="language">de</label>
  </languageKey>
</data>
</T3locallang>

```

```

<label index="more">Weiter</label>
<label index="new">neu</label>
<label index="to">Bis</label>
<label index="year">Jahr</label>
<label index="administration_bookings_actual">Buchungen ab heute</label>
<label index="administration_bookings_year">Buchungen Jahr</label>
<label index="administration_date_format">Datumsformat: tt.mm.jjjj</label>
<label index="administration_header">Vermietung</label>
<label index="administration_new">Neuer Eintrag</label>
<label index="administration_none">Keine Einträge vorhanden.</label>
<label index="administration_years">Buchungen von ganzen Jahren</label>
<label index="calendar_ifformat">%d.%m.%Y</label>
<label index="exists_header">Der Datensatz wurde nicht erstellt, da er bereits
existiert.</label>
<label index="overlap_header">Einträge überschneiden sich</label>
<label index="pi1_comes_after">kommt nach</label>
<label index="pi1_erroneous">ist fehlerhaft</label>
<label index="pi1_has_an_error">hat einen Fehler</label>
<label index="pi1_has_error">hat den Fehler</label>
<label index="to_be_changed_header">Eintrag ändern</label>
<label index="to_be_deleted_header">Folgender Eintrag soll gelöscht werden</label>
<label index="was_created_header">Der Datensatz wurde erstellt.</label>
<label index="was_deleted_header">Der Datensatz wurde gelöscht.</label>
<label index="wrong_entry_header">Falscher Eintrag</label>
</languageKey>
</data>
</T3locallang>

```

pi2/locallang.xml:

```

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<T3locallang>
<meta type="array">
<type>module</type>
<description>Language labels for plugin &quot;tx_netcosrenting_pi2&quot;</description>
</meta>
<data type="array">
<languageKey index="default" type="array">
<label index="listFieldHeader_from">From</label>
<label index="listFieldHeader_to">To</label>
<label index="list_mode_1">Mode 1</label>
<label index="list_mode_2">Mode 2</label>
<label index="list_mode_3">Mode 3</label>
<label index="back">Back</label>
<label index="pi_list_browserresults_prev">&lt; Previous</label>
<label index="pi_list_browserresults_page">Page</label>
<label index="pi_list_browserresults_next">Next &gt;</label>
<label index="pi_list_browserresults_displays">Displaying results ####SPAN_BEGIN####%s to
%s&lt;/span&gt; out of ####SPAN_BEGIN####%s&lt;/span&gt;</label>
<label index="pi_list_searchBox_search">Search</label>

<label index="booked">booked</label>

```

<label index="free">frei</label>
<label index="year">year</label>

<label index="previous_month">Previous Month</label>
<label index="actual_month">Actual Month</label>
<label index="next_month">Next Month</label>

<label index="january">January</label>
<label index="february">February</label>
<label index="march">March</label>
<label index="april">April</label>
<label index="may">May</label>
<label index="june">June</label>
<label index="july">July</label>
<label index="august">August</label>
<label index="september">September</label>
<label index="october">October</label>
<label index="november">November</label>
<label index="december">December</label>

<label index="calendar_week">calendar week</label>

<label index="monday">Monday</label>
<label index="tuesday">Tuesday</label>
<label index="wednesday">Wednesday</label>
<label index="thursday">Thursday</label>
<label index="friday">Friday</label>
<label index="saturday">Saturday</label>
<label index="sunday">Sunday</label>

<label index="booked_change">change month</label>
<label index="booked_future">Choose future month</label>
<label index="booked_header">Renting</label>

<label index="error_back">back to the actual month</label>
<label index="error_header">No Access to Month</label>
<label index="error_text">The oldest possible month is the previous month.
No limits for the future.</label>

</languageKey>

<languageKey index="de" type="array">
<label index="booked">gebucht</label>
<label index="free">frei</label>
<label index="year">Jahr</label>

<label index="previous_month">Voriger Monat</label>
<label index="actual_month">Aktueller Monat</label>
<label index="next_month">Nächster Monat</label>

<label index="january">Januar</label>
<label index="february">Februar</label>
<label index="march">März</label>
<label index="april">April</label>

<label index="may">Mai</label>
<label index="june">Juni</label>
<label index="july">Juli</label>
<label index="august">August</label>
<label index="september">September</label>
<label index="october">Oktober</label>
<label index="november">November</label>
<label index="december">Dezember</label>

<label index="calendar_week">Kalenderwoche</label>

<label index="monday">Montag</label>
<label index="tuesday">Dienstag</label>
<label index="wednesday">Mittwoch</label>
<label index="thursday">Donnerstag</label>
<label index="friday">Freitag</label>
<label index="saturday">Samstag</label>
<label index="sunday">Sonntag</label>

<label index="booked_change">Monat wechseln</label>
<label index="booked_future">Zukünftigen Monat wählen:</label>
<label index="booked_header">Vermietung</label>

<label index="error_back">Zurück zum aktuellen Monat</label>
<label index="error_header">Kein Zugriff auf den Monat</label>
<label index="error_text">In die Vergangenheit kann höchstens auf den
Vormonat zugegriffen werden. Für die Zukunft bestehen keine
Einschränkungen. </label>

</languageKey>

</data>

</T3locallang>

4. Suit the PHP source code

The main changes occur in every plugin directory *typo3conf/ext/netcos_renting/pi?* in the file *class.tx_netcosrenting_pi?.php*. The symbol ? stands for 1 oder 2 because we configured two plugins. These files build the extension classes for **tslib_pibase**. Therein, the method **main** calls the whole representation process.

4.1 Database access

Normally TYPO3 uses the database *MySQL*. But a general concept for accessing other databases as well was created. This generated new functions that replace the known *MySQL* commands of PHP.

We use this concept although we actually access only *MySQL*. For that the TYPO3 extension *dbal* must be installed. Documentation for their database functions is available at [TYPO3 Core Documentation, chapter 1.4](#).

We use the following variable:

```
$table_name = 'tx_netcosrenting_fromto';
```

In the plugins, the following database accesses are used:

```
$where = 'uid='.$$uid;
$GLOBALS['TYPO3_DB']->exec_DELETEquery($table_name, $where);
```

This deletes the entry with uid *\$uid* from the table *\$table_name*.

```
$query = 'SELECT uid FROM '.$table_name .
' WHERE from_date='.$$f->string().' AND to_date='.$$t->string();
$res = $GLOBALS['TYPO3_DB']->sql_query($query);
if ($GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
    ...
} else {
    $fields_values = array('from_date' => $f->string(), 'to_date' => $t->string());
    $GLOBALS['TYPO3_DB']->exec_INSERTquery($table_name, $fields_values);
    ...
}
```

This is a SELECT request called by *sql_query()*. The result **\$res** is split by *sql_fetch_assoc()* into associative arrays. When no suitable data record exists a new data record is generated from the values; this is stored by *exec_INSERTquery()* in the data table.

4.2 First Plugin (Renting Administration)

In this section, we introduce the file *typo3conf/ext/netcos_renting/pi1/class.tx_netcosrenting_pi1.php*. As it is very long we split it in many parts that must be combined. But of course you can download the whole file from the extension source code.

The line breaks must be removed.

Start and license:

```
<?php
```

```

/*****
* Copyright notice
*
* (c) 2007,2008 Bernd Warken <be.warken@netcos.de>
* All rights reserved
*
* This script is part of the TYPO3 project. The TYPO3 project is
* free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* The GNU General Public License can be found at
* http://www.gnu.org/copyleft/gpl.html.
*
* This script is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* This copyright notice MUST APPEAR in all copies of the script!
*****/

```

Embedding of PHP files:

```

require_once(PATH_tslib.'class.tslib_pibase.php');
require_once(t3lib_extMgm::extPath('netcos_renting').'class_termin.php');
require_once(t3lib_extMgm::extPath('smarty').'class.tx_smarty.php');
require_once(t3lib_extMgm::extPath('smarty').'Smarty/libs/Smarty.class.php');

```

Documentation of *kickstarter*:

```

/**
* Plugin 'own Booking Administration' for the 'eigene_buchungen' extension.
*
* @author Bernd Warken <be.warken@netcos.de>
* @package TYPO3
* @subpackage tx_eigenebuchungen
*/

```

Class definition and variables of *kickstarter*:

```

class tx_netcosrenting_pi1 extends tslib_pibase {
    var $prefixId = 'tx_netcosrenting_pi1'; // Same as class name
    var $scriptRelPath = 'pi1/class.tx_netcosrenting_pi1.php';
    // Path to this script relative to the extension dir.
    var $extKey = 'netcos_renting'; // The extension key.
}

```

Assures that the data from HTML forms is sent and received (POST und GET):

```

var $pi_checkCHash = true;

```

Documentation of *kickstarter* for function *main*:

```
/**
 * Main method of your PlugIn
 *
 * @param string $content: The content of the PlugIn
 * @param array $conf: The PlugIn Configuration
 * @return The content that should be displayed on the website
 */
function main($content,$conf) {
```

From now on we are located within the function *main*.

Store the data from the previous run in *\$POST*:

```
global $_POST, $LANG;
$POST = t3lib_div::_POST();
$_POST = array();
```

Activate the multilingualism:

```
$this->pi_loadLL();
```

General configuration for *smarty*:

```
$this->smarty = new Smarty;
$this->smarty->compile_dir = t3lib_extMgm::extPath($this->extKey).'tpl_c';
$this->smarty->template_dir = t3lib_extMgm::extPath($this->extKey).'tpl';
$this->smarty->left_delimiter = '{..';
$this->smarty->right_delimiter = '..}';
```

That configures that in the HTML template files of *smarty* variables of the form *{..\$var..}* can be used. The values for these variables is set in PHP with the command *\$this->smarty->assign*.

The used *smarty* HTML template files:

```
$this->templates['administration'] = 'tpl/pi1/administration.tpl';
$this->templates['wrong_entry'] = 'tpl/pi1/wrong_entry.tpl';
$this->templates['overlap'] = 'tpl/pi1/overlap.tpl';
$this->templates['to_be_changed'] = 'tpl/pi1/to_be_changed.tpl';
$this->templates['to_be_deleted'] = 'tpl/pi1/to_be_deleted.tpl';
$this->templates['exists'] = 'tpl/pi1/exists.tpl';
$this->templates['was_created'] = 'tpl/pi1/was_created.tpl';
$this->templates['was_deleted'] = 'tpl/pi1/was_deleted.tpl';
```

Smarty abbreviation variables:

```
$this->smarty->assign('DIR_JSCALENDAR', '/typo3conf/ext/'.$this->extKey.'/jscalendar-1.0/');
```

Name of the *MySQL* table of the plugin:

```
$table_name = 'tx_netcosrenting_fromto';
```

Date of today:

```
$heutetermin = new Termin((string)date('d').'!.(string)date('n').'!.(string)date('Y'));
```

In the *Termin* of today, replace the day by 01:

```
$erster_des_Monats = $heutetermin->erster_des_Monats();
```

Smarty variables:

```
$this->smarty->assign('ACTUAL', $this->pi_getLL('actual'));
$this->smarty->assign('APPLY', $this->pi_getLL('apply'));
$this->smarty->assign('CANCEL', $this->pi_getLL('cancel'));
$this->smarty->assign('CHANGE1', $this->pi_getLL('change1'));
$this->smarty->assign('CHANGE2', $this->pi_getLL('change2'));
$this->smarty->assign('DELETE', $this->pi_getLL('delete'));
$this->smarty->assign('EXISTING', $this->pi_getLL('existing'));
$this->smarty->assign('FROM', $this->pi_getLL('from'));
$this->smarty->assign('MORE', $this->pi_getLL('more'));
$this->smarty->assign('NEW', $this->pi_getLL('new'));
$this->smarty->assign('TO', $this->pi_getLL('to'));
$this->smarty->assign('YEAR', $this->pi_getLL('year'));
$this->smarty->assign('ADMINISTRATION_DATE_FORMAT',
    $this->pi_getLL('administration_date_format'));
$this->smarty->assign('ADMINISTRATION_HEADER',
    $this->pi_getLL('administration_header'));
$this->smarty->assign('ADMINISTRATION_NEW', $this->pi_getLL('administration_new'));
$this->smarty->assign('ADMINISTRATION_YEARS',
    $this->pi_getLL('administration_years'));
$this->smarty->assign('CALENDAR_IFFORMAT', $this->pi_getLL('calendar_iformat'));
$this->smarty->assign('EXISTS_HEADER', $this->pi_getLL('exists_header'));
$this->smarty->assign('OVERLAP_HEADER', $this->pi_getLL('overlap_header'));
$this->smarty->assign('TO_BE_CHANGED_HEADER',
    $this->pi_getLL('to_be_changed_header'));
$this->smarty->assign('TO_BE_DELETED_HEADER',
    $this->pi_getLL('to_be_deleted_header'));
$this->smarty->assign('WAS_CREATED_HEADER', $this->pi_getLL('was_created_header'));
$this->smarty->assign('WAS_DELETED_HEADER', $this->pi_getLL('was_deleted_header'));
$this->smarty->assign('WRONG_ENTRY_HEADER',
    $this->pi_getLL('wrong_entry_header'));
```

By *pi_getLL*, the multilingual terms for the actual language are called. The terms are defined in *pi1/locallang.xml* for this plugin.

Configuration for the calendar:

```
$dir_jscalendar = '/typo3conf/ext/netcos_jscalendar/jscalendar/';
$import_jscalendar = '
<style type="text/css">@import url('.$dir_jscalendar.'calendar-green.css);</style>
<script type="text/javascript" src="'.$dir_jscalendar.'calendar.js"></script>
<script type="text/javascript" src="'.$dir_jscalendar.'calendar-setup.js"></script>
<script type="text/javascript" src="'.$dir_jscalendar.'lang/calendar-en.js"></script>';
';
$lang = $this->pi_getLL('language');
if ($lang == 'en') {
    $date_lang = 'date_en';
} else {
    $date_lang = 'Datum';
}
$import_jscalendar .= '<script type="text/javascript" src="'.
    $dir_jscalendar.'lang/calendar-' . $lang . '.js"></script>';
```

```
}  
$this->smarty->assign('IMPORT_JSCALENDAR', $import_jscalendar);
```

By that, the German language file is only embedded when German is the active language.

The following sections of the kind **if (\$POST['...']) {...}** retrieve POST parameter from former runs. Eventually some *smarty* abbreviation variables are set by **assign**. The command **fetch** calls a *smarty* template file. Its content **\$content** finishes this run by

```
return $this->pi_wrapInBaseClass($content);
```

```
if ($POST['to_be_deleted']) {  
    $this->smarty->assign('UID', $POST['uid']);  
    $this->smarty->assign('VALUE_FROM', $POST['data_from']);  
    $this->smarty->assign('VALUE_TO', $POST['data_to']);  
    $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).  
        $this->templates['to_be_deleted']);  
    return $this->pi_wrapInBaseClass($content);  
}  
  
if ($POST['to_be_changed']) {  
    $this->smarty->assign('UID', $POST['uid']);  
    $this->smarty->assign('VALUE_FROM', $POST['data_from']);  
    $this->smarty->assign('VALUE_TO', $POST['data_to']);  
    $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).  
        $this->templates['to_be_changed']);  
    return $this->pi_wrapInBaseClass($content);  
}
```

Here the data record with a given UID is deleted from the table:

```
if ($POST['delete']) {  
    $uid = $POST['uid'];  
    if ($uid) {  
        $where = 'uid='.$uid;  
        $GLOBALS['TYPO3_DB']->exec_DELETEquery($table_name, $where);  
        $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).  
            $this->templates['was_deleted']);  
        return $this->pi_wrapInBaseClass($content);  
    }  
}
```

Here the data record with a given UID is deleted from the table as well and the variable *\$run_apply* is configured for the next section:

```
$run_apply = FALSE;  
if ($POST['change']) {  
    $uid = $POST['uid'];  
    if ($uid) {  
        $where = 'uid='.$uid;  
        $GLOBALS['TYPO3_DB']->exec_DELETEquery($table_name, $where);  
    }  
    $run_apply = TRUE;
```

```
}
```

Here a new data record is created:

```
if ($POST['apply'] || $run_apply) {
    $richtig = TRUE;
    $errmsg = "";
    $df = $POST['data_from'];
    if ($df) {
        $f = new Termin($df);
        if ($f->is_true()) {

        } else {
            $richtig = FALSE;
            $errmsg .= "".$this->pi_getLL("from")." " . $this->pi_getLL("pi1_erroneous")."!";
            $f->get_error();
        }
    } else {
        $richtig = FALSE;
        $errmsg .= "".$this->pi_getLL("from")." " . $this->pi_getLL("pi1_erroneous")."!";
    }
    $dt = $POST['data_to'];
    if ($dt) {
        $t = new Termin($dt);
        if ($t->is_true()) {

        } else {
            $richtig = FALSE;
            $errmsg .= "".$this->pi_getLL("to")." " . $this->pi_getLL("pi1_erroneous")."!";
            $t->get_error();
        }
    } else {
        $richtig = FALSE;
        $errmsg .= "".$this->pi_getLL("to")." " . $this->pi_getLL("pi1_erroneous")."!";
    }
    if ($richtig) {
        $query = 'SELECT uid FROM '.$table_name .
            ' WHERE from_date='.$f->string().' AND to_date='.$t->string();
        $res = $GLOBALS['TYPO3_DB']->sql_query($query);
        if ($GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
            $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
                $this->templates['exists']);
            return $this->pi_wrapInBaseClass($content);
        } else {
            $fields_values = array('from_date' => $f->string(), 'to_date' => $t->string());
            $GLOBALS['TYPO3_DB']->exec_INSERTquery($table_name, $fields_values);
            $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
                $this->templates['was_created']);
            return $this->pi_wrapInBaseClass($content);
        }
    } else {
        $this->smarty->assign('ERRMSG', $errmsg);
        $this->smarty->assign('UID', $uid);
    }
}
```

```

$this->smarty->assign('VALUE_FROM', $df);
$this->smarty->assign('VALUE_TO', $dt);
$content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
    $this->templates['wrong_entry']);
return $this->pi_wrapInBaseClass($content);
}
}

```

Check whether a changing of year is intended to be produced:

```

$jahr = 0;
if ($POST['wechseln']) {
    $jahr = $POST['jahr'];
    if ($jahr < 100) $jahr += 2000;
    if ($jahr >= 3000 || $jahr < 2000) $jahr = 0;
}

```

Search all booked Termins from the database table:

```

$query = 'SELECT uid,from_date,to_date FROM '.$table_name;
$res = $GLOBALS['TYPO3_DB']->sql_query($query);
$fromto = array();
$from = array();
while ($row = $GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
    $u = $row["uid"];
    $f = new Termin($row["from_date"]);
    $t = new Termin($row["to_date"]);
    $act = FALSE;
    $errmsg = "";

    if ($f->is_false() || $t->is_false()) {
        $value_from = "";
        $value_to = "";
        $act = TRUE;

        if ($f->is_false()) {
            $msg = $f->get_error();
            if ($msg) {
                $errmsg .= ' "' . $this->pi_getLL("from") . "' ' . $this->pi_getLL("pi1_has_error") . "': '$msg;
            } else {
                $errmsg .= ' "' . $this->pi_getLL("from") . "' ' . $this->pi_getLL("pi1_has_an_error") . "': ";
            }
        } else {
            $value_from = $f->$date_lang();
        }
    }

    if ($t->is_false()) {
        $msg = $t->get_error();
        if ($msg) {
            $errmsg .= ' "' . $this->pi_getLL("to") . "' ' . $this->pi_getLL("pi1_has_error") . "': '$msg;
        } else {
            $errmsg .= ' "' . $this->pi_getLL("to") . "' ' . $this->pi_getLL("pi1_has_an_error") . "': ";
        }
    }
}

```

```

    } else {
        $value_to = $t->$date_lang();
    }
} else {
    if ($f->int() > $t->int()) {
        $value_from = $f->$date_lang();
        $value_to = $t->$date_lang();
        $errmsg = "'. $this->pi_getLL("from")."' ' $this->pi_getLL("pi1_comes_after").' '".
            $this->pi_getLL("to")."'.';
        $act = TRUE;
    }
}
if ($act) {
    $this->smarty->assign('ERRMSG', $errmsg);
    $this->smarty->assign('UID', $u);
    $this->smarty->assign('VALUE_FROM', $value_from);
    $this->smarty->assign('VALUE_TO', $value_to);
    $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
        $this->templates['wrong_entry']);
    return $this->pi_wrapInBaseClass($content);
}
$from[$u] = $f;
$fromto[$u] = array("from_date" => $f, "to_date" => $t);
}
asort($from);
$sortiert = array_keys($from);

```

Diagnose errors when 2 data records overlap:

```

$t0 = 0;
$f0 = 0;
$u0 = 0;
foreach ($sortiert as $uid) {
    $f = $from[$uid];
    $f1 = $f->int();
    $t = $fromto[$uid]['to_date'];
    $t1 = $t->int();

    if ($t0 > $f1) {
        $this->smarty->assign('UID1', $u0);
        $this->smarty->assign('UID2', $uid);
        $this->smarty->assign('VALUE_FROM1', $from[$u0]->$date_lang());
        $this->smarty->assign('VALUE_TO1', $fromto[$u0]['to_date']->$date_lang());
        $this->smarty->assign('VALUE_FROM2', $f->$date_lang());
        $this->smarty->assign('VALUE_TO2', $t->$date_lang());
        $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
            $this->templates['overlap']);
        return $this->pi_wrapInBaseClass($content);
    }

    $f0 = $f1;
    $t0 = $t1;
}

```

```

    $u0 = $uid;
}

```

Filter the suitable booking terms:

```

$aktuell_sortiert = array();
if ($jahr) {
    $this->smarty->assign('BOOKINGS_TITLE',
        $this->pi_getLL('administration_bookings_year').'.$jahr);
    $erster = new Termin('1.1.'.$jahr);
    $letzter = new Termin('31.12.'.$jahr);
    foreach ($sortiert as $uid) {
        $f = $from[$uid];
        $t = $fromto[$uid]['to_date'];
        if ( ($f >= $erster && $f <= $letzter) ||
            ($t >= $erster && $t <= $letzter) ||
            ($f < $erster && $t > $letzter)
        ) array_push($aktuell_sortiert, $uid);
    }
} else {
    $this->smarty->assign('BOOKINGS_TITLE',
        $this->pi_getLL('administration_bookings_actual'));
    foreach ($sortiert as $uid) {
        if ($fromto[$uid]['to_date'] > $heutetermin) array_push($aktuell_sortiert, $uid);
    }
}

```

Create a diagramm for month:

```

if (count($aktuell_sortiert) == 0) {
    $aktuelle_buchungen = $this->pi_getLL('administration_none');
} else {
    $aktuelle_buchungen = '<table width="350px"><tr align="center">
        <th>UID</th><th>'.$this->pi_getLL("from").'</th><th>
        $this->pi_getLL("to").'</th><th></th><th></th></tr>';
    foreach ($aktuell_sortiert as $uid) {
        $aktuelle_buchungen .= '<tr>
            <form method="post">
            <td style="background-color: yellow;">
                <input type="hidden" name="uid" value="'. $uid.' />'. $uid.'
            </td>
            <td style="background-color: LightGreen;">
                <input type="hidden" name="data_from" value="'. $from[$uid]->$date_lang().' />'.
                $from[$uid]->$date_lang().'</td>
            <td style="background-color: yellow;">
                <input type="hidden" name="data_to" value="'. $fromto[$uid]["to_date"]->$date_lang().
                ' />'. $fromto[$uid]["to_date"]->$date_lang().'</td>
            <td><input type="submit" name="to_be_deleted" value="'. $this->pi_getLL("delete").'" />
            </td>
            <td><input type="submit" name="to_be_changed" value="'. $this->pi_getLL("change1").
                '" /></td>
            <td></td>
            </form></tr>';
    }
}

```

```

    }
    $aktuelle_buchungen .= '</table>';
}
$this->smarty->assign('AKTUELLES_JAHR', $heutetermin->Jahr());
$this->smarty->assign('AKTUELLE_BUCHUNGEN', $aktuelle_buchungen);
$content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
    $this->templates['administration']);
return $this->pi_wrapInBaseClass($content);

```

End of file:

```

    }
}

if (defined('TYPO3_MODE') && $TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
    ['ext/netcos_renting/pi1/class.tx_netcosrenting_pi1.php']) {
    include_once($TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
        ['ext/netcos_renting/pi1/class.tx_netcosrenting_pi1.php']);
}
?>

```

4.3 Second Plugin (rented)

In this section, we represent the file *typo3conf/ext/netcos_renting/pi2/class.tx_netcosrenting_pi2.php*. As it is very long we split it in many parts that must be combined. But you can also extract the whole file from the extension source code.

The line breaks must be removed.

Start and license:

```

<?php
/*****
* Copyright notice
*
* (c) 2007,2008 Bernd Warken <be.warken@netcos.de>
* All rights reserved
*
* This script is part of the TYPO3 project. The TYPO3 project is
* free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* The GNU General Public License can be found at
* http://www.gnu.org/copyleft/gpl.html.
*
* This script is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

```

```
* GNU General Public License for more details.
*
* This copyright notice MUST APPEAR in all copies of the script!
*****/
```

Embedding of PHP files:

```
require_once(PATH_tslib.'class.tslib_pibase.php');
require_once('class_termin.php');
require_once(t3lib_extMgm::extPath('smarty').'class.tx_smarty.php');
require_once(t3lib_extMgm::extPath('smarty').'Smarty/libs/Smarty.class.php');
```

Kickstarter documentation, class definition, used variables:

```
/**
 * Plugin 'vermietet' for the 'netcos_renting' extension.
 *
 * @author Bernd Warken <be.warken@netcos.de>
 * @package TYPO3
 * @subpackage tx_netcosrenting
 */
class tx_netcosrenting_pi2 extends tslib_pibase {
    var $prefixId = 'tx_netcosrenting_pi2'; // Same as class name
    var $scriptRelPath = 'pi2/class.tx_netcosrenting_pi2.php';
    // Path to this script relative to the extension dir.
    var $extKey = 'netcos_renting'; // The extension key.

    var $pi_checkCHash = true;
```

Method *main()*:

```
/**
 * Main method of your PlugIn
 *
 * @param string $content: The content of the PlugIn
 * @param array $conf: The PlugIn Configuration
 * @return The content that should be displayed on the website
 */
function main($content,$conf) {
    $this->conf=$conf;
    $this->pi_setPiVarDefaults();
    $this->pi_loadLL();
```

Store the data from the previous run in \$POST:

```
global $ _POST;
$POST = t3lib_div::_POST();
$_POST = array();
```

Activate the multilingualism:

```
$this->pi_loadLL();
```

Smarty configuration:

```
$this->smarty = new Smarty;
$this->smarty->compile_dir = t3lib_extMgm::extPath($this->extKey).'tmpl_c';
$this->smarty->template_dir = t3lib_extMgm::extPath($this->extKey).'tmpl';
$this->smarty->left_delimiter = '{..';
$this->smarty->right_delimiter = '..}';

$this->templates['booked'] = 'tmpl/pi2/booked.tpl';
$this->templates['error'] = 'tmpl/pi2/error.tpl';
```

Name of the *MySQL* table:

```
$table_name = 'tx_netcosrenting_fromto';
```

Smarty variables for the HTML template files:

```
$this->smarty->assign('PREVIOUS_MONTH', $this->pi_getLL('previous_month'));
$this->smarty->assign('ACTUAL_MONTH', $this->pi_getLL('actual_month'));
$this->smarty->assign('NEXT_MONTH', $this->pi_getLL('next_month'));

$this->smarty->assign('MONDAY', $this->pi_getLL('monday'));
$this->smarty->assign('TUESDAY', $this->pi_getLL('tuesday'));
$this->smarty->assign('WEDNESDAY', $this->pi_getLL('wednesday'));
$this->smarty->assign('THURSDAY', $this->pi_getLL('thursday'));
$this->smarty->assign('FRIDAY', $this->pi_getLL('friday'));
$this->smarty->assign('SATURDAY', $this->pi_getLL('saturday'));
$this->smarty->assign('SUNDAY', $this->pi_getLL('sunday'));

$this->smarty->assign('CALENDAR_WEEK', $this->pi_getLL('calendar_week'));

$this->smarty->assign('BOOKED_CHANGE', $this->pi_getLL('booked_change'));
$this->smarty->assign('BOOKED_FUTURE', $this->pi_getLL('booked_future'));
$this->smarty->assign('BOOKED_HEADER', $this->pi_getLL('booked_header'));

$this->smarty->assign('ERROR_BACK', $this->pi_getLL('error_back'));
$this->smarty->assign('ERROR_HEADER', $this->pi_getLL('error_header'));
$this->smarty->assign('ERROR_TEXT', $this->pi_getLL('error_text'));
```

Variables for the actual date:

```
$aktuelles_jahr = date('Y');
$aktueller_monat = date('n');
$heutetermin = new Termin((string)date('d').'.'.(string)date('n').'.'.(string)date('Y'));
$monate = array(0, $this->pi_getLL("january"), $this->pi_getLL("february"),
    $this->pi_getLL("march"), $this->pi_getLL("april"), $this->pi_getLL("mai"),
    $this->pi_getLL("june"), $this->pi_getLL("july"), $this->pi_getLL("august"),
    $this->pi_getLL("september"), $this->pi_getLL("october"),
    $this->pi_getLL("november"), $this->pi_getLL("december"));
```

Variables from POST values of the previous run:

```
$monat = $POST['Monat'];  
$jahr = $POST['Jahr'];
```

Change to the suitable month:

```
$aktuell_text = "  
if ( (int)$monat > 0 && (int)$jahr > 0 ) {  
  if ($jahr == $aktuelles_jahr && $monat == $aktueller_monat)  
    $aktuell_text = $this->pi_getLL('actual_month').': ': '  
  else {  
    $vormonat = $aktueller_monat - 1;  
    $vormonat_jahr = $aktuelles_jahr;  
    if ($vormonat == 0) {  
      $vormonat = 12;  
      $vormonat_jahr -= 1;  
    }  
    if ($jahr < $vormonat_jahr ||  
        ($jahr == $vormonat_jahr && $monat < $vormonat)) {  
      # Verbot für Monate vor dem Vormonat  
      $this->smarty->assign('ZEIT', $monate[$monat].'.$jahr);  
      $this->smarty->assign('AKTUELLES_JAHR', $aktuelles_jahr);  
      $this->smarty->assign('AKTUELLER_MONAT', $aktueller_monat);  
      $content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).  
        $this->templates['error']);  
      $res = $this->pi_wrapInBaseClass($content);  
      return $res;  
    }  
    if ($jahr == $vormonat_jahr && $monat == $vormonat)  
      $aktuell_text = $this->pi_getLL('previous_month').': ': '  
    else {  
      $folgemonat = $aktueller_monat + 1;  
      $folgemonat_jahr = $aktuelles_jahr;  
      if ($folgemonat == 13) {  
        $folgemonat = 1;  
        $folgemonat_jahr += 1;  
      }  
      if ($jahr == $folgemonat_jahr && $monat == $folgemonat)  
        $aktuell_text = $this->pi_getLL('next_month').': ': '  
    }  
  }  
} else {  
  switch($monat) {  
  case $this->pi_getLL('previous_month'):  
    if ($aktueller_monat == 1) {  
      $monat = 12;  
      $jahr = $aktuelles_jahr - 1;  
    } else {  
      $monat = $aktueller_monat - 1;  
      $jahr = $aktuelles_jahr;  
    }  
  }  
}
```

```

    $aktuell_text = $this->pi_getLL('previous_month').': ';
    break;
case $this->pi_getLL('next_month'):
    if ($aktueller_monat == 12) {
        $monat = 1;
        $jahr = $aktuelles_jahr + 1;
    } else {
        $monat = $aktueller_monat + 1;
        $jahr = $aktuelles_jahr;
    }
    $aktuell_text = $this->pi_getLL('next_month').': ';
    break;
default:
    $jahr = $aktuelles_jahr;
    $monat = $aktueller_monat;
    $aktuell_text = $this->pi_getLL('actual_month').': ';
}
}
}

```

Create display for the chosen month:

```

$termin_erster = new Termin('1.'.$smonat.'.'.$sjahr);
$stei = $termin_erster->int();
$max_tage = (string)($termin_erster->max_Tage_des_Monats());
$termin_letzter = new Termin($max_tage.'.'.$smonat.'.'.$sjahr);
$stli = $termin_letzter->int();
$stage_gebucht = array();
for ($si = 0; $si <= $max_tage; $si++) $stage_gebucht[$si] = 0;
$query = 'SELECT from_date,to_date ' .
    'FROM '.$stable_name.' ' .
    'WHERE (from_date>='.$stei.' AND from_date<='.$stli.') OR ' .
    '(to_date>='.$stei.' AND to_date<='.$stli.') OR ' .
    '(from_date<'.$stei.' AND to_date>'.$stli.');
$res = $GLOBALS['TYPO3_DB']->sql_query($query);
while ($row = $GLOBALS['TYPO3_DB']->sql_fetch_assoc($res)) {
    $v = new Termin($row['from_date']);
    $b = new Termin($row['to_date']);

    if ($v->is_true() && $b->is_true()) {
        $vi = $v->int();
        $vti = (int)$v->Tag();
        $bi = $b->int();
        $bti = (int)$b->Tag();
        if ($vi <= $stei && $bi >= $stli) {
            for ($si=1; $si <= $max_tage; $si++) $stage_gebucht[$si] = 1;
            break;
        } elseif ($vi <= $stei && $bi <= $stli) {
            for ($si=1; $si <= $bti; $si++)
                $stage_gebucht[$si] = 1;
        } elseif ($vi >= $stei && $bi <= $stli) {
            for ($si=$vti; $si <= $bti; $si++)
                $stage_gebucht[$si] = 1;
        }
    }
}

```

```

} elseif ($vi >= $tei && $bi > $tli) {
    for ($i=$vti; $i<=$max_tage; $i++)
        $stage_gebucht[$i] = 1;
    }
}
}

$ekw = $termin_erster->get_Kalenderwoche();
$ewz = $termin_erster->get_Wochentag_Zahl();
$lkw0 = $termin_letzter->get_Kalenderwoche();
if ($lkw0 == 1) $lkw = 53;
else $lkw = $lkw0;
$letzter_tag = $termin_letzter->Tag();

$a = array();
for ($j = 1; $j < $ewz; $j++) {
    $a[$ekw][$j] = 0;
}
$stag = 1;
for ($j = $ewz; $j <= 7; $j++) {
    $a[$ekw][$j] = $stag;
    $stag++;
}
for ($i = $ekw + 1; $i <= $lkw - 1; $i++) {
    for ($j = 1; $j <= 7; $j++) {
        $a[$i][$j] = $stag;
        $stag++;
    }
}
for ($j = 1; $stag <= $letzter_tag; $j++) {
    $a[$lkw][$j] = $stag;
    $stag++;
}
for ($j; $j <= 7; $j++) {
    $a[$lkw][$j] = 0;
}

$a2 = array();
for ($i = $ekw; $i <= $lkw; $i++) {
    if ($i == 53) $i0 = $lkw0;
    else $i0 = $i;
    $a2[$i][0] = '<tr><td align="center" style="background-color: #FFFFCC;">'. $i0. '</td>';
    for ($j = 1; $j <= 7; $j++) {
        if ($a[$i][$j] == 0) $a2[$i][$j] = '<td></td>';
        else {
            if ($stage_gebucht[$a[$i][$j]])
                $a2[$i][$j] = '<td align="center" style="background-color: LightGreen;">'. $a[$i][$j].
                    '<br />'. $this->pi_getLL("booked"). '</td>';
            else
                $a2[$i][$j] = '<td align="center" style="background-color: yellow;">'. $a[$i][$j]. '<br />'.
                    $this->pi_getLL("free"). '</td>';
        }
    }
}

```

```

    }
    $a2[$i][7] .= '</tr>';
}

$kalender_monat = "";
for ($i = $ekw; $i <= $lkw; $i++) {
    for ($j = 0; $j <= 7; $j++) {
        $kalender_monat .= $a2[$i][$j];
    }
}

$monatsradio = "";
for ($i = 1; $i <= 12; $i++) {
    $checked = "";
    if ($i == $aktueller_monat) $checked = ' checked="checked"';
    $monatsradio .= '
        <input name="Monat" type="radio" value="'. $i. '" .
        $checked.'>'. $monate[$i]. '</input>';
}

$this->smarty->assign('AKTUELL_TEXT', $aktuell_text);
$this->smarty->assign('MONAT', $monate[$monat]);
$this->smarty->assign('JAHR', $jahr);
$this->smarty->assign('BUCHUNGEN', $tag_str);
$this->smarty->assign('KALENDER_MONAT', $kalender_monat);
$this->smarty->assign('MONATSRADIO', $monatsradio);
$this->smarty->assign('AKTUELLES_JAHR', $aktuelles_jahr);

$content = $this->smarty->fetch(t3lib_extMgm::extPath($this->extKey).
    $this->templates['booked']);
$res = $this->pi_wrapInBaseClass($content);
return $res;

```

End of file:

```

    }
}
if (defined('TYPO3_MODE') && $TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
['ext/netcos_renting/pi2/class.tx_netcosrenting_pi2.php']) {
    include_once($TYPO3_CONF_VARS[TYPO3_MODE]['XCLASS']
['ext/netcos_renting/pi2/class.tx_netcosrenting_pi2.php']);
}
?>

```

5. HTML Templates

These templates exist by *smarty*. They are HTML files with additional abbreviation variables of *smarty*. According to our definition, the variables look like `{..$VALUE..}`. All template files are stored in the template directory we defined as `typo3conf/ext/netcos_renting/tmpl`. In this directory, we define two subdirectories *pi1* and *pi2* that store the templates for the different plugins.

5.1 First Plugin (Renting Administration)

The subdomain for the first plugin `tmpl/pi1` contains 8 *smarty* template files.

Administration.tmpl:

```
<html><head>
  {..$IMPORT_JSCALENDAR..}
</head><body>
  <h1>{..$ADMINISTRATION_HEADER..}</h1><br /><br />
  <h2>{..$ADMINISTRATION_NEW..}</h2><br />
  <p>{..$ADMINISTRATION_DATE_FORMAT..}</p><br />
  <form method="post">
    <input type="text" id="data_from" name="data_from" />
    <button id="trigger_from">{..$FROM..}</button>
    <script type="text/javascript">
      Calendar.setup(
        {
          inputField : "data_from",    // ID of the input field
          ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
          button     : "trigger_from"  // ID of the button
        }
      );
    </script>

    <input type="text" id="data_to" name="data_to" />
    <button id="trigger_to">{..$TO..}</button>
    <script type="text/javascript">
      Calendar.setup(
        {
          inputField : "data_to",    // ID of the input field
          ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
          button     : "trigger_to"  // ID of the button
        }
      );
    </script>
    <input type="submit" name="apply" value="{..$APPLY..}" />
  </form>
  <br />
  <h2>{..$ADMINISTRATION_YEARS..}</h2><br />
  <form method="post">
    <span>{..$YEAR..}</span>
    <input type="text" size="4" maxlength="4" name="jahr"
      value="{..$AKTUELLES_JAHR..}" />
    <input type="submit" name="wechseln" value="{..$CHANGE2..}" />
    <input type="submit" name="aktuell" value="{..$ACTUAL..}" />
  </form>
```

```
</form>
<br />
<h2>{..$BOOKINGS_TITLE..}</h2><br />
{..$AKTUELLE_BUCHUNGEN..}
</body></html>
```

exists.tmpl:

```
<html><head></head><body>
<h2>{..$EXISTS_HEADER..}</h2><br />
<form method="post">
  <td><input type="submit" name="more" value="{..$MORE..}" /></td>
</form>
</body></html>
```

overlap.tmpl:

```
<html><head>
{..$IMPORT_JSCALENDAR..}
</head><body>
<h2>{..$OVERLAP_HEADER..}</h2><br />
<table width="700px">
  <tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
  <th></th><th></th></tr>
  <tr><td><form method="post">
    <td><input type="hidden" name="uid" value="{..$UID1..}" />{..$UID1..}</td>
    <td><input type="text" id="data_from1"
      name="data_from" value="{..$VALUE_FROM1..}" />
    <button id="trigger_from1">{..$FROM..}</button>
    <script type="text/javascript">
      Calendar.setup(
      {
        inputField : "data_from1", // ID of the input field
        ifFormat : "{..$CALENDAR_IFFORMAT..}", // the date format
        button : "trigger_from1" // ID of the button
      }
    );
  </script></td>
    <td><input type="text" id="data_to1" name="data_to" value="{..$VALUE_TO1..}" />
    <button id="trigger_to1">{..$TO..}</button>
    <script type="text/javascript">
      Calendar.setup(
      {
        inputField : "data_to1", // ID of the input field
        ifFormat : "{..$CALENDAR_IFFORMAT..}", // the date format
        button : "trigger_to1" // ID of the button
      }
    );
  </script></td>
    <td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
    <td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
```

```

<td>{..$NEW..}</td>
</form></tr><tr><form method="post">
<td><input type="hidden" name="uid" value="{..$UID2..}" />{..$UID2..}</td>
<td><input type="text" id="data_from2" name="data_from"
value="{..$VALUE_FROM2..}" />
<button id="trigger_from2">{..$FROM..}</button>
<script type="text/javascript">
Calendar.setup(
{
inputField : "data_from2", // ID of the input field
ifFormat : "{..$CALENDAR_IFFORMAT..}", // the date format
button : "trigger_from2" // ID of the button
}
);
</script></td>

<td><input type="text" id="data_to2" name="data_to" value="{..$VALUE_TO2..}" />
<button id="trigger_to2">{..$TO..}</button>
<script type="text/javascript">
Calendar.setup(
{
inputField : "data_to2", // ID of the input field
ifFormat : "{..$CALENDAR_IFFORMAT..}", // the date format
button : "trigger_to2" // ID of the button
}
);
</script></td>
<td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
<td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
<td>{..$EXISTING..}</td>
</form></tr></table>
</body></html>

```

to_be_changed.tmpl:

```

<html><head>
{..$IMPORT_JSCALENDAR..}
</head><body>
<h2>{..$TO_BE_CHANGED_HEADER..}</h2><br />
<table width="700px">
<tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
<th></th><th></th></tr>
<form method="post">
<tr><td><input type="hidden" name="uid" value="{..$UID..}" />{..$UID..}</td>
<td><input type="text" id="data_from" name="data_from" value="{..$VALUE_FROM..}" />
<button id="trigger_from">{..$FROM..}</button>
<script type="text/javascript">
Calendar.setup(
{
inputField : "data_from", // ID of the input field
ifFormat : "{..$CALENDAR_IFFORMAT..}", // the date format
button : "trigger_from" // ID of the button

```

```

    }
  );
</script></td>
<td><input type="text" id="data_to" name="data_to" value="{..$VALUE_TO..}" />
<button id="trigger_to">{..$TO..}</button>
<script type="text/javascript">
  Calendar.setup(
  {
    inputField : "data_to",      // ID of the input field
    ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
    button     : "trigger_to"   // ID of the button
  }
  );
</script></td>
<td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
<td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
</tr></table>
</body></html>

```

to_be_deleted.tmpl:

```

<html><head></head><body>
<h2>{..$TO_BE_DELETED_HEADER..}</h2><br />
<table width="700px">
  <tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
  <th></th><th></th></tr>
  <tr align="center"><form method="post">
  <td><input type="hidden" name="uid" value="{..$UID..}">{..$UID..}</td>
  <td><input type="hidden" name="data_from"
  value="{..$VALUE_FROM..}" />{..$VALUE_FROM..}</td>
  <td><input type="hidden" name="data_to"
  value="{..$VALUE_TO..}" />{..$VALUE_TO..}</td>
  <td><input type="submit" name="delete" value="{..$DELETE..}" /></td>
  <td><input type="submit" name="more" value="{..$CANCEL..}" /></td>
  </form></tr></table>
</body></html>

```

was_created.tmpl:

```

<html><head></head><body>
<h2>{..$WAS_CREATED_HEADER..}</h2><br />
<form method="post">
  <td><input type="submit" name="more" value="{..$MORE..}" /></td>
</form>
</body></html>

```

was_deleted.tmpl:

```

<html><head></head><body>
<h2>{..$WAS_DELETED_HEADER..}</h2><br />
<form method="post">

```

```

    <td><input type="submit" name="more" value="{..$MORE..}" /></td>
</form>
</body></html>

```

wrong_entry.tpl:

```

<html><head>
  {..$IMPORT_JSCALENDAR..}
</head><body>
  <h2>{..$WRONG_ENTRY_HEADER..}</h2><br />
  <p>{..$ERRMSG..}</p><br />
  <table width="700px">
    <tr align="center"><th>UID</th><th>{..$FROM..}</th><th>{..$TO..}</th>
      <th></th><th></th></tr>
    <form method="post">
      <tr><td><input type="hidden" name="uid" value="{..$UID..}">{..$UID..}</td>
        <td><input type="text" id="data_from" name="data_from" value="{..$VALUE_FROM..}" />
        <button id="trigger_from">{..$FROM..}</button>
        <script type="text/javascript">
          Calendar.setup(
            {
              inputField : "data_from",      // ID of the input field
              ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
              button     : "trigger_from"    // ID of the button
            }
          );
        </script></td>

        <td><input type="text" id="data_to" name="data_to" value="{..$VALUE_TO..}" />
        <button id="trigger_to">{..$TO..}</button>
        <script type="text/javascript">
          Calendar.setup(
            {
              inputField : "data_to",      // ID of the input field
              ifFormat   : "{..$CALENDAR_IFFORMAT..}", // the date format
              button     : "trigger_to"    // ID of the button
            }
          );
        </script></td>
        <td><input type="submit" name="change" value="{..$CHANGE1..}" /></td>
        <td><input type="submit" name="to_be_deleted" value="{..$DELETE..}" /></td>
      </tr></table>
</body></html>

```

5.2 Second Plugin (rented)

The subdirectory for the second plugin *tmpl/pi2* contains 2 *smarty* template files.

booked.tpl:

```

<html><head>
</head><body>
<h1>{..$BOOKED_HEADER..}</h1>
<br />
<h2>{..$AKTUELL_TEXT..} {..$MONAT..} {..$JAHR..}</h2>
<br />
<table width="700px" style="border: 10px rgb(135,148,116) solid;">
  <tr align="center" style="background-color: #CCFFCC;">
    <th>{..$CALENDAR_WEEK..}</th>
    <th>{..$MONDAY..}</th><th>{..$TUESDAY..}</th><th>{..$WEDNESDAY..}</th>
    <th>{..$THURSDAY..}</th><th>{..$FRIDAY..}</th><th>{..$SATURDAY..}</th>
    <th>{..$SUNDAY..}</th></tr>
  {..$KALENDER_MONAT..}
</table><br />
<form method="post">
  <input type="submit" name="Monat" value="{..$PREVIOUS_MONTH..}" />
  <input type="submit" name="Monat" value="{..$ACTUAL_MONTH..}" />
  <input type="submit" name="Monat" value="{..$NEXT_MONTH..}" />
</form><br />
<form method="post">
<h3>{..$BOOKED_FUTURE..}</h3>
  {..$MONATSRADIO..}
  <br />
  <span>{..$YEAR..}</span>
  <input name="Jahr" size="4" maxlength="4" type="text"
  value="{..$AKTUELLES_JAHR..}" />
  <br />
  <input type="submit" name="wechseln" value="{..$BOOKED_CHANGE..}" />
</form>
</body></html>

```

error.tmpl:

```

<html><head>
</head><body>
<h2>{..$ERROR_HEADER..} {..$ZEIT..}</h2>
<br />
  {..$ERROR_TEXT..}
<br />
<br />
<form method="post">
  <input type="hidden" name="Monat" value="{..$AKTUELLER_MONAT..}" />
  <input type="hidden" name="Jahr" value="{..$AKTUELLES_JAHR..}" />
  <input type="submit" name="wechseln" value="{..$ERROR_BACK..}" />
</form>
</body></html>

```