PDF-form creation using pdflatex, hyperref and insdljs

Dirk Krause

February 7, 2008

Contents

1 Introduction **4**

1.1 Copyright . . . 4

1.2 Purpose of this tutorial . . . 4

2 hyperref and PDF forms **5**

3 Before we start **6**

3.1 Document preamble . . . 6

3.1.1 Document class . . . 6

3.1.2 Fonts . . . 6

3.1.3 Page layout . . . 6

3.1.4 Graphics and colors . . . 6

3.1.5 hyperref-package . . . 7

3.2 Form element attributes . . . 8

4 The first form **9**

4.1 Goal . . . 9

4.2 First attempt: ex0001 . . . 9

4.3 Aligning elements in a table: ex0002 . . . 11

4.4 Spaces and element widths: ex0003 . . . 13

4.5 Label text position: ex0004 . . . 15

4.6 Colors: ex0005 . . . 17

4.7 JavaScript: ex0006 . . . 19

4.8 Document events: ex0008 and ex0009 . . . 24

4.9 Form element events: ex0010 . . . 32

4.10 Editable choice list: ex0011 . . . 37

4.11 Adding help . . . 42

4.11.1 Overview . . . 42

4.11.2 PDF annotations . . . 42

4.11.3 Tooltips . . . 43

5 Another form **44**

5.1 Goal . . . 44

5.2 Initializing a choice list using JavaScript: ex0101 . . . 44

6 Form number three: Interactive application **51**

6.1 Goal . . . 51

6.2 Processing button click events: ex0201 . . . 51

7 Form number four: using both insdljs and inputenc **54**

A History **57**

B GNU Free Documentation License 58
B.1 APPLICABILITY AND DEFINITIONS 58
B.2 VERBATIM COPYING 60
B.3 COPYING IN QUANTITY 60
B.4 MODIFICATIONS 60
B.5 COMBINING DOCUMENTS 62
B.6 COLLECTIONS OF DOCUMENTS 63
B.7 AGGREGATION WITH INDEPENDENT WORKS 63
B.8 TRANSLATION 63
B.9 TERMINATION 63
B.10 FUTURE REVISIONS OF THIS LICENSE 64

1 Introduction

1.1 Copyright

1.2 Purpose of this tutorial

This tutorial shows how to create PDF forms using pdfLaTeX, hyperref and insdljs. The focus is set on forms which are filled interactively on screen and printed.

You need a LaTeX distribution, pdflatex, hyperref and insdljs must be installed. The insdljs package is a part of the AcroTeX Education Bundle, see `http://www.acrotex.net`.

You should have the following additional documentation available:

- the hyperref manual: [RO04],
- guides, specifications and references for Acrobat JavaScript: [Sup03], [ADS05a] and [ADS05b] and
- the PDF file format reference: [ASI01].

  This tutorial shows three examples, the first one is developed step-by-step. The final versions of the forms are "ex0011", "ex0101" and "ex0201".

2 hyperref and PDF forms

The hyperref package provides environments and macros to create forms. The form elements must be placed in a `Form` environment, `Form` environments are limited to one `Form` environment per document. The following macros can be used to create form elements:

- `\TextField[`*Parameter*`]{`*Text*`}`
  creates a text field.
- `\CheckBox[`*Parameter*`]{`*Text*`}`
  creates a check box.
- `\ChoiceMenu[`*Parameter*`]{`*Text*`}{`*Auswahl-Optionen*`}`
  creates a choice list.
- `\PushButton[`*Parameter*`]{`*Text*`}`
  creates a button.
- `\Submit[`*Parameter*`]{`*Text*`}`
  creates a Submit-Button. Pushing the button sends form data to the URL specified as action-parameter in the `Form` environment.
- `\Reset[`*Parameter*`]{`*Text*`}`
  creates a Reset-button.

The hyperref documentation [RO04] provides a more detailed description of the `Form` environment and the macros above.

3 Before we start

3.1 Document preamble

3.1.1 Document class

The usual document classes – i. e. scrartcl or article – can be used to create forms.

3.1.2 Fonts

The PDF standard requires each PDF viewer to render 14 fonts even if the font is not embedded into the PDF file, especially the font families Times, Helvetica and Courier. JavaScript code can be used to change form field fonts to these fonts. We want to use JavaScript to change fonts, so we load the font setup packages using

```
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
```

Additionally we use

```
\usepackage[T1]{fontenc}
```

to change the font encoding from OT1 (LaTeX default) to T1.
The default family is switched to Helvetica using

```
\renewcommand*{\familydefault}{\phv}
```

because sans-serif fonts are better readable on screen than serifed fonts.

3.1.3 Page layout

The page margins are set up by:

```
\usepackage[left=25mm, top=25mm, bottom=10mm, right=10mm]{geometry}
```

Page header and footer are set up by:

```
\usepackage{fancyhdr}
\lhead{Firma ABC}\chead{}\rhead{Abteilung XYZ}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
```

3.1.4 Graphics and colors

The packages

```
\usepackage{graphicx}
\usepackage{color}
```

can be used to add graphics (i. e. logos) and colors.

3.1.5 hyperref-package

The hyperref package is used:

```
\usepackage[
   pdftex,a4paper=true,colorlinks=true,
   pdftitle={Key form},
   pdfsubject={Key},
   pdfauthor={Front of House Manager},
   pdfpagemode=UseNone,pdfstartview=FitH,pdfhighlight={/N}
]{hyperref}
```

3.2 Form element attributes

I recommend to set form elements colors and fonts as shown in table 1.

Table 1: Form element colors and fonts

	Border color	**Fill color**	**Font weight**
Screen	colored	colored	bold
Printout	black	white	regular
Screen and Printout[1]	colored	white	regular

All form elements should be marked fully colored when shown on screen so users can find all elements easily. For readability we choose a bold font to show text on a colored background.

Printouts of forms are typically done on black-and-white printers. On printout we want a black border and white filling for form elements. The text is black on white now, so we do not need a bold font.

JavaScript must be enabled in the PDF viewer to change colors and fonts before and after printing the document.

The color for marking form elements should be the same color for all elements. We want to show black text on color-filled areas, so we should use a light color for a good contrast.

---

[1] if JavaScript is deactivated

4 The first form

4.1 Goal

We want to design a form which can be used to request keys for new empolyees. The following data is needed:

- First name and name,
- Department,
- Employment time (unlimited or limited from … until …),
- the doors to open.

Text fields are used for first name and name. Department and employment time are choosen from a list. The doors are selected in check boxes.

4.2 First attempt: ex0001

The ex0001.tex file shows a first attempt to create the form.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\begin{document}
\section*{Request for a key}
\begin{Form}
```

```
\textbf{Employee}\\
First name: \TextField[name=vorname]{}\\
Name: \TextField[name=name]{}\\
Department: \ChoiceMenu[name=abt]{}{%
Sales=v,Production=f,Service=s}\\
\textbf{Time}\\
Time: \ChoiceMenu[name=zeit]{}{%
limited=b,unlimited=u}\\
from: \TextField[name=from]{}\\
until: \TextField[name=until]{}\\
\textbf{Doors}\\
Front door: \CheckBox[name=ht]{}\\
Ground floor: \CheckBox[name=e1]{}\\
First floor: \CheckBox[name=e2]{}\\
\end{Form}
\end{document}
```

After running

```
pdflatex ex0001 && pdflatex ex0001
```

we can open ex0001.pdf in the PDF viewer.
The form does not look good but contains the required functionality.

The source shows:

- A unique name (identifier) is assigned to each form element using the *name* parameter.
- The last argument of the *ChoiceMenu[]{}{}* macro is a list containing the available choices. Each list entry contains the full text to show and an abbreviation, separated by "=".

4.3 Aligning elements in a table: ex0002

We use a *tabular* environment to align the form elements in a table.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
   pdftex,a4paper=true,colorlinks=true,
    pdftitle={Key form},pdfsubject={Key},
   pdfauthor={ich},pdfpagemode=UseNone,pdfstartview=FitH,
   pagebackref,pdfhighlight={/N}
]{hyperref}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
First name:&%
\TextField[name=vorname]{}\\
Name:&%
\TextField[name=name]{}\\
Department:&%
\ChoiceMenu[name=abt]{}{%
Sales=v,Production=f,Service=s}\\
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
Time:&%
\ChoiceMenu[name=zeit]{}{%
limited=b,unlimited=u}\\
from:&%
\TextField[name=from]{}\\
until:&%
\TextField[name=until]{}\\
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Doors}}\\
Front door:&%
\CheckBox[name=ht, width=1.2em]{}\\
Ground floor:&%
\CheckBox[name=e1, width=1.2em]{}\\
First floor:&%
\CheckBox[name=e2, width=1.2em]{}\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.4 Spaces and element widths: ex0003

We use

```
&&\\*[-0.9em]
```

to add vertical space between the table lines. In the last table line we use

```
&&\\*
```

In the *ChoiceMenu* we set the “popdown=true” parameter to establish popdown-behaviour. The parameter “width=20em” parameter sets the horizontal size.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm, top=25mm, bottom=10mm, right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]First name:&%
\TextField[name=vorname,width=20em]{}\\
&\\*[-0.9em]Name:&%
\TextField[name=name,width=20em]{}\\
&\\*[-0.9em]Department:&%
\ChoiceMenu[{}={},name=abt,width=20em,popdown=true]{}{%
Sales=v,Production=f,Service=s}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]Time:&%
\ChoiceMenu[name=zeit,width=20em,popdown=true]{}{%
limited=b,unlimited=u}\\
&\\*[-0.9em]from:&%
\TextField[name=from,width=10em]{}\\
&\\*[-0.9em]until:&%
\TextField[name=until,width=10em]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Doors}}\\
&\\*[-0.9em]Front door:&%
\CheckBox[name=ht,width=1.2em]{}\\
&\\*[-0.9em]Ground floor:&%
\CheckBox[name=e1,width=1.2em]{}\\
&\\*[-0.9em]First floor:&%
\CheckBox[name=e2,width=1.2em]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.5 Label text position: ex0004

The *y*-position of form element texts and label texts differ because the form elements have an inner and outer border.

We use text fields to show the labels and make the borders invisible. So label text and form element text is at the same *y*-position now.

We create a white border for the labels

```
bordercolor={1 1 1}
```

and set text alignment to right-aligned:

```
align=2
```

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm, top=25mm, bottom=10mm, right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,%
align=2,bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
```

```
\TextField[name=vorname,width=20em]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name,width=20em]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
\ChoiceMenu[name=abt,width=20em,popdown=true,value=v]{}{%
Sales=v,Production=f,Service=s}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit,width=20em,popdown=true,value=u]{}{%
unlimited=u,limited=b}\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from,width=10em]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until,width=10em]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht,width=1.2em]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
\CheckBox[name=e1,width=1.2em]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2,width=1.2em]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.6 Colors: ex0005

We use

```
bordercolor={0.65 0.79 0.94}
```

to change the border color to light blue for the form elements.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,pagebackref,pdfhighlight={/N}
]{hyperref}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
\TextField[name=vorname,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
```

```
\ChoiceMenu[name=abt, width=20em, popdown=true, value=v,%
bordercolor={0.65 0.79 0.94}]{}{%
Sales=v, Production=f, Service=s}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit, width=20em, popdown=true, value=u,%
bordercolor={0.65 0.79 0.94}]{}{%
unlimited=u, limited=b}\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{%
\textbf{Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
\CheckBox[name=e1, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.7 JavaScript: ex0006

The insdljs package is used to add document level JavaScript. The package requires to specify the output driver.

```
\usepackage[pdftex]{insdljs}
```

The JavaScript-code to add is placed in an *insDLJS* environment. The optional argument and the first required argument must be LaTeX identifiers (use a string containing characters only, no numbers). The second required argument is a comment

```
\begin{insDLJS}[exaaaa]{exaaaa}{Document Level JavaScript}
// ... Place Java source code here ...
\end{insDLJS}
```

The identifier is used to build a file name (“exaaaa.djs”) to temporarily store the packed JavaScript code.

The *OpenAction{}* macro specifies a function which is executed when the document is opened.

```
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
```

The JavaScript-code is enclosed in brackets. The example code contains a function invokation.

The function may be invoked multiple times (each time one navigates to page 1). We use a variable as a flag indicating whether the function is run for the first time or not.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
```

```
\usepackage[pdftex]{insdljs}
\begin{insDLJS}[exaaaa]{exaaaa}{Document Level JavaScript}

// indicate that the function was not yet run
var done = 0;

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    app.alert("The form was opened.");
  }
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
\TextField[name=vorname,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
\ChoiceMenu[name=abt,width=20em,popdown=true,value=v,%
bordercolor={0.65 0.79 0.94}]{}{%
Sales=v,Production=f,Service=s}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit,width=20em,popdown=true,value=u,%
bordercolor={0.65 0.79 0.94}]{}{%
unlimited=u,limited=b}\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from,width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until,width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{%
Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
\CheckBox[name=e1,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

A message box is shown when the document is opened. In the german version the character ö is malformed to “¨o” because the inputenc package replaces german umlauts (and other characters) by the corresponding LaTeX code sequence. There are two ways to solve the problem:

- Remove the inputenc package and convert german umlauts to their LaTeX code sequence manually (shown in the listing below at the end of this section) or
- use JavaScript-Unicode notation for umlauts in JavaScript instead of typing the umlauts directly (shown in section 7 on page 54).

```
\usepackage[latin1]{inputenc}
```

from our source.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
```

```
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\usepackage[pdftex]{insdljs}
\begin{insDLJS}[exaaab]{exaaab}{Document Level JavaScript}

// indicate that the function was not yet run
var done = 0;

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    app.alert("The form was opened.");
  }
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
\TextField[name=vorname,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
```

```
\ChoiceMenu[name=abt,width=20em,popdown=true,value=v,%
bordercolor={0.65 0.79 0.94}]{}{%
Sales=v,Production=f,Service=s}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit,width=20em,popdown=true,value=u,%
bordercolor={0.65 0.79 0.94}]{}{%
unlimited=u,limited=b}\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from,width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until,width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{%
\textbf{Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
\CheckBox[name=e1,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.8 Document events: ex0008 and ex0009

To process document events we have to add an /AA entry to the PDF catalogue. The entry value must be a dictionary defining event handlers. Entries in this dictionary are key/value pairs, the value is again a dictionary specifying the action to take.
The following keys can occur:

Key	Event
/WP	The document will be printed.
/DP	The document was printed.
/WS	The document will be saved.
/DS	The document was saved.
/DC	The document will be closed.

In the example we run *willPrint();* for the WillPrint event and *didPrint();* for the DidPrint event.

```
\immediate\pdfobj{
<<
  /WP << /S /JavaScript /JS (willPrint();) >>
  /DP << /S /JavaScript /JS (didPrint();) >>
>>
}
\pdfcatalog{/AA \the\pdflastobj\space 0 R}
```

Before and after printing ex0008.pdf a message box is shown.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\usepackage[pdftex]{insdljs}
\immediate\pdfobj {
<<
  /WP << /S /JavaScript /JS (willPrint();) >>
  /DP << /S /JavaScript /JS (didPrint();) >>
>>
}
\pdfcatalog{/AA \the\pdflastobj\space 0 R}
\begin{insDLJS}[exaaac]{exaaac}{Document Level JavaScript}

// indicate that the function was not yet run
var done = 0;

function willPrint()
{
  app.alert("The document will be printed.");
}

function didPrint()
{
  app.alert("The document was printed.");
}

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    app.alert("The form was opened.");
  }
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
```

```
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
\TextField[name=vorname, width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name, width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
\ChoiceMenu[name=abt, width=20em,popdown=true, value=v,%
bordercolor={0.65 0.79 0.94}]{}{%
Sales=v,Production=f,Service=s}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit, width=20em,popdown=true, value=u,%
bordercolor={0.65 0.79 0.94}]{}{%
unlimited=u,limited=b}\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{%
\textbf{Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
\CheckBox[name=e1, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

In ex0009.pdf we change stroke and fill colors and text fonts before and after printing. We have to change the *fillColor*, *strokeColor* and *textFont* attributes.
The changes are enclosed in

```
   this.delay = true
   // ...
   this.delay = false
```

to avoid multiple screen refreshes after each change.

In *myFirstJavaScriptFunction()* we set defaults for the choice lists *abt* and *zeit* using:

```
this.getField("abt").currentValueIndices = 0;
this.getField("zeit").currentValueIndices = 0;
```

We set the focus to the *vorname* field by:

```
this.getField("vorname").setFocus();
```

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
   pdftex,a4paper=true,colorlinks=true,
   pdftitle={Key form},pdfsubject={Key},
   pdfauthor={ich},
   pdfpagemode=UseNone,pdfstartview=FitH,
   pagebackref,pdfhighlight={/N}
]{hyperref}
\usepackage[pdftex]{insdljs}
\immediate\pdfobj {
<<
   /WP << /S /JavaScript /JS (willPrint();) >>
   /DP << /S /JavaScript /JS (didPrint();) >>
>>
}
\pdfcatalog{/AA \the\pdflastobj\space 0 R}
\begin{insDLJS}[exaaad]{exaaad}{Document Level JavaScript}

// indicate that the function was not yet run
```

```
var done = 0;

function colorsPrinting()
{
  this.getField("vorname").fillColor = color.white;
  this.getField("name").fillColor = color.white;
  this.getField("abt").fillColor = color.white;
  this.getField("zeit").fillColor = color.white;
  this.getField("from").fillColor = color.white;
  this.getField("until").fillColor = color.white;
  this.getField("ht").fillColor = color.white;
  this.getField("e1").fillColor = color.white;
  this.getField("e2").fillColor = color.white;

  this.getField("vorname").strokeColor = color.black;
  this.getField("name").strokeColor = color.black;
  this.getField("abt").strokeColor = color.black;
  this.getField("zeit").strokeColor = color.black;
  this.getField("from").strokeColor = color.black;
  this.getField("until").strokeColor = color.black;
  this.getField("ht").strokeColor = color.black;
  this.getField("e1").strokeColor = color.black;
  this.getField("e2").strokeColor = color.black;

  this.getField("vorname").textFont = font.Helv;
  this.getField("name").textFont = font.Helv;
  this.getField("abt").textFont = font.Helv;
  this.getField("zeit").textFont = font.Helv;
  this.getField("from").textFont = font.Helv;
  this.getField("until").textFont = font.Helv;
}

function colorsScreen()
{

  var newcolor = [ "RGB", 0.65, 0.79, 0.94];
  this.getField("vorname").fillColor = newcolor;
  this.getField("name").fillColor = newcolor;
  this.getField("abt").fillColor = newcolor;
  this.getField("zeit").fillColor = newcolor;
  this.getField("from").fillColor = newcolor;
  this.getField("until").fillColor = newcolor;
  this.getField("ht").fillColor = newcolor;
  this.getField("e1").fillColor = newcolor;
  this.getField("e2").fillColor = newcolor;

  this.getField("vorname").strokeColor = newcolor;
  this.getField("name").strokeColor = newcolor;
  this.getField("abt").strokeColor = newcolor;
  this.getField("zeit").strokeColor = newcolor;
  this.getField("from").strokeColor = newcolor;
  this.getField("until").strokeColor = newcolor;
  this.getField("ht").strokeColor = newcolor;
  this.getField("e1").strokeColor = newcolor;
  this.getField("e2").strokeColor = newcolor;

  this.getField("vorname").textFont = font.HelvB;
  this.getField("name").textFont = font.HelvB;
  this.getField("abt").textFont = font.HelvB;
  this.getField("zeit").textFont = font.HelvB;
  this.getField("from").textFont = font.HelvB;
  this.getField("until").textFont = font.HelvB;
}

function willPrint()
{
  this.delay = true;
  colorsPrinting();
  this.delay = false;
}

function didPrint()
{
  this.delay = true;
  colorsScreen();
  this.delay = false;
}

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    this.delay = true;
    colorsScreen();
    this.getField("abt").currentValueIndices = 0;
    this.getField("zeit").currentValueIndices = 0;
    this.getField("vorname").setFocus();
```

```
      this.delay = false;
    }
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
\TextField[name=vorname,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name,width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
\ChoiceMenu[name=abt,width=20em,popdown=true,value=v,%
bordercolor={0.65 0.79 0.94}]{}{%
Sales=v,Production=f,Service=s}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit,width=20em,popdown=true,value=u,%
bordercolor={0.65 0.79 0.94}]{}{%
unlimited=u,limited=b}\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from,width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until,width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{%
\textbf{Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
```

```
\CheckBox[name=e1,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2,width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.9 Form element events: ex0010

Events can be triggered by form elements, in the example we assign the function *validateTime()* as an event handler for the Validate event of the *zeit* choice list.

```
validate={validateTime();}
```

If the user leaves the choice field the function is invoked and turns the “from” and “until” text fields visible

```
this.getField("von").display = display.visible;
this.getField("bis").display = display.visible;
```

or invisible.

```
this.getField("von").display = display.hidden;
this.getField("bis").display = display.hidden;
```

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\usepackage[pdftex]{insdljs}
\immediate\pdfobj {
<<
  /WP << /S /JavaScript /JS (willPrint();) >>
  /DP << /S /JavaScript /JS (didPrint();) >>
>>
}
\pdfcatalog{/AA \the\pdflastobj\space 0 R}
```

```
\begin{insDLJS}[exaaae]{exaaae}{Document Level JavaScript}

// indicate that the function was not yet run
var done = 0;

function colorsPrinting()
{
  this.getField("vorname").fillColor = color.white;
  this.getField("name").fillColor = color.white;
  this.getField("abt").fillColor = color.white;
  this.getField("zeit").fillColor = color.white;
  this.getField("from").fillColor = color.white;
  this.getField("until").fillColor = color.white;
  this.getField("ht").fillColor = color.white;
  this.getField("e1").fillColor = color.white;
  this.getField("e2").fillColor = color.white;

  this.getField("vorname").strokeColor = color.black;
  this.getField("name").strokeColor = color.black;
  this.getField("abt").strokeColor = color.black;
  this.getField("zeit").strokeColor = color.black;
  this.getField("from").strokeColor = color.black;
  this.getField("until").strokeColor = color.black;
  this.getField("ht").strokeColor = color.black;
  this.getField("e1").strokeColor = color.black;
  this.getField("e2").strokeColor = color.black;

  this.getField("vorname").textFont = font.Helv;
  this.getField("name").textFont = font.Helv;
  this.getField("abt").textFont = font.Helv;
  this.getField("zeit").textFont = font.Helv;
  this.getField("from").textFont = font.Helv;
  this.getField("until").textFont = font.Helv;
  this.getField("from").textColor = color.black;
  this.getField("until").textColor = color.black;
}

function colorsScreen()
{

  var newcolor = [ "RGB", 0.65, 0.79, 0.94];
  this.getField("vorname").fillColor = newcolor;
  this.getField("name").fillColor = newcolor;
  this.getField("abt").fillColor = newcolor;
  this.getField("zeit").fillColor = newcolor;
  this.getField("from").fillColor = newcolor;
  this.getField("until").fillColor = newcolor;
  this.getField("ht").fillColor = newcolor;
  this.getField("e1").fillColor = newcolor;
  this.getField("e2").fillColor = newcolor;

  this.getField("vorname").strokeColor = newcolor;
  this.getField("name").strokeColor = newcolor;
  this.getField("abt").strokeColor = newcolor;
  this.getField("zeit").strokeColor = newcolor;
  this.getField("from").strokeColor = newcolor;
  this.getField("until").strokeColor = newcolor;
  this.getField("ht").strokeColor = newcolor;
  this.getField("e1").strokeColor = newcolor;
  this.getField("e2").strokeColor = newcolor;

  this.getField("vorname").textFont = font.HelvB;
  this.getField("name").textFont = font.HelvB;
  this.getField("abt").textFont = font.HelvB;
  this.getField("zeit").textFont = font.HelvB;
  this.getField("from").textFont = font.HelvB;
  this.getField("from").textColor = color.black;
  this.getField("until").textColor = color.black;
  this.getField("until").textFont = font.HelvB;
}

function validateTime()
{
  this.delay = true;
  if((event.value == "u") || (event.value == "unlimited")) {
    this.getField("from").display = display.hidden;
    this.getField("until").display = display.hidden;
  } else {
    this.getField("from").display = display.visible;
    this.getField("until").display = display.visible;
    this.getField("from").setFocus();
  }
  this.delay = false;
}

function willPrint()
{
  this.delay = true;
  colorsPrinting();
  this.delay = false;
}

function didPrint()
{
  this.delay = true;
  colorsScreen();
  this.delay = false;
}

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    this.delay = true;
    colorsScreen();
    this.getField("abt").currentValueIndices = 0;
    this.getField("zeit").currentValueIndices = 0;
    this.getField("vorname").setFocus();
    this.delay = false;
  }
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
\TextField[name=vorname, width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name, width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
\ChoiceMenu[name=abt, width=20em,popdown=true, value=v,%
```

```
bordercolor={0.65 0.79 0.94}]{}{%
Sales=v, Production=f, Service=s }\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit, width=20em, popdown=true, value=u,%
validate={validateTime();},%
bordercolor={0.65 0.79 0.94}]{}{%
unlimited=u, limited=b }\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{%
\textbf{Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
\CheckBox[name=e1, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.10 Editable choice list: ex0011

In the choice list *abt* we want to either choose an entry from the list or add text.
The parameter

```
combo=true
```

activates combo box hehaviour.
Additionally we add a JavaScript code line

```
this.getField("abt").editable = true;
```

into our initialization function *myFirstJavaScriptFunction()*.

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Key form},pdfsubject={Key},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\usepackage[pdftex]{insdljs}
\immediate\pdfobj {
<<
  /WP << /S /JavaScript /JS (willPrint();) >>
  /DP << /S /JavaScript /JS (didPrint();) >>
>>
}
\pdfcatalog{/AA \the\pdflastobj\space 0 R}
\begin{insDLJS}[exaaae]{exaaae}{Document Level JavaScript}

// indicate that the function was not yet run
var done = 0;
```

```

function colorsPrinting()
{
  this.getField("vorname").fillColor = color.white;
  this.getField("name").fillColor = color.white;
  this.getField("abt").fillColor = color.white;
  this.getField("zeit").fillColor = color.white;
  this.getField("from").fillColor = color.white;
  this.getField("until").fillColor = color.white;
  this.getField("ht").fillColor = color.white;
  this.getField("e1").fillColor = color.white;
  this.getField("e2").fillColor = color.white;

  this.getField("vorname").strokeColor = color.black;
  this.getField("name").strokeColor = color.black;
  this.getField("abt").strokeColor = color.black;
  this.getField("zeit").strokeColor = color.black;
  this.getField("from").strokeColor = color.black;
  this.getField("until").strokeColor = color.black;
  this.getField("ht").strokeColor = color.black;
  this.getField("e1").strokeColor = color.black;
  this.getField("e2").strokeColor = color.black;

  this.getField("vorname").textFont = font.Helv;
  this.getField("name").textFont = font.Helv;
  this.getField("abt").textFont = font.Helv;
  this.getField("zeit").textFont = font.Helv;
  this.getField("from").textFont = font.Helv;
  this.getField("until").textFont = font.Helv;
  this.getField("from").textColor = color.black;
  this.getField("until").textColor = color.black;
}

function colorsScreen()
{

  var newcolor = [ "RGB", 0.65, 0.79, 0.94];
  this.getField("vorname").fillColor = newcolor;
  this.getField("name").fillColor = newcolor;
  this.getField("abt").fillColor = newcolor;
  this.getField("zeit").fillColor = newcolor;
  this.getField("from").fillColor = newcolor;
  this.getField("until").fillColor = newcolor;
  this.getField("ht").fillColor = newcolor;
  this.getField("e1").fillColor = newcolor;
  this.getField("e2").fillColor = newcolor;

  this.getField("vorname").strokeColor = newcolor;
  this.getField("name").strokeColor = newcolor;
  this.getField("abt").strokeColor = newcolor;
  this.getField("zeit").strokeColor = newcolor;
  this.getField("from").strokeColor = newcolor;
  this.getField("until").strokeColor = newcolor;
  this.getField("ht").strokeColor = newcolor;
  this.getField("e1").strokeColor = newcolor;
  this.getField("e2").strokeColor = newcolor;

  this.getField("vorname").textFont = font.HelvB;
  this.getField("name").textFont = font.HelvB;
  this.getField("abt").textFont = font.HelvB;
  this.getField("zeit").textFont = font.HelvB;
  this.getField("from").textFont = font.HelvB;
  this.getField("from").textColor = color.black;
  this.getField("until").textColor = color.black;
  this.getField("until").textFont = font.HelvB;
}

function validateTime()
{
  this.delay = true;
  if((event.value == "u") || (event.value == "unlimited")) {
    this.getField("from").display = display.hidden;
    this.getField("until").display = display.hidden;
  } else {
    this.getField("from").display = display.visible;
    this.getField("until").display = display.visible;
    this.getField("from").setFocus();
  }
  this.delay = false;
}

function willPrint()
{
  this.delay = true;
  colorsPrinting();
  this.delay = false;
}

```

```
function didPrint()
{
  this.delay = true;
  colorsScreen();
  this.delay = false;
}

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    this.delay = true;
    colorsScreen();
    this.getField("abt").currentValueIndices = 0;
    this.getField("abt").editable = true;
    this.getField("zeit").currentValueIndices = 0;
    this.getField("vorname").setFocus();
    this.delay = false;
  }
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for a key}
\begin{Form}
\begin{tabular}{|rl|}
\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Employee}}\\
&\\*[-0.9em]\textforlabel{vn}{First name:}&%
\TextField[name=vorname, width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{nn}{Name:}&%
\TextField[name=name, width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{ab}{Department:}&%
\ChoiceMenu[name=abt, width=20em,popdown=true,combo=true,value=v,%
bordercolor={0.65 0.79 0.94}]{}{%
Sales=v,Production=f,Service=s}\\
&\\*[-0.9em]\hline
```

```
&\\*[-0.9em]\multicolumn{2}{|c|}{\textbf{Time}}\\
&\\*[-0.9em]\textforlabel{z}{Time:}&%
\ChoiceMenu[name=zeit, width=20em, popdown=true, value=u,%
validate={validateTime();},%
bordercolor={0.65 0.79 0.94}]{}{%
unlimited=u, limited=b}\\
&\\*[-0.9em]\textforlabel{v}{from:}&%
\TextField[name=from, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{b}{until:}&%
\TextField[name=until, width=10em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
&\\*[-0.9em]\multicolumn{2}{|c|}{%
\textbf{Doors}}\\
&\\*[-0.9em]\textforlabel{th}{Front door:}&%
\CheckBox[name=ht, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t1}{Ground floor:}&%
\CheckBox[name=e1, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{t2}{First floor:}&%
\CheckBox[name=e2, width=1.2em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

4.11 Adding help

4.11.1 Overview

Online help can be provided on different ways:

- PDF annotations
  Annotations are represented using icons. If the user moves the mouse over an annotation icon, the annotation text is shown in a tooltip. If the user double-clicks the annotation, the note is shown on the screen.
  Advantage: Annotations do not need JavaScript.
  Disadvantage: Place is needed for the icons.

- tooltips
  Tooltip text is shown if the user moves the mouse over a form field. Most users know the concept of tooltips from other applications.
  Tooltip text can not (yet?[2]) be set using hyperrefs form field options, so we use JavaScript instructions to set up tooltips.

4.11.2 PDF annotations

To create a help annotation use

```
~\pdfannot width 20pt {
        /Subtype /Text
        /Name /Help
        /C [0.65 0.79 0.94]
        /T (Help text title)
        /Contents (The help text to show.)
}\(\qquad\)
```

No room is reserved by pdfLaTeX for the icon, so we add some space (qquad). To have some room between the form field and the help icon we use a non-breaking space.

The "/Name /Help" entry chooses the annotation icon (question mark).
The "/C" entry sets the icon color, I recommend to use the same color as used in the form fields.

Example ex0012 shows how to use PDF annotations to provide help.

---

[2] 2008-02-07

4.11.3 Tooltips

Use

```
this.getField("fieldname").userName = "Help text here";
```

to set up tooltip texts. The instruction should be run in the function executed automatically when the document is opened.

Example ex0013 shows how to set up tooltip texts.

If you use the “inputenc” package you have to use special notations in JavaScript code to express foreign characters (i. e. german umlauts), see 7 on page 54.

5 Another form

5.1 Goal

We want to create a form to require software licenses and installation.

Information about the PC and the owner are entered in text fields. The software is choosen from choice lists. We have several lists providing the same choices. The choice lists are combo boxes so users can manually type in names of special software applications if necessary.

5.2 Initializing a choice list using JavaScript: ex0101

The choice lists can be initialized using JavaScript instead of writing the choices in the *\ChoiceMenu[]{}{}* macro.
This provides a better overview.

Choices are specified in an array. Each array element is an array again, consisting of two entries: the full text shown on screen and an abbreviation:

```
var softwareAuswahl = [
  [ "", ""],
  [ "Betriebssystem", "os" ],
  [ "Textverarbeitung", "tp" ],
  [ "Tabellenkalkulation", "sc" ],
  [ "Präsentationsprogramm", "pp" ],
  [ "E-Mail-Client", "mc" ],
  [ "Web browser", "wb" ],
  [ "Mathematik-Software", "ms" ],
  [ "PDF-Anzeigeprogramm", "pv" ]
];
```

The *setItems()* method sets the choices for a choice list:

```
this.getField("sw01").setItems(softwareAuswahl);
```

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm, top=25mm, bottom=10mm, right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{ABC, Inc.}\chead{}\rhead{XYZ Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
```

```
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={Software installation request},pdfsubject={Software},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\usepackage[pdftex]{insdljs}
\immediate\pdfobj {
<<
  /WP << /S /JavaScript /JS (willPrint();) >>
  /DP << /S /JavaScript /JS (didPrint();) >>
>>
}
\pdfcatalog{/AA \the\pdflastobj\space 0 R}
\begin{insDLJS}[exaaba]{exaaba}{Document Level JavaScript}

// indicate that the function was not yet run
var done = 0;

function colorsPrinting()
{
  this.getField("vorname").fillColor = color.white;
  this.getField("vorname").strokeColor = color.black;
  this.getField("vorname").textFont = font.Helv;
  this.getField("name").fillColor = color.white;
  this.getField("name").strokeColor = color.black;
  this.getField("name").textFont = font.Helv;
  this.getField("inventory").fillColor = color.white;
  this.getField("inventory").strokeColor = color.black;
  this.getField("inventory").textFont = font.Helv;
  this.getField("sw01").fillColor = color.white;
  this.getField("sw01").strokeColor = color.black;
  this.getField("sw01").textFont = font.Helv;
  this.getField("sw02").fillColor = color.white;
  this.getField("sw02").strokeColor = color.black;
  this.getField("sw02").textFont = font.Helv;
  this.getField("sw03").fillColor = color.white;
  this.getField("sw03").strokeColor = color.black;
  this.getField("sw03").textFont = font.Helv;
  this.getField("sw04").fillColor = color.white;
  this.getField("sw04").strokeColor = color.black;
  this.getField("sw04").textFont = font.Helv;
  this.getField("sw05").fillColor = color.white;
  this.getField("sw05").strokeColor = color.black;
  this.getField("sw05").textFont = font.Helv;
  this.getField("sw06").fillColor = color.white;
  this.getField("sw06").strokeColor = color.black;
  this.getField("sw06").textFont = font.Helv;
  this.getField("sw07").fillColor = color.white;
  this.getField("sw07").strokeColor = color.black;
  this.getField("sw07").textFont = font.Helv;
  this.getField("sw08").fillColor = color.white;
  this.getField("sw08").strokeColor = color.black;
  this.getField("sw08").textFont = font.Helv;
  this.getField("sw09").fillColor = color.white;
  this.getField("sw09").strokeColor = color.black;
  this.getField("sw09").textFont = font.Helv;
  this.getField("sw10").fillColor = color.white;
  this.getField("sw10").strokeColor = color.black;
  this.getField("sw10").textFont = font.Helv;
}

function colorsScreen()
{

  var newcolor = [ "RGB", 0.65, 0.79, 0.94];
  this.getField("vorname").fillColor = newcolor;
  this.getField("vorname").strokeColor = newcolor;
  this.getField("vorname").textFont = font.HelvB;
  this.getField("name").fillColor = newcolor;
  this.getField("name").strokeColor = newcolor;
  this.getField("name").textFont = font.HelvB;
  this.getField("inventory").fillColor = newcolor;
  this.getField("inventory").strokeColor = newcolor;
  this.getField("inventory").textFont = font.HelvB;
  this.getField("sw01").fillColor = newcolor;
  this.getField("sw01").strokeColor = newcolor;
  this.getField("sw01").textFont = font.HelvB;
  this.getField("sw02").fillColor = newcolor;
  this.getField("sw02").strokeColor = newcolor;
  this.getField("sw02").textFont = font.HelvB;
  this.getField("sw03").fillColor = newcolor;
  this.getField("sw03").strokeColor = newcolor;
  this.getField("sw03").textFont = font.HelvB;
  this.getField("sw04").fillColor = newcolor;
  this.getField("sw04").strokeColor = newcolor;
  this.getField("sw04").textFont = font.HelvB;
  this.getField("sw05").fillColor = newcolor;
  this.getField("sw05").strokeColor = newcolor;
  this.getField("sw05").textFont = font.HelvB;
  this.getField("sw06").fillColor = newcolor;
  this.getField("sw06").strokeColor = newcolor;
  this.getField("sw06").textFont = font.HelvB;
  this.getField("sw07").fillColor = newcolor;
  this.getField("sw07").strokeColor = newcolor;
  this.getField("sw07").textFont = font.HelvB;
  this.getField("sw08").fillColor = newcolor;
  this.getField("sw08").strokeColor = newcolor;
  this.getField("sw08").textFont = font.HelvB;
  this.getField("sw09").fillColor = newcolor;
  this.getField("sw09").strokeColor = newcolor;
  this.getField("sw09").textFont = font.HelvB;
  this.getField("sw10").fillColor = newcolor;
  this.getField("sw10").strokeColor = newcolor;
  this.getField("sw10").textFont = font.HelvB;
}

function willPrint()
{
  this.delay = true;
  colorsPrinting();
  this.delay = false;
}

function didPrint()
{
  this.delay = true;
  colorsScreen();
  this.delay = false;
}

var softwareAuswahl = [
  [ "", ""],
  [ "Operating system", "os" ],
  [ "Text processing", "tp" ],
  [ "Spreadsheet calculation", "sc" ],
  [ "Präsentationsprogramm", "pp" ],
  [ "E–mail client", "mc" ],
  [ "Web browser", "wb" ],
  [ "Mathematics software", "ms" ],
  [ "PDF viewer", "pv" ]
];

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    this.delay = true;
    colorsScreen();
    this.getField("sw01").setItems(softwareAuswahl);
    this.getField("sw01").currentValueIndices = 0;
    this.getField("sw01").editable = true;
    this.getField("sw02").setItems(softwareAuswahl);
    this.getField("sw02").currentValueIndices = 0;
    this.getField("sw02").editable = true;
    this.getField("sw03").setItems(softwareAuswahl);
    this.getField("sw03").currentValueIndices = 0;
    this.getField("sw03").editable = true;
    this.getField("sw04").setItems(softwareAuswahl);
    this.getField("sw04").currentValueIndices = 0;
    this.getField("sw04").editable = true;
    this.getField("sw05").setItems(softwareAuswahl);
    this.getField("sw05").currentValueIndices = 0;
    this.getField("sw05").editable = true;
    this.getField("sw06").setItems(softwareAuswahl);
    this.getField("sw06").currentValueIndices = 0;
    this.getField("sw06").editable = true;
    this.getField("sw07").setItems(softwareAuswahl);
    this.getField("sw07").currentValueIndices = 0;
    this.getField("sw07").editable = true;
    this.getField("sw08").setItems(softwareAuswahl);
    this.getField("sw08").currentValueIndices = 0;
    this.getField("sw08").editable = true;
    this.getField("sw09").setItems(softwareAuswahl);
    this.getField("sw09").currentValueIndices = 0;
    this.getField("sw09").editable = true;
    this.getField("sw10").setItems(softwareAuswahl);
    this.getField("sw10").currentValueIndices = 0;
    this.getField("sw10").editable = true;
    this.getField("vorname").setFocus();
    this.delay = false;
  }
}
```

```
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=9em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section*{Request for software licenses and installation}
\begin{Form}
\begin{tabular}{|rr|}
\hline
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|c|}{\textbf{Systems owner}}\\
&\\*[-0.9em]\textforlabel{101}{First name:}&\TextField[%
name=vorname, width=20em, bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\textforlabel{102}{Name:}&\TextField[%
name=name, width=20em, bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|c|}{\textbf{PC}}\\
&\\*[-0.9em]\textforlabel{103}{inventory number:}%
&\TextField[name=inventory, width=20em,%
bordercolor={0.65 0.79 0.94}]{}\\
&\\*[-0.9em]\hline
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|c|}{\textbf{Software}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw01, width=30em,%
popdown=true,combo=true, value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw02, width=30em,%
popdown=true,combo=true, value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw03, width=30em,%
popdown=true,combo=true, value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw04, width=30em,%
popdown=true,combo=true, value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw05, width=30em,%
popdown=true,combo=true,value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw06,width=30em,%
popdown=true,combo=true,value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw07,width=30em,%
popdown=true,combo=true,value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw08,width=30em,%
popdown=true,combo=true,value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw09,width=30em,%
popdown=true,combo=true,value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{}\\*[-0.9em]
\multicolumn{2}{|r|}{\ChoiceMenu[name=sw10,width=30em,%
popdown=true,combo=true,value={},%
bordercolor={0.65 0.79 0.94}]{}{}}\\
\multicolumn{2}{|c|}{~}\\*\hline
\end{tabular}
\end{Form}
\end{document}
```

6 Form number three: Interactive application

6.1 Goal

The third form is an interactive application, a click on a button starts an investment calculation. The user can provide an investment amount in €, an investment time in years and the annual rate in percent. A click on the start button calculates the investment result.

6.2 Processing button click events: ex0201

This form is not intended to be printed, so we can use a non-white background. The paper size is set to 5.2057 inches × 3.75 inches

The arcument

```
onclick={doCalculation()}
```

to the “Start” button specifies the function to invoke when the button is pressed.
Note: The function is specified without semicolon here.

The *doCalculation()* function retrieves the input values and runs the calculation. The result is rounded to cents and printed in a text field.

```
\documentclass{scrartcl}
\setcounter{secnumdepth}{0}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\setlength{\paperwidth}{5.2075in}
\setlength{\paperheight}{3.75in}
\renewcommand*{\familydefault}{phv}
\usepackage[pdftex,margin=0.5in]{geometry}
\usepackage{fancyhdr}
\lhead{Big Money, Inc.}\chead{}
\rhead{Promises Dept.}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[pdftex,pdfpagelayout=SinglePage,%
pdftitle={Investment calculation},pdfsubject={Investment}%
]{hyperref}
\definecolor{background}{rgb}{0.99,0.98,0.85}
\pagecolor{background}
\setlength{\parindent}{0.0cm}
\usepackage[pdftex]{insdljs}
```

```
\begin{insDLJS}[exaaca]{exaaca}{JavaScript}
var done = 0;

function docOpen()
{
  if(!done) {
    done = 1;
    var newcolor = [ "RGB", 0.99, 0.98, 0.85];
    var borcolor = [ "RGB", 0.65, 0.79, 0.94]
    this.getField("101").fillColor = newcolor;
    this.getField("101").strokeColor = newcolor;
    this.getField("102").fillColor = newcolor;
    this.getField("102").strokeColor = newcolor;
    this.getField("103").fillColor = newcolor;
    this.getField("103").strokeColor = newcolor;
    this.getField("104").fillColor = newcolor;
    this.getField("104").strokeColor = newcolor;
    this.getField("zeit").currentValueIndices = 0;
    this.getField("zeit").fillColor = newcolor;
    this.getField("summe").fillColor = newcolor;
    this.getField("zins").fillColor = newcolor;
    this.getField("ergebnis").fillColor = newcolor;
  }
}

function doCalculation()
{
  var summe = 0.0 + this.getField("summe").value;
  var zins = 0.0 + this.getField("zins").value;
  var zeit = 0 + this.getField("zeit").value;
  var i, faktor, wert;
  faktor = 1.0 + 0.01 * zins;
  for(i = 0; i < zeit; i++) {
    summe = summe * faktor;
  }
  wert = Math.round(100.0 * summe);
  wert = 0.01 * wert;
  wert = wert.toFixed(2);
  wert = wert.toString();
  this.getField("ergebnis").value = wert;
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (docOpen();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=9em,align=2,%
bordercolor={0.99 0.98 0.85},%
fillcolor={0.99 0.98 0.85},readonly=true]{}%
}
\begin{document}
\begin{Form}
\begin{tabular}{|rl|}
\hline
\multicolumn{2}{|c|}{}\\*[-0.8em]\multicolumn{2}{|c|}{%
\textsc{I\,n\,p\,u\,t~D\,a\,t\,a}%
}\\
&\\*[-0.8em]\textforlabel{101}{Investment capital:}&\TextField[%
name=summe,width=10em,bordercolor={0.65 0.79 0.94}]{}%
~\texteuro\\
&\\*[-0.8em]\textforlabel{102}{Zins:}&\TextField[%
name=zins,width=10em,bordercolor={0.65 0.79 0.94}]{}~\%\\
&\\*[-0.8em]\textforlabel{103}{Investment time:}&%
\ChoiceMenu[name=zeit,popdown=true,bordercolor={0.65 0.79 0.94}%
]{}{{1}={1},{2}={2},{3}={3},{4}={4},{5}={5},%
{6}={6},{7}={7},{8}={8},{9}={9},{10}={10}}~years\\
&\\*[-0.8em]\hline
\multicolumn{2}{|c|}{}\\*[-0.8em]
\multicolumn{2}{|c|}{\textsc{C\,a\,l\,c\,u\,l\,a\,t\,i\,o\,n}}\\
\multicolumn{2}{|c|}{}\\*[-0.8em]
\multicolumn{2}{|c|}{\PushButton[name=start,%
onclick={doCalculation()},bordercolor={0.65 0.79 0.94},%
fillcolor={0.5 0.5 0.5}]{Start}}\\
\multicolumn{2}{|c|}{}\\*[-0.8em]\hline
\multicolumn{2}{|c|}{}\\*[-0.8em]
\multicolumn{2}{|c|}{\textsc{R\,e\,s\,u\,l\,t}}\\
\multicolumn{2}{|c|}{}\\*[-0.8em]\textforlabel{104}{%
Result:}&\TextField[name=ergebnis,width=10em,%
bordercolor={0 0 0},readonly=true]{}~\texteuro\\
\multicolumn{2}{|c|}{}\\
\hline
\end{tabular}
\end{Form}
\end{document}
```

7 Form number four: using both insdljs and inputenc

The inputenc package replaces some characters – i. e. german umlauts – by LaTeX code sequences. This is a good idea for “normal” text but not for JavaScript code.

To avoid the conversion in JavaScript code the characters must not be added directly to strings, add the corresponding JavaScript-Unicode notation instead.

Table 2 contains some example Unicode notations for german umlauts and ligatures.

Table 2: JavaScript-Unicode notation for german umlauts

umlaut/ligature	**JavaScript-Unicode notation**
ä	\u00E4
ö	\u00F6
ü	\u00FC
A	\u00C4
Ö	\u00D6
Ü	\u00DC
ß	\u00DF

Here is an example how to use the Unicode notation (see variable *theText*):

```
\documentclass[ngerman,a4paper]{scrartcl}
\usepackage[latin1]{inputenc}
\usepackage{mathptmx}
\usepackage[scaled=.92]{helvet}
\usepackage{courier}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
\renewcommand*{\familydefault}{phv}
\usepackage[left=25mm,top=25mm,bottom=10mm,right=10mm]{geometry}
\usepackage{fancyhdr}
\lhead{Test}\chead{}\rhead{inputenc and insdljs}
\lfoot{}\cfoot{}\rfoot{}
\pagestyle{fancy}
\usepackage{graphicx}
\usepackage{color}
\usepackage[
  pdftex,a4paper=true,colorlinks=true,
  pdftitle={insdljs und inputenc},pdfsubject={Test},
  pdfauthor={ich},
  pdfpagemode=UseNone,pdfstartview=FitH,
  pagebackref,pdfhighlight={/N}
]{hyperref}
\usepackage[pdftex]{insdljs}
\begin{insDLJS}[exaada]{exaada}{JavaScript auf Document Level}
// indicate that the function was not yet run
var done = 0;

// text to show when file is opened
var theText =
"Test to show \u00E4\u00F6\u00FC\u00C4\u00D6\u00DC\u00DF";

// function to run when opening the document
function myFirstJavaScriptFunction()
{
  if(!done) {
    done = 1;
    app.alert(theText);
  }
}
\end{insDLJS}
\OpenAction{/S /JavaScript /JS (myFirstJavaScriptFunction();)}
\newcommand{\textforlabel}[2]{%
\TextField[name={#1},value={#2},width=7em,align=2,%
bordercolor={1 1 1},readonly=true]{}%
}
\begin{document}
\section{Testing insdljs and inputenc}
This is a test whether insdljs and inputenc can be used combined.
The pop-up box should show the following german umlauts and ligatures:
\begin{itemize}
\item  ä (ae)
\item  ö (oe)
\item  ü (ue)
\item  Ä (Ae)
\item  Ö (Oe)
\item  Ü (Ue)
\item  ß (sz)
\end{itemize}
\end{document}
```

A History

- Version 0.9.5 (2008-02-08)
  shows how to add help using PDF annotations or tooltip texts.

- Version 0.9.4 (2007-06-20)
  contains a new section how to use the insdljs and inputenc packages together. German umlauts must not be written directly to strings in JavaScript code, the corresponding JavaScript Unicode notations must be used instead.

- Version 0.9.3 (2007-06-15)
  contains corrected options for the hyperref package.

```
pdfpagemode=UseNone
```

  was replaced by

```
pdfpagemode=UseNone
```

- Version 0.9.2 (2007-06-14)
  uses other fonts than previous versions. The document preamble was derived from another document containing mathematics. As this document does not contain mathematics we do no need the packages related to mathematics.

- Version 0.9.1 (2007-06-13)
  contains corrections in the “Fonts” section.

```
\usepackage[T1]{fontenc}
```

  does not set up LaTeX to use type-1 fonts (as stated in version 0.9.0).
  Instead it changes the preferred font encoding vector to T1 instead of OT1 if type-1 fonts are used.

- Version 0.9.0
  is the first version published on CTAN.
  I’m not a native english speaker, so the english tutorial and examples may need corrections in style, spelling and grammar.
  Please feel free to contact me using the web form http://www.fh-schmalkalden.de/url.php?/page/1026/select_wert/3023 if you have suggestions.

B GNU Free Documentation License

Version 1.2, November 2002

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

B.1 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

B.2 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section B.3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

B.3 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

B.4 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections B.2 and B.3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

B.5 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section B.4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

B.6 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

B.7 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section B.3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

B.8 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section B.4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section B.4) to Preserve its Title (section B.1) will typically require changing the actual title.

B.9 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.10 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

References

[ADS05a] Adobe Developer Support. *Acrobat JavaScript Scripting Guide*. Adobe Systems Incorporated, 2005.

[ADS05b] Adobe Developer Support. *Acrobat JavaScript Scripting Reference*. Adobe Systems Incorporated, 2005.

[ASI01] Adobe Systems Incorporated. *PDF Reference, third edition*. Adobe Systems Incorporated, 2001.

[RO04] Sebastian Rahtz and Heiko Oberdiek. *Hypertext marks in LaTeX: a manual for hyperref*. 2004.

[Sup03] Adobe Technical Support. *Acrobat JavaScript Object Specification Version 5.1*. Adobe Systems Incorporated, 2003.