Package: staplr (via r-universe)

October 30, 2026

Type Package

Title A Toolkit for PDF Files

Version 3.2.2

Depends R (>= 3.4.0)

Description Provides functions to manipulate PDF files: fill out PDF forms; merge multiple PDF files into one; remove selected pages from a file; rename multiple files in a directory; rotate entire pdf document; rotate selected pages of a pdf file; Select pages from a file; splits single input PDF document into individual pages; splits single input PDF document into parts from given points.

SystemRequirements Java (>= 8.0)

License GPL-3

RoxygenNote 7.2.3

Imports tcltk, stringr, assertthat, glue, XML, rJava, fs, purrr, pdftools

Suggests lattice, testthat

Encoding UTF-8

BugReports https://github.com/pridiltal/staplr/issues

NeedsCompilation no

Author Priyanga Dilini Talagala [aut, cre], Ogan Mancarci [aut], Daniel Padfield [aut], Granville Matheson [aut], Pedro Rafael D. Marinho [ctb] (<https://orcid.org/0000-0003-1591-8300>), Marc Vinyals [cph, aut] (Author and copyright holder of included pdftk-java package)

Maintainer Priyanga Dilini Talagala <pritalagala@gmail.com>

Config/pak/sysreqs default-jdk

Repository https://cran.r-universe.dev

Date/Publication 2023-09-18 15:31:04 UTC

RemoteUrl https://github.com/cran/staplr

RemoteRef HEAD

RemoteSha 9f95546e36bd48a5523536903415db64efb0d5a1

Contents

combine_pdf . . . 2
get_fields . . . 3
identify_form_fields . . . 4
remove_pages . . . 5
rename_files . . . 6
rotate_pages . . . 7
rotate_pdf . . . 9
select_pages . . . 10
set_fields . . . 11
split_from . . . 13
split_pdf . . . 14
staple_pdf . . . 15
staplr . . . 16

Index **17**

`combine_pdf` *Combine multiple PDF files*

Description

Combine multiple PDF files by delimiting the sequences of pages in each file.

Usage

```
combine_pdf(vec_input, output = "output.pdf", start_pages = NA, end_pages = NA)
```

Arguments

`vec_input`	Vector with paths of PDF files to be combined.
`output`	PDF file path result of the combination.
`start_pages`	Vector with the initial pages of each file. If `NA`, the default, will be considered the first page.
`end_pages`	Vector with the final pages of each file. If `NA`, the default, will be considered the last page.

Value

In the path informed in `output`, the PDF file resulting from the combination of multiple files passed to `vec_output` will be saved.

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:2) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}

output <- tempfile(fileext = '.pdf')
combine_pdf(
   vec_input =
      file.path(dir, paste("plot", 1:2, ".pdf", sep = "")),
   output = output,
   start_pages = c(NA, NA),
   end_pages = c(NA, NA)
 )
 }
```

`get_fields` *Get form fields from a pdf form*

Description

If the toolkit Pdftk is available in the system, it will be called to get form fields from a pdf file.

See the reference for detailed usage of `pdftk`.

Usage

```
get_fields(
  input_filepath = NULL,
  convert_field_names = FALSE,
  encoding_warning = TRUE
)
```

Arguments

`input_filepath`	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`convert_field_names`	By default `pdftk` will encode certain characters of the field names in plain text UTF-8 so if using a non-latin alphabet, your field names might be illegible. Setting this to TRUE will turn the UFT-8 code into characters. However this process it not guaranteed to be perfect as pdftk does not differentiate between encoded text and regular text using escape characters. If you have field names that intentionally include components that look like encoded characters this will

attempt to fix them. Use this option only when necessary. If TRUE, remember to set it to TRUE when using set_fields as well.

encoding_warning

If field names include strings that look like plain text UTF-8 codes, the function will return a warning by default, suggesting setting convert_field_names to codeTRUE. If encoding_warning is FALSE, these warnings will be silenced.

Value

A list of fields. With type, name and value components. To use with set_fields edit the value element of the fields you want to modify. If the field of type "button", the value will be a factor. In this case the factor levels describe the possible values for the field. For example for a checkbox the typical level names would be "Off" and "Yes", corresponding to non checked and checked states respectively.

Author(s)

Ogan Mancarci

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

See Also

link{set_fields}

Examples

```
pdfFile = system.file('simpleForm.pdf',package = 'staplr')
fields = get_fields(pdfFile)
```

---

identify_form_fields *Identify text form fields*

Description

Helps identification of text forum fields by creating a file that is filled with field names. Some pdf editors show field names when you mouse over the fields as well.

Usage

```
identify_form_fields(
  input_filepath = NULL,
  output_filepath = NULL,
  overwrite = TRUE,
  convert_field_names = FALSE,
  encoding_warning = TRUE
)
```

Arguments

input_filepath	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
output_filepath	the path of the output PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
overwrite	If a file exists in output_filepath, should it be overwritten.
convert_field_names	By default pdftk will encode certain characters of the field names in plain text UTF-8 so if using a non-latin alphabet, your field names might be illegible. Setting this to TRUE will turn the UFT-8 code into characters. However this process it not guaranteed to be perfect as pdftk does not differentiate between encoded text and regular text using escape characters. If you have field names that intentionally include components that look like encoded characters this will attempt to fix them. Use this option only when necessary. If TRUE, remember to set it to TRUE when using set_fields as well.
encoding_warning	If field names include strings that look like plain text UTF-8 codes, the function will return a warning by default, suggesting setting convert_field_names to codeTRUE. If encoding_warning is FALSE, these warnings will be silenced.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Examples

```
output <- tempfile(fileext = '.pdf')
pdfFile = system.file('simpleForm.pdf',package = 'staplr')
identify_form_fields(pdfFile, output)
```

---

remove_pages *Remove selected pages from a file*

Description

If the toolkit Pdftk is available in the system, it will be called to remove the given pages from the seleted PDF files.

See the reference for detailed usage of pdftk.

Usage

```
remove_pages(
  rmpages,
  input_filepath = NULL,
  output_filepath = NULL,
  overwrite = TRUE
)
```

Arguments

rmpages	a vector of page numbers to be removed
input_filepath	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
output_filepath	the path of the output PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
overwrite	If a file exists in output_filepath, should it be overwritten.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Priyanga Dilini Talagala

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:3) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}
output_file <- file.path(dir, paste('Full_pdf.pdf',  sep = ""))
staple_pdf(input_directory = dir,  output_filepath = output_file)
input_path <- file.path(dir, paste("Full_pdf.pdf",  sep = ""))
output_path <-  file.path(dir, paste("trimmed_pdf.pdf",  sep = ""))
remove_pages(rmpages = 1, input_path, output_path)
}
```

---

rename_files *Rename multiple files*

Description

Rename multiple files in a directory and write renamed files back to directory

Usage

```
rename_files(input_directory = NULL, new_names)
```

Arguments

input_directory	the path of the input PDF files. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
new_names	a vector of names for the output files.

Value

A logical vector indicating which files were successfully renamed.

Author(s)

Priyanga Dilini Talagala

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

---

rotate_pages *Rotate selected pages of a pdf file*

Description

If the toolkit Pdftk is available in the system, it will be called to rotate the given pages of the seleted PDF files

See the reference for detailed usage of pdftk.

Usage

```
rotate_pages(
  rotatepages,
  page_rotation = c(0, 90, 180, 270),
  input_filepath = NULL,
  output_filepath = NULL,
  overwrite = TRUE
)
```

Arguments

rotatepages	a vector of page numbers to be rotated
page_rotation	An integer value from the vector c(0, 90, 180, 270). Each option sets the page orientation as follows: north: 0, east: 90, south: 180, west: 270. Note that the orientation cannot be cummulatively changed (eg. 90 (east) will always turn the page so the beginning of the page is on the right side)
input_filepath	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
output_filepath	the path of the output PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
overwrite	If a file exists in output_filepath, should it be overwritten.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Priyanga Dilini Talagala

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:3) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}
output_file <- file.path(dir, paste('Full_pdf.pdf',  sep = ""))
staple_pdf(input_directory = dir, output_filepath = output_file)
input_path <- file.path(dir, paste("Full_pdf.pdf",  sep = ""))
output_path <-  file.path(dir, paste("Rotated_pgs_pdf.pdf",  sep = ""))
rotate_pages(rotatepages = c(2,3), page_rotation = 90,  input_path, output_path)
}
```

rotate_pdf *Rotate entire pdf document*

Description

If the toolkit Pdftk is available in the system, it will be called to rotate the entire PDF document

See the reference for detailed usage of `pdftk`.

Usage

```
rotate_pdf(
  page_rotation = c(0, 90, 180, 270),
  input_filepath = NULL,
  output_filepath = NULL,
  overwrite = TRUE
)
```

Arguments

`page_rotation`	An integer value from the vector c(0, 90, 180, 270). Each option sets the page orientation as follows: north: 0, east: 90, south: 180, west: 270. Note that the orientation cannot be cummulatively changed (eg. 90 (east) will always turn the page so the beginning of the page is on the right side)
`input_filepath`	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`output_filepath`	the path of the output PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`overwrite`	If a file exists in `output_filepath`, should it be overwritten.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Priyanga Dilini Talagala

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:3) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}
output_file <- file.path(dir, paste('Full_pdf.pdf',  sep = ""))
staple_pdf(input_directory = dir, output_filepath = output_file)
input_path <- file.path(dir, paste("Full_pdf.pdf",  sep = ""))
output_path <-  file.path(dir, paste("rotated_pdf.pdf",  sep = ""))
rotate_pdf( page_rotation = 90,  input_path, output_path)
}
```

`select_pages` *Select pages from a file*

Description

If the toolkit Pdftk is available in the system, it will be called to combine the selected pages in a new pdf file.

See the reference for detailed usage of `pdftk`.

Usage

```
select_pages(
  selpages,
  input_filepath = NULL,
  output_filepath = NULL,
  overwrite = TRUE
)
```

Arguments

`selpages`	a vector of page numbers to be selected
`input_filepath`	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`output_filepath`	the path of the output PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`overwrite`	If a file exists in `output_filepath`, should it be overwritten.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Granville Matheson, Priyanga Dilini Talagala

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:3) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}
output_file <- file.path(dir, paste('Full_pdf.pdf',  sep = ""))
staple_pdf(input_directory = dir, output_filepath = output_file)
input_path <- file.path(dir, paste("Full_pdf.pdf",  sep = ""))
output_path <-  file.path(dir, paste("trimmed_pdf.pdf",  sep = ""))
select_pages(selpages = 1, input_path, output_path)
}
```

`set_fields` *Set fields of a pdf form*

Description

If the toolkit Pdftk is available in the system, it will be called to fill a pdf form with given a list of fields. List of fields can be acquired by `get_fields` function.

See the reference for detailed usage of `pdftk`.

Usage

```
set_fields(
  input_filepath = NULL,
  output_filepath = NULL,
  fields,
  overwrite = TRUE,
  convert_field_names = FALSE,
  flatten = FALSE
)
```

Arguments

input_filepath	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
output_filepath	the path of the output PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
fields	Fields returned from get_fields function. To make changes in a PDF, edit the values component of an element within this list
overwrite	If a file exists in output_filepath, should it be overwritten.
convert_field_names	If you set convert_field_names when using get_fields you should set this to TRUE as well so the fields can be matched correctly.
flatten	If TRUE, the form fields will be flattened and turned into plain text.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Ogan Mancarci

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

See Also

get_fields

Examples

```
pdfFile = system.file('simpleForm.pdf',package = 'staplr')
fields = get_fields(pdfFile)

fields$TextField$value = 'this is text'
fields$TextField2$value = 'more text'
fields$Checkbox$value = 'Yes'

output <- tempfile(fileext = '.pdf')

set_fields(pdfFile,output,fields)
```

split_from *Splits single input PDF document into parts from given points*

Description

If the toolkit Pdftk is available in the system, it will be called to Split a single input PDF document into two parts from a given point

See the reference for detailed usage of `pdftk`.

Usage

```
split_from(
  pg_num,
  input_filepath = NULL,
  output_directory = NULL,
  prefix = "part",
  overwrite = TRUE
)
```

Arguments

`pg_num`	A vector of non-negative integers. Split the pdf document into parts from the numbered pages.
`input_filepath`	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`output_directory`	the path of the output directory
`prefix`	A string for output filename prefix
`overwrite`	If a file exists in `output_filepath`, should it be overwritten.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Priyanga Dilini Talagala and Ogan Mancarci

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:4) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}
staple_pdf(input_directory = dir, output_filepath = file.path(dir, 'Full_pdf.pdf'))
input_path <- file.path(dir, "Full_pdf.pdf")
split_from(pg_num=2, input_filepath = input_path ,output_directory = dir )
}
```

---

split_pdf *Splits single input PDF document into individual pages.*

Description

If the toolkit Pdftk is available in the system, it will be called to Split a single input PDF document into individual pages.

See the reference for detailed usage of `pdftk`.

Usage

```
split_pdf(input_filepath = NULL, output_directory = NULL, prefix = "page_")
```

Arguments

`input_filepath`	the path of the input PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`output_directory`	the path of the output directory
`prefix`	A string for output filename prefix

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Priyanga Dilini Talagala and Ogan Mancarci

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:3) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}
staple_pdf(input_directory = dir, output_filepath = file.path(dir, 'Full_pdf.pdf'))
split_pdf(input_filepath = file.path(dir, paste("Full_pdf.pdf",  sep = "")),output_directory = dir )
}
```

---

staple_pdf *Merge multiple PDF files into one*

Description

If the toolkit Pdftk is available in the system, it will be called to merge the PDF files.

See the reference for detailed usage of `pdftk`.

Usage

```
staple_pdf(
  input_directory = NULL,
  input_files = NULL,
  output_filepath = NULL,
  overwrite = TRUE
)
```

Arguments

`input_directory`	the path of the input PDF files. The default is set to NULL. If NULL, it prompt the user to select the folder interactively.
`input_files`	a vector of input PDF files. The default is set to NULL. If NULL and `input_directory` is also NULL, the user is propted to select a folder interactively.
`output_filepath`	the path of the output PDF file. The default is set to NULL. IF NULL, it prompt the user to select the folder interactively.
`overwrite`	If a file exists in `output_filepath`, should it be overwritten.

Value

TRUE if the operation was succesfful. FALSE if the operation fails.

Author(s)

Priyanga Dilini Talagala and Daniel Padfield

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

Examples

```
if (requireNamespace("lattice", quietly = TRUE)) {
dir <- tempfile()
dir.create(dir)
for(i in 1:3) {
pdf(file.path(dir, paste("plot", i, ".pdf", sep = "")))
print(lattice::xyplot(iris[,1] ~ iris[,i], data = iris))
dev.off()
}
output_file <- file.path(dir, paste('Full_pdf.pdf',  sep = ""))
staple_pdf(input_directory = dir, output_filepath = output_file)
}
```

---

staplr *staplr: A package containing a toolkit for PDF files*

Description

This package provides function to manipulate PDF files: merging multiple PDF files into one.

Author(s)

Priyanga Dilini Talagala, Ogan Mancarci and Daniel Padfield

References

https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

See Also

The core functions in this package: staple_pdf, remove_pages, split_pdf, rename_files

Index

combine_pdf, 2

get_fields, 3, *11, 12*

identify_form_fields, 4

remove_pages, 5, *16*
rename_files, 6, *16*
rotate_pages, 7
rotate_pdf, 9

select_pages, 10
set_fields, *4, 5*, 11
split_from, 13
split_pdf, 14, *16*
staple_pdf, 15, *16*
staplr, 16