

# Package ‘tabulapdf’

May 21, 2024

**Type** Package

**Title** Extract Tables from PDF Documents

**Description** Bindings for the 'Tabula' <<https://tabula.technology/>> 'Java' library, which can extract tables from PDF files. This tool can reduce time and effort in data extraction processes in fields like investigative journalism. It allows for automatic and manual table extraction, the latter facilitated through a 'Shiny' interface, enabling manual areas selection\ with a computer mouse for data retrieval.

**Version** 1.0.5-3

**License** Apache License (>= 2)

**URL** <https://docs.ropensci.org/tabulapdf/> (website)  
<https://github.com/ropensci/tabulapdf/>

**BugReports** <https://github.com/ropensci/tabulapdf/issues/>

**Imports** png, readr, rJava, tools, utils

**Suggests** graphics, grDevices, knitr, miniUI, shiny, testthat, rmarkdown, covr

**SystemRequirements** Java (>= 7.0): openjdk-11-jdk (deb), java-11-openjdk.x86\_64 (rpm), openjdk@11 (brew)

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Thomas J. Leeper [aut] (<<https://orcid.org/0000-0003-4097-6326>>),  
Mauricio Vargas Sepulveda [aut, cre]  
(<<https://orcid.org/0000-0003-1017-7574>>),  
Tom Paskhalis [aut] (<<https://orcid.org/0000-0001-9298-8850>>),  
Manuel Aristaran [ctb],  
David Gohel [ctb] (rOpenSci reviewer),  
Lincoln Mullen [ctb] (rOpenSci reviewer)

**Maintainer** Mauricio Vargas Sepulveda <m.sepulveda@mail.utoronto.ca>

**Repository** CRAN

**Date/Publication** 2024-05-21 15:20:03 UTC

**R topics documented:**

tabulapdf-package . . . . .	2
extract_metadata . . . . .	3
extract_tables . . . . .	4
extract_text . . . . .	6
get_page_dims . . . . .	7
locate_areas . . . . .	9
make_thumbnails . . . . .	11
split_pdf . . . . .	12
stop_logging . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

tabulapdf-package	<i>tabulapdf</i>
-------------------	------------------

---

**Description**

Bindings for “Tabula” PDF Table Extractor Library

**Details**

Tabula is a Java library designed to computationally extract tables from PDF documents. `tabulapdf` provides a thin R package with bindings to the library. It presently offers two principal functions: [extract\\_tables](#), which mimics the command line functionality of Tabula, and [extract\\_areas](#) which provides an interactive interface to the former.

**Author(s)**

Thomas J. Leeper <thosjleeper@gmail.com>

**References**

[tabula](#)

**See Also**

[extract\\_tables](#), [extract\\_areas](#)

---

extract_metadata	<i>extract_metadata</i>
------------------	-------------------------

---

## Description

Extract metadata from a file

## Usage

```
extract_metadata(file, password = NULL, copy = FALSE)
```

## Arguments

file	A character string specifying the path or URL to a PDF file.
password	Optionally, a character string containing a user password to access a secured PDF.
copy	Specifies whether the original local file(s) should be copied to <code>tempdir()</code> before processing. FALSE by default. The argument is ignored if <code>file</code> is URL.

## Details

This function extracts metadata from a PDF

## Value

A list.

## Author(s)

Thomas J. Leeper <thosjleeper@gmail.com>

## See Also

[extract\\_tables](#), [extract\\_areas](#), [extract\\_text](#), [split\\_pdf](#)

## Examples

```
# simple demo file
f <- system.file("examples", "mtcars.pdf", package = "tabulapdf")

extract_metadata(f)
```

---

extract_tables	<i>extract_tables</i>
----------------	-----------------------

---

## Description

Extract tables from a file

## Usage

```
extract_tables(
  file,
  pages = NULL,
  area = NULL,
  columns = NULL,
  col_names = TRUE,
  guess = TRUE,
  method = c("decide", "lattice", "stream"),
  output = c("tibble", "matrix", "character", "asis", "csv", "tsv", "json"),
  outdir = NULL,
  password = NULL,
  encoding = NULL,
  copy = FALSE,
  ...
)
```

## Arguments

file	A character string specifying the path or URL to a PDF file.
pages	An optional integer vector specifying pages to extract from.
area	An optional list, of length equal to the number of pages specified, where each entry contains a four-element numeric vector of coordinates (top,left,bottom,right) containing the table for the corresponding page. As a convenience, a list of length 1 can be used to extract the same area from all (specified) pages. Only specify area or columns. Warning: area is ignored if guess is TRUE.
columns	An optional list, of length equal to the number of pages specified, where each entry contains a numeric vector of horizontal (x) coordinates separating columns of data for the corresponding page. As a convenience, a list of length 1 can be used to specify the same columns for all (specified) pages. Only specify area or columns. Warning: columns is ignored if guess is TRUE.
col_names	A logical indicating whether to include column names in the output tibbles. Default is TRUE.
guess	A logical indicating whether to guess the locations of tables on each page. If FALSE, area or columns must be specified; if TRUE, area and columns are ignored.
method	A string identifying the preferred method of table extraction.

	<ul style="list-style-type: none"> <li>• <code>method = "decide"</code> (default) automatically decide (for each page) whether spreadsheet-like formatting is present and "lattice" is appropriate</li> <li>• <code>method = "lattice"</code> use Tabula's spreadsheet extraction algorithm</li> <li>• <code>method = "stream"</code> use Tabula's basic extraction algorithm</li> </ul>
output	A function to coerce the Java response object (a Java ArrayList of Tabula Tables) to some output format. The default method, "matrices", returns a list of character matrices. See Details for other options.
outdir	Output directory for files if output is set to "csv", "tsv" or "json", ignored otherwise. If equals NULL (default), uses R sessions temporary directory <code>tempdir()</code> .
password	Optionally, a character string containing a user password to access a secured PDF.
encoding	Optionally, a character string specifying an encoding for the text, to be passed to the assignment method of <a href="#">Encoding</a> .
copy	Specifies whether the original local file(s) should be copied to <code>tempdir()</code> before processing. FALSE by default. The argument is ignored if <code>file</code> is URL.
...	These are additional arguments passed to the internal functions dispatched by <code>method</code> .

### Details

This function mimics the behavior of the Tabula command line utility. It returns a list of R character matrices containing tables extracted from a file by default. This response behavior can be changed by using the following options.

- `output = "tibble"` attempts to coerce the structure returned by `method = "character"` into a list of tibbles and returns character strings where this fails.
- `output = "character"` returns a list of single-element character vectors, where each vector is a tab-delimited, line-separate string of concatenated table cells.
- `output = "csv"` writes the tables to comma-separated (CSV) files using Tabula's `CSVWriter` method in the same directory as the original PDF. `method = "tsv"` does the same but with tab-separated (TSV) files using Tabula's `TSVWriter` and `method = "json"` does the same using Tabula's `JSONWriter` method. Any of these three methods return the path to the directory containing the extract table files.
- `output = "asis"` returns the Java object reference, which can be useful for debugging or for writing a custom parser.

[extract\\_areas](#) implements this functionality in an interactive mode allowing the user to specify extraction areas for each page.

### Value

By default, a list of character matrices. This can be changed by specifying an alternative value of `method` (see Details).

### Author(s)

Thomas J. Leeper <thosjleeper@gmail.com>, Tom Paskhalis <tpaskhalis@gmail.com>

**References**[Tabula](#)**See Also**[extract\\_areas](#), [get\\_page\\_dims](#), [make\\_thumbnails](#), [split\\_pdf](#)**Examples**

```
# simple demo file
f <- system.file("examples", "mtcars.pdf", package = "tabulapdf")

# extract tables from only second page
extract_tables(f, pages = 2)
```

---

`extract_text`*extract\_text*

---

**Description**

Extract text from a file

**Usage**

```
extract_text(
  file,
  pages = NULL,
  area = NULL,
  password = NULL,
  encoding = NULL,
  copy = FALSE
)
```

**Arguments**

<code>file</code>	A character string specifying the path or URL to a PDF file.
<code>pages</code>	An optional integer vector specifying pages to extract from.
<code>area</code>	An optional list, of length equal to the number of pages specified, where each entry contains a four-element numeric vector of coordinates (top,left,bottom,right) containing the table for the corresponding page. As a convenience, a list of length 1 can be used to extract the same area from all (specified) pages.
<code>password</code>	Optionally, a character string containing a user password to access a secured PDF.
<code>encoding</code>	Optionally, a character string specifying an encoding for the text, to be passed to the assignment method of <a href="#">Encoding</a> .
<code>copy</code>	Specifies whether the original local file(s) should be copied to <code>tempdir()</code> before processing. FALSE by default. The argument is ignored if <code>file</code> is URL.

**Details**

This function converts the contents of a PDF file into a single unstructured character string.

**Value**

If pages = NULL (the default), a length 1 character vector, otherwise a vector of length length(pages).

**Author(s)**

Thomas J. Leeper <thosjleeper@gmail.com>

**See Also**

[extract\\_tables](#), [extract\\_areas](#), [split\\_pdf](#)

**Examples**

```
# simple demo file
f <- system.file("examples", "fortytwo.pdf", package = "tabulapdf")

# extract all text
extract_text(f)

# extract all text from page 1 only
extract_text(f, pages = 1)

# extract text from selected area only
extract_text(f, area = list(c(209.4, 140.5, 304.2, 500.8)))
```

---

get\_page\_dims

*Page length and dimensions*

---

**Description**

Get Page Length and Dimensions

**Usage**

```
get_page_dims(file, doc, pages = NULL, password = NULL, copy = FALSE)

get_n_pages(file, doc, password = NULL, copy = FALSE)
```

## Arguments

file	A character string specifying the path or URL to a PDF file.
doc	Optionally, in lieu of file, an rJava reference to a PDDocument Java object.
pages	An optional integer vector specifying pages to extract from.
password	Optionally, a character string containing a user password to access a secured PDF.
copy	Specifies whether the original local file(s) should be copied to tempdir() before processing. FALSE by default. The argument is ignored if file is URL.

## Details

get\_n\_pages returns the page length of a PDF document. get\_page\_dims extracts the dimensions of specified pages in a PDF document. This can be useful for figuring out how to specify the area argument in [extract\\_tables](#)

## Value

For get\_n\_pages, an integer. For get\_page\_dims, a list of two-element numeric vectors specifying the width and height of each page, respectively.

## Author(s)

Thomas J. Leeper <thosjleeper@gmail.com>

## References

[Tabula](#)

## See Also

[extract\\_tables](#), [extract\\_text](#), [make\\_thumbnails](#)

## Examples

```
# simple demo file
f <- system.file("examples", "mtcars.pdf", package = "tabulapdf")

get_n_pages(file = f)
get_page_dims(f)
```



---

locate_areas	<i>extract_areas</i>
--------------	----------------------

---

## Description

Interactively identify areas and extract

## Usage

```
locate_areas(
  file,
  pages = NULL,
  resolution = 60L,
  widget = c("shiny", "native", "reduced"),
  copy = FALSE
)

extract_areas(file, pages = NULL, guess = FALSE, copy = FALSE, ...)
```

## Arguments

file	A character string specifying the path to a PDF file. This can also be a URL, in which case the file will be downloaded to the R temporary directory using <code>download.file</code> .
pages	An optional integer vector specifying pages to extract from. To extract multiple tables from a given page, repeat the page number (e.g., <code>c(1, 2, 2, 3)</code> ).
resolution	An integer specifying the resolution of the PNG images conversions. A low resolution is used by default to speed image loading.
widget	A one-element character vector specifying the type of “widget” to use for locating the areas. The default (“shiny”) is a shiny widget. The alternatives are a widget based on the native R graphics device (“native”, where available), or a very reduced functionality model (“reduced”).
copy	Specifies whether the original local file(s) should be copied to <code>tempdir()</code> before processing. FALSE by default. The argument is ignored if <code>file</code> is URL.
guess	See <a href="#">extract_tables</a> (note the different default value).
...	Other arguments passed to <a href="#">extract_tables</a> .

## Details

`extract_areas` is an interactive mode for [extract\\_tables](#) allowing the user to specify areas of each PDF page in a file that they would like extracted. When used, each page is rendered to a PNG file and displayed in an R graphics window sequentially, pausing on each page to call [locator](#) so the user can click and highlight an area to extract.

The exact behaviour is a somewhat platform-dependent, and depends on the value of `widget` (and further, whether you are working in RStudio or the R console). In RStudio (where `widget =`

"shiny"), a Shiny gadget is provided which allows the user to click and drag to select areas on each page of a file, clicking "Done" on each page to advance through them. It is not possible to return to previous pages. In the R console, a Shiny app will be launched in a web browser.

For other values of `widget`, functionality is provided through the graphics device. If graphics events are supported, then it is possible to interactively highlight a page region, make changes to that region, and navigate through the pages of the document while retaining the area highlighted on each page. If graphics events are not supported, then some of this functionality is not available (see below).

In *full functionality mode* (`widget = "native"`), areas are input in a native graphics device. For each page, the first mouse click on a page initializes a highlighting rectangle; the second click confirms it. If unsatisfied with the selection, the process can be repeated. The window also responds to keystrokes. `PgDn`, `Right`, and `Down` advance to the next page image, while `PgUp`, `Left`, and `Up` return to the previous page image. `Home` returns to the first page image and `End` advances to the final page image. `Q` quits the interactive mode and proceeds with extraction. When navigating between pages, any selected areas will be displayed and can be edited. `Delete` removes a highlighted area from a page (and then displays it again). (This mode may not work correctly from within RStudio.)

In *reduced functionality mode* (where `widget = "reduced"` or on platforms where graphics events are unavailable), the interface requires users to indicate the upper-left and lower-right (or upper-right and lower-left) corners of an area on each page, this area will be briefly confirmed with a highlighted rectangle and the next page will be displayed. Dynamic page navigation and area editing are not possible.

In any of these modes, after the areas are selected, `extract_areas` passes these user-defined areas to `extract_tables`. `locate_areas` implements the interactive component only, without actually extracting; this might be useful for interactive work that needs some modification before executing `extract_tables` computationally.

## Value

For `extract_areas`, see `extract_tables`. For `locate_areas`, a list of four-element numeric vectors (`top`,`left`,`bottom`,`right`), one per page of the file.

## Author(s)

Thomas J. Leeper <thosjleeper@gmail.com>

## See Also

[extract\\_tables](#), [make\\_thumbnails](#), [get\\_page\\_dims](#)

## Examples

```
if (interactive()) {
  # simple demo file
  f <- system.file("examples", "mtcars.pdf", package = "tabulapdf")

  # locate areas only, using Shiny app
  locate_areas(f)

  # locate areas only, using native graphics device
```

```
locate_areas(f, widget = "shiny")

# locate areas and extract
extract_areas(f)
}
```

---

make\_thumbnails      *make\_thumbnails*

---

## Description

Convert Pages to Image Thumbnails

## Usage

```
make_thumbnails(
  file,
  outdir = NULL,
  pages = NULL,
  format = c("png", "jpeg", "bmp", "gif"),
  resolution = 72,
  password = NULL,
  copy = FALSE
)
```

## Arguments

file	A character string specifying the path or URL to a PDF file.
outdir	An optional character string specifying a directory into which to split the resulting files. If NULL, the outdir is <code>tempdir()</code> . If file is a URL, both file and thumbnails are stored in the R session's temporary directory.
pages	An optional integer vector specifying pages to extract from.
format	A character string specifying an image file format.
resolution	A numeric value specifying the image resolution in DPI.
password	Optionally, a character string containing a user password to access a secured PDF.
copy	Specifies whether the original local file(s) should be copied to <code>tempdir()</code> before processing. FALSE by default. The argument is ignored if file is URL.

## Details

This function save each (specified) page of a document as an image with 720 dpi resolution. Images are saved in the same directory as the original file, with file names specified by the original file name, a page number, and the corresponding file format extension.

**Value**

A character vector of file paths.

**Note**

This may generate Java “INFO” messages in the console, which can be safely ignored.

**Author(s)**

Thomas J. Leeper <thosjleeper@gmail.com>

**References**

[Tabula](#)

**See Also**

[extract\\_tables](#), [extract\\_text](#), [make\\_thumbnails](#)

**Examples**

```
# simple demo file
f <- system.file("examples", "mtcars.pdf", package = "tabulapdf")

make_thumbnails(f)
```

---

split\_pdf

*Split and merge PDFs*

---

**Description**

Split PDF into separate pages or merge multiple PDFs into one.

**Usage**

```
split_pdf(file, outdir = NULL, password = NULL, copy = FALSE)
```

```
merge_pdfs(file, outfile, copy = FALSE)
```

**Arguments**

file	For <code>merge_pdfs</code> , a character vector specifying the path to one or more <i>local</i> PDF files. For <code>split_pdf</code> , a character string specifying the path or URL to a PDF file.
outdir	For <code>split_pdf</code> , an optional character string specifying a directory into which to split the resulting files. If <code>NULL</code> , the <code>outdir</code> is <code>tempdir()</code> . If <code>file</code> is a URL, both the original file and separate pages are stored in the R session’s temporary directory.

password	Optionally, a character string containing a user password to access a secured PDF. Currently, encrypted PDFs cannot be merged with <code>merge_pdfs</code> .
copy	Specifies whether the original local file(s) should be copied to <code>tempdir()</code> before processing. FALSE by default. The argument is ignored if <code>file</code> is URL.
outfile	For <code>merge_pdfs</code> , a character string specifying the path to the PDF file to create from the merged documents.

### Details

[split\\_pdf](#) splits the file listed in `file` into separate one-page documents. [merge\\_pdfs](#) creates a single PDF document from multiple separate PDF files.

### Value

For `split_pdfs`, a character vector specifying the output file names, which are patterned after the value of `file`. For `merge_pdfs`, the value of `outfile`.

### Author(s)

Thomas J. Leeper <[thosjleeper@gmail.com](mailto:thosjleeper@gmail.com)>

### See Also

[extract\\_areas](#), [get\\_page\\_dims](#), [make\\_thumbnails](#)

### Examples

```
# simple demo file
f <- system.file("examples", "mtcars.pdf", package = "tabulapdf")
get_n_pages(file = f)

# split PDF by page
sf <- split_pdf(f)

# merge pdf
mf <- file.path(tempdir(), "merged.pdf")
merge_pdfs(sf, mf)
get_n_pages(mf)
```

---

stop\_logging

*rJava logging*

---

### Description

Toggle verbose rJava logging

### Usage

```
stop_logging()
```

**Details**

This function turns off the somewhat verbose rJava logging, most of which is uninformative. It is called automatically when `tabulapdf` is attached via `library()`, `require`, etc. To keep logging on, load the package namespace using `requireNamespace("tabulapdf")` and reference functions in using fully qualified references (e.g., `tabulapdf::extract_tables()`).

**Value**

NULL, invisibly.

**Note**

This resets a global Java setting and may affect logging of other rJava operations, requiring a restart of R.

**Author(s)**

Thomas J. Leeper <thosjleeper@gmail.com>

**Examples**

```
stop_logging()
```

# Index

Encoding, [5](#), [6](#)

extract\_areas, [2](#), [3](#), [5–7](#), [13](#)

extract\_areas (locate\_areas), [9](#)

extract\_metadata, [3](#)

extract\_tables, [2](#), [3](#), [4](#), [7–10](#), [12](#)

extract\_text, [3](#), [6](#), [8](#), [12](#)

get\_n\_pages (get\_page\_dims), [7](#)

get\_page\_dims, [6](#), [7](#), [10](#), [13](#)

locate\_areas, [9](#)

locator, [9](#)

make\_thumbnails, [6](#), [8](#), [10](#), [11](#), [12](#), [13](#)

merge\_pdfs, [13](#)

merge\_pdfs (split\_pdf), [12](#)

split\_pdf, [3](#), [6](#), [7](#), [12](#), [13](#)

stop\_logging, [13](#)

tabulapdf (tabulapdf-package), [2](#)

tabulapdf-package, [2](#)