Description of the module PDF Form

Uwe Stöhr

By loading the module PDF Form the whole document or a part of it can be made a PDF form. You can add all possible form elements like text fields buttons etc. This document describes how this is done.

1 Preparation

At first load the module PDF Form in the document settings of your file. To make the whole document a form, insert the environment Begin PDF Form from LyX's dropdown menu at the beginning of the document. If only a part of the document should be a form, add Begin PDF Form at the position where the form should start.

Note: It is only possible to have one form in a PDF file!

If the form data should be submitted to a server or the like, then you must specify the URL as parameter of the Begin PDF Form environment. This is done by using the menu Insert ▷ PDF Form Parameters. For example in this document the following was inserted to the parameter inset:

action=mailto:forms@lyx.test?subject=The submitted PDF form, method=post

This sends the form data as email to *forms@lyx.test* when the user presses the submit button. The email subject will be "The submitted PDF form" and the submit method is **post**. (The other possible method would be **get**.) For more about submitting see sec. 2.5.

2 Form elements

There are 6 possible elements in a PDF form:

2.1 Text field

A text field is added by inserting the custom inset TextField using the menu Insert ▷ Custom Insets. Write inside the inset the label for the text field that will be printed in the PDF output before the field. Here is an example:

Enter your name here:

You must also specify a name for text fields. To do this set the cursor into its Params inset and insert the parameter **name=customer** (where **customer** is the field name).

Using the same name for several text fields results in an automatic duplication of the inserted text. This is for example useful if the user should input a date that should appear at different places in the form.
Here are 2 fields as example: Fill one and see that the other one will be filled automatically when you click outside of the field:

Note: Special characters should be avoided in element names.

All form fields can be customized by adding parameters.

Here is a customized multiline text field with a width of 60 % of the text width, 3 cm height and a proposed content:

Enter your name here: proposed text

Note: It is highly recommended to use a uniform layout of all fields in a form. See sec. 3 how this is done.

2.2 Check box

A check box is added by inserting the custom inset CheckBox. Write inside the inset the label for the box. Here is an example:

Are you older than 18 years?

You must specify also for check boxes a name. Insert e. g. the parameter **name=age** (where **age** is the name) into the Params inset.

2.3 Choice menu

A choice menu is added by inserting the custom inset ChoiceMenu. A label for the inset is added by using the menu Insert ▷ Label. The different choices are inserted to the inset as comma-separated list. There are 3 choice menu types:

Radio Only one choice can be selected

Combo The choices are listed in a combo box (dropdown list) but the user is allowed to input something that is not in the predefined list.

Popdown All choices are listed below each other. If the menu is not high enough a scroll bar is automatically added.

To determine the choice menu type one uses the parameter **radio**, **combo** or **popdown**.

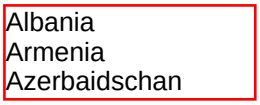
Here is an example for the radio type:

Sex: male ☐ female ☐

Here is an example for the combo type:

Country: 

Here is an example for the popdown type:

Country: 

You must also specify a name for choice menus. Insert for example the parameter **name=country** (where **country** is the name) into the Params inset.

It is recommended to add a short name for every choice to be able to access them and to know which one was chosen by the user. This is done by adding **=name** behind each choice (where **name** is the choice name). For example the radio type example contains this choice list: **male=m, female=f**. If the user chose "male" you know that he chose the choice "m". With the name you can also preset/propose e.g. the choice "male" by adding the parameter **default=m**. Here is an example:

Sex: male ☐ female ☐

2.4 Push button

A push button is added by inserting the custom inset PushButton. Write inside the inset the label for the button. Here is an example:

Don't click on this button or your fridge will be destroyed!!!

The action that is triggered by pressing the button is specified by JavaScript code. To do this, add the parameter **onclick={}** and insert the JavaScript code between

the braces. For info about JavaScript, see its documentation, [1]. In the following example this JavaScript code was used:
app.alert("What the hell? Now you destroyed your fridge. “Congratulations.”")

Don’t click on this button or your fridge will be destroyed!!!

Note: The outer quotes are part of the JavaScript code and must therefore be inserted as TEX code!

A common usage for push buttons is to open a weblink. Such a button is created by inserting a hyperlink as button text. Here is an example:

Information how to fill out this form

To get rid of the predefined frame for weblinks add this to the additional options in the document settings under PDF properties:

urlbordercolor={1 0 0}

if your push button border color is red (the default), otherwise use the same color definition as for your push button border color.

2.5 Submit button

A submit button is used to submit the form data to a server. It is added by inserting the custom inset SubmitButton. The label for the button is written into the inset. Here is an example:

Send your data via email

The submit button creates a forms data format (FDF) file (file extension *.fdf) which is then submitted. The FDF file contains only the form data. They can later only be applied to a PDF form if all elements in the form have a name.

Note: You can only use a submit button if you specified the submit method and a target in the inset PDF Form Parameters! See sec. 1 for the description. If there are no specifications you will get LATEX errors.

2.6 Reset button

A reset button is used to reset all form elements to the initial state. It is added by inserting the custom inset ResetButton. The label for the button is written into the inset. Here is an example:

Reset the form

3 Form element customization

Since all form elements should look uniform, one can determine their layout using the following styles:

Text Field Style

Check Box Style

List Box Style affects all choice menu types

Combo Box Style affects only the combo box style

Popdown Box Style affects only the popdown box style

Radio Box Style affects only the radio box style

Push Button Style

Submit Button Style

Reset Button Style

The content of the styles is a comma-separated list of parameters. The possible parameters are listed in sec. 6.2 “Forms optional parameters” of the documentation of the LaTeX package **hyperref**, [2]. It is important that the parameter *print* is always part of the definition. Otherwise the elements will not appear in the PDF. The defined style is applied to all elements following the definition.

Here are some examples:

- Text field with gray background, lime text color, a red border only below the field, right alignment, 16 pt font size and a limitation for maximal 10 characters to insert:

  Enter your name here:

Note: To see in *Adobe Reader* and *Acrobat* custom background colors you must disable in these programs the highlighting of form fields. (Option “Show border hover color for fields”)

- Check box with normal border and symbol \ding{55}:

  Are you older than 18 years?

The symbol is either specified as number or with the command
\ding{number}
where **number** is one of the possible numbers listed in Table 2 of the documentation of the LaTeX package **pifont**, [3].

- Combo choice menu with dashed, colored border where the last entry is preselected:

  Country: Azerbaidschan

- Popdown choice menu where the second entry is preselected:

  Country: Albania / Armenia / Azerbaidschan

Note: The parameters **borderstyle B** and **I** and **color** have no effect for popdown choice menus.

- Radio choice menu with inverted bevel border and symbol number 3 as checkmark:

  Sex: male ○ female ○

Note: The parameters **backgroundcolor**, **color**, **height** and **width** have no effect for push, submit and reset buttons.

As workaround use a colored box and/or color the box text.

- Push button with a bevel border and colored text:

  Don't click on this button!!!

- Submit button without border, with cyan background and increased height:

  Send your data via mail

- Reset button with a width of 7 cm:

  Reset the form

4 PDF action buttons

One often needs basic actions to be done by the PDF viewer program, for example to print the form. Such PDF program-specific actions can be triggered by adding inserting the custom inset PDFAction. The PDF viewers *Acrobat* and *Adobe Reader* can handle all possible actions while other PDF viewers might only support some of them. However, all PDF viewers support the basic things like printing, save as, view in fullscreen etc..

To specify the action insert its name to the Action inset. A list with possible action names can be found in sec. 5 "Acrobat-specific behavior" of the documentation of the LaTeX package **hyperref**, [2].

Here are some examples:

Printing: Print the document

Save as: Save document as

View in fullscreen: View the form in fullscreen

4.1 Action button customization

Customizing the action buttons requires the usage of boxes because things like the width, height and border separation cannot be specified as button parameters.

The easiest way to customize the buttons is to fill it with a custom box. For example the button should be 5 cm wide and have the height of 2 lines. Then create this parbox:

Save form as

inside a PDFAction inset. This is the result:

Save form as

To customize the border color, border thickness and the distance to the border one changes the border settings of the box. An example with a 4 pt thick, teal border and 3 pt border separation :

Save form as

To customize the border thickness, add the command **pdfborder={0 0 t}** to the additional options in the document settings under PDF properties and replace **t** with a number that is the thickness in pixels. If you want to change the border thickness only for certain buttons, use the environment PDF link setup before the button and insert there the command **pdfborder**. The default value of **pdfborder** is **{0 0 1}**.

Note: pdfborder affects all link types, not only the action links.

An example with a 4 pixel thick border:
Save form as

To change the border color use the command **menubordercolor={r g b}** where **r**, **g** and **b** are numbers between 0 and 1 for the colors red, green and blue. The default value of **menubordercolor** is **{1 0 0}**.

An example with **menubordercolor={0.1 0.9 0.5}**:
Save form as

To change the background color use a colored box.

An example without a border and with lime background color:
Save form as

5 Dynamic form elements

It is also possible to have dynamic form elements. This means that depending on the actions of the user elements can (dis)appear or change their appearance. To use this feature, add these lines to your LaTeX preamble:

```
\usepackage[pdftex]{insdljs}
\pdfcatalog{/AA \the\pdflastobj\space 0 R}
```

Then add the necessary JavaScript code to your document LaTeX preamble or as TEX code to your document. For info about JavaScript, see its documentation, [1].

You need to install the package **insdljs** to see the content of this section in the output.

6 General notes

- Submitting and applying data requires that all form elements have a name.
- Creating a PDF form requires pdfTEX or LuaTEX. Use therefore either the LYX export formats PDF (pdflatex) or PDF (LuaTeX).
- To see in *Adobe Reader* and *Acrobat* custom background colors you must disable in these programs the highlighting of form fields. (Option "Show border hover color for fields")

References

[1] JavaScript reference

[2] Documentation of the LATEX package **hyperref**

[3] Documentation of the LATEX package **pifont**