

# SDAPS

Suchen:

SDAPS NEWS Documentation MailingList

Documentation / Tutorial

Locked History Dateianhänge

Weitere Aktionen:

## SDAPS

- [Contribute](#)
- [Download](#)
- [Issue Tracker](#)
- [Mailing List](#)
- [News](#)
- [Tutorial](#)
- [Documentation](#)
  - [Dependencies](#)
  - [LaTeX](#)
  - [LibreOffice](#)
  - [Scanning](#)
  - [GUI](#)
  - [Recognition](#)
  - [Developer Documentation](#)
  - [Missing](#)
- [References](#)
- [Impressum](#)

### Inhaltsverzeichnis

1. [Tutorial describing the usage of SDAPS](#)
  1. [Preface](#)
  2. [Creating the Questionnaire](#)
  3. [Intializing Setup](#)
  4. [Printing](#)
  5. [Scanning](#)
  6. [Add the images to the project](#)
  7. [Running the optical mark recognition](#)
  8. [Manual correction](#)
  9. [Creating a PDF report](#)
  10. [CSV export](#)
  11. [Other things to try](#)
    1. [Questionnaire-ID](#)
    2. [Global-ID](#)
    3. [Annotate](#)
    4. [Reorder](#)
  12. [LaTeX based report](#)

## Tutorial describing the usage of SDAPS

In this tutorial we will go trough the lifecycle of a SDAPS project. We are going to use an existing LaTeX questionnaire example. There are also example scans for this questionnaire available so that you don't need to have a scanner and printer ready right away.

### Preface

In the example we need to call the SDAPS executable. It will be called using `sdaps` as if it installed on the system. You can also use the `sdaps.py` script from the source directory instead. The `$` denotes the shells prompt, everything else is output from the program.

We are going to use `/tmp/project` as the surveys path. This is obviously a bad idea for any real project.

### Creating the Questionnaire


The first step to conduct a survey is to design the questionnaire itself. You'll need to take some time to first figure out what questions to ask before designing the questionnaire.

We are going to use this [example questionnaire \(PDF Version\)](#) here. If you would like to play around a bit with it, you can compile the LaTeX document yourself. Note that you need to copy the SDAPS-LaTeX data into the directory before doing that. It lives in the `tex` Directory of the source code or in `$PREFIX/share/sdaps/tex` if it is installed. Where `$PREFIX` will usually be `/usr` or `/usr/local`.

You can familiarize yourself with the LaTeX-Document and the resulting PDF-file. Notice that the PDF has a "draft" text overlayed. This is because the barcode at the bottom is just an example and it will change once the project is created.

Have a look at the [LaTeX](#) page for some more information about the different LaTeX macros that exists in SDAPS. You can also change the size of the paper using the normal LaTeX methods.

### Intializing Setup

 It is best to use the `example.tex` from this page and not a modified version! If you modify the document the example scan will not work later on!

Once we are happy with the questionnaire, we can create the survey directory that SDAPS uses to store all the data that belongs to the project.

This is the first time that we need to run an SDAPS command. The syntax for the command is generally the following:

```
$ sdaps PROJECT_DIR COMMAND [arguments]
```

Where `PROJECT_DIR` is the directory for the survey, and `command` is the SDAPS command that is executed. Most commands will require some arguments. You can always get a help by running:

zuletzt geändert am 2014-07-30 12:19:15 durch BenjaminBerg

So we create the project using the provided `example.tex`. The tutorial assumes that the LaTeX file is in the current directory.

```
$ sdaps /tmp/project setup_tex example.tex
-----
- SDAPS -- setup_tex
-----
Running pdflatex now twice to generate the questionnaire.
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
 restricted \write18 enabled.
entering extended mode
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
 restricted \write18 enabled.
entering extended mode
Running pdflatex now twice to generate the questionnaire.
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
 restricted \write18 enabled.
entering extended mode
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
 restricted \write18 enabled.
entering extended mode
The Title
Date: 10.03.2013
Author: The Author
Questionnaire
1(Head) Range Questions
1.1(Mark) How often do you use SDAPS? {1}
          never - daily
          0(Checkbox) 87.8 78.4 3.5 3.5
[SNIP]
```

Now, what does the output tell us? First the document is compiled four times. After the first two compiles SDAPS reads all the metadata and calculates the "Survey-ID" using the box positions and sizes and some other metadata. After that, the document is recompiled again using the correct "Survey-ID".

The rest of the output is a textual representation of the questionnaire. We can ignore it, especially because the "annotate" command creates a much nicer view of everything.

## Printing

After the setup of the example we now have a [questionnaire.pdf](#) in the project directory that can be printed. This depends on the setup of the survey (done using the documentclass options in the LaTeX document). If we make each printout unique using a "Questionnaire-ID" then we would need to use the "stamp" command to create the printable questionnaire. Please refer to the section at the bottom of the page for more information.

Simply print the created PDF file using your favorite PDF reader. It is a good idea to disable any "scale page to fit printable area" option.

## Scanning

After you have a couple of printed and filled in questionnaires you need to scan them. There is a whole [scanning section](#) about this. Please have a look there.

You can skip this step for now and instead use the provided [example.tif](#) file.


## Add the images to the project

Once you have a scan in the correct format you can add it to survey directory that was created earlier. This is done using the add command:

```
$ sdaps /tmp/project add example.tif
-----
- SDAPS -- add
-----
Processing example.tif
Done
```

If everything worked fine you will see no further output. The original file is copied into the project directory as `1.tif`.

You can repeat this step if you have multiple scans.

 Do **not** remove or modify the copied TIFF files. SDAPS stores information that references these files (ie. it creates a record for each page). If you accidentally added a file, you can recreate the project and start from scratch.

## Running the optical mark recognition

The next step is to run the optical mark recognition. This works using the `recognize` command. So from the command line again we run:

```
$ sdaps /tmp/project recognize
-----
- SDAPS -- recognize
-----
3 sheets
|#####| 100% 00:00:02
0.887902 seconds per sheet
```

This step takes longer as the recognition algorithm needs to do its work for each image. The progress bar shows how much time it is expected to take. Usually it will take less than a second for a two page questionnaire.

## Manual correction

Sometimes the computer might not correctly identify the state of a checkbox (SDAPS sometimes has some trouble because of the feature to uncheck a box by filling it out).

To correct any errors we can use the graphical user interface. We start it using

```
$ sdaps /tmp/project gui
-----
- SDAPS -- gui
-----
```

There is a much more [complete section](#) about it. You can quickly go through all images and correct any errors using the mouse. When the view is focused you can go forward/backward using Enter and Shift+Enter.

There is an estimation for the quality of the recognition. You can sort the data using this estimation and only go through the first couple of pages. The amount of time to spend on this will depend on the required accuracy.

## Creating a PDF report

To create a PDF report with the results simply run:

```
$ sdaps /tmp/project report
-----
- SDAPS -- report
-----
```

It creates a PDF file `report_1.pdf`. Have a look at the file; you can also [download the report](#) that will be created for the example data.

Note that we can also do partial reports by using filters. Just a quick example (please refer to the rest of the documentation for an explanation):

```
$ sdaps /tmp/project report -f '_1_2_3 == 5'
-----
- SDAPS -- report
-----
```

This filters for question 1.2.3 (ease of use of LaTeX) and the rightmost choice "easy".

## CSV export

Obviously sometimes it might be necessary to feed the data into another program. For this the CSV export command was created:

```
$ sdaps /tmp/project csv export
-----
- SDAPS -- csvdata
-----
```

A file called `data_1.csv` will be created in the project directory.

## Other things to try

### Questionnaire-ID

If it is required, you can put a unique barcode on every created questionnaire. This is internally called the "Questionnaire-ID" by SDAPS.

To do this, you need to add the `print_questionnaire_id` to the document class before running the setup routine. The document will now also contain a barcode for the "Questionnaire-ID" which will be located in the bottom left corner of the page.

In this setup it is now necessary to generate the required amount of unique documents. This is done using the `stamp` command in SDAPS. You have the choice of either creating numeric random IDs, or supplying a set of IDs (anything that can be encoded in CODE128 is allowed) in a file.

For example, to create 15 questionnaires with randomized IDs you can run

```
$ sdaps /tmp/project stamp -r 15
```

To specify non-random IDs create a file with one ID per line. It might look like the following. Lets call it `ids.txt`:

```
First ID
Second ID
Some Name
```

Then run the stamp command, with the created file as an argument:

```
$ sdaps /tmp/project stamp -f ids.txt
```

Both commands will create a new `stamp_X.pdf` file (where X is a number) which can be printed.

## Global-ID

There is a third ID that SDAPS can have (besides the internal Survey-ID and Questionnaire-ID), which is called the "Global-ID". This ID is printed in the center of the page and exists for user defined purposes. It is not used by SDAPS itself.

An example use case for the "Global-ID" might be handing out the same questionnaire in different lectures. You could encode the lecture in the "Global-ID" and separate everything using this information.

## Annotate

As mentioned before, you can create a PDF to see if the values read from the designed questionnaire are all correct (checkbox positions, etc.). To use run:

```
$ sdaps /tmp/project annotate
```

The file `annotated_questionnaire.pdf` is created. Might be a bit ugly, but one can easily check that everything is good.

 This command requires the GObject Introspection binding information for poppler to be installed.

## Reorder

To try out this command we need a questionnaire that is printed on multiple pages, and unique Questionnaire-IDs.

If the questionnaire has multiple pages it can happen that the pages get mixed before the scan happens. The "reorder" command will sort all pages so that everything is together again.

First identify all pages ie. read all the barcodes:

```
$ sdaps /tmp/project recognize --identify
```

Then reorder the pages:

```
$ sdaps /tmp/project reorder
```


And when that is done you can do the normal "recognize" step:

```
$ sdaps /tmp/project recognize
```

## LaTeX based report

You can create a report that is rendered using LaTeX.

```
$ sdaps /tmp/project report_tex
```

 This command requires the siunitx LaTeX package to work properly.