

**putpdf begin** — Create a PDF file[Description](#)[Remarks and examples](#)[Quick start](#)[References](#)[Syntax](#)[Also see](#)[Options](#)

## Description

`putpdf begin` creates a PDF file. This is the active document that the remaining `putpdf` commands modify.

`putpdf describe` describes the active PDF file.

`putpdf save` saves and closes the PDF file.

`putpdf clear` closes the PDF file without saving.

## Quick start

Create a document in memory onto which subsequent contents are added

```
putpdf begin
```

Same as above, but with 1.5 inch margins on the left and right

```
putpdf begin, margin(left,1.5) margin(right,1.5)
```

Save the document in memory to disk as `myfile.pdf`

```
putpdf save myfile.pdf
```

Same as above, but overwrite `myfile.pdf` if it already exists

```
putpdf save myfile.pdf, replace
```

Close the document in memory without saving the changes

```
putpdf clear
```

## Syntax

Create document for export

```
putpdf begin [ , begin_options ]
```

Describe active document

```
putpdf describe
```

Save and close document

```
putpdf save filename [ , replace nomsg ]
```

Close without saving

```
putpdf clear
```

<i>begin_options</i>	Description
<code>pagesize(<i>psize</i>)</code>	set document page size
<code>landscape</code>	change document orientation to landscape
<code>font(<i>fspec</i>)</code>	set font, font size, and font color for the document
<code>halign(<i>hvalue</i>)</code>	set horizontal alignment for the document
<code>margin(<i>type</i>, # [<i>unit</i> ])</code>	set page margins for the document
<code>bgcolor(<i>color</i>)</code>	set background color

`collect` is allowed with `putpdf describe`; see [U] [11.1.10 Prefix commands](#).

## Options

Options are presented under the following headings:

[Options for putpdf begin](#)

[Options for putpdf save](#)

### Options for putpdf begin

`pagesize(psize)` sets the page size of the document. *psize* may be `letter`, `legal`, `A3`, `A4`, `A5`, `B4`, or `B5`. The default is `pagesize(letter)`.

`landscape` changes the document orientation from portrait (the default) to landscape.

`font(fontname [ , size [ , color ] ])` sets the font, font size, and font color for the document.

*fontname* may be any supported font installed on the user's computer. Base 14 fonts, Type 1 fonts (`.pfa` or `.pfb`), TrueType fonts (`.ttf` or `.ttc`), and OpenType fonts (`.otf`) are supported. TrueType and OpenType fonts that cannot be embedded may not be used. If *fontname* includes spaces, then it must be enclosed in double quotes. The default font is Helvetica.

*size* is a numeric value that represents font size measured in points. The default is 11.

*color* sets the text color. *color* may be one of the colors listed in *Colors* of [RPT] [Appendix for putpdf](#); a valid RGB value in the form `### ## #`, for example, 171 248 103; or a valid RRGGBB hex value in the form `#####`, for example, ABF867.

The font size and font color may be specified individually without specifying *fontname*. Use `font("", size)` to specify font size only. Use `font("", "", color)` to specify font color only. For both cases, the default font will be used.

`halign(hvalue)` sets the horizontal alignment of the document within the paragraphs, images, and tables. *hvalue* may be `left`, `right`, or `center`. The default is `halign(left)`.

`margin(type, #[unit])` sets the page margins of the document. This option may be specified multiple times in a single command to account for different margin settings.

*type* identifies the location of the margin inside the document. *type* may be `top`, `left`, `bottom`, `right`, or `all`.

*unit* may be `in` (inch), `pt` (point), `cm` (centimeter), or `twip` (twentieth of a point). An inch is equivalent to 72 points, 2.54 centimeters, or 1440 twips. The default is `in`.

`bgcolor(color)` sets the background color for the document. *color* may be one of the colors listed in *Colors* of [RPT] [Appendix for putpdf](#); a valid RGB value in the form `### ## #`, for example, 171 248 103; or a valid RRGGBB hex value in the form `#####`, for example, ABF867.

## Options for putpdf save

`replace` specifies to overwrite *filename*, if it exists, with the contents of the document in memory.

`nomsg` suppresses the message that contains a link to *filename*.

## Remarks and examples

[stata.com](http://stata.com)

Remarks are presented under the following headings:

*Creating and formatting a PDF file*

*Describing the document*

*Saving or clearing the PDF file*

## Creating and formatting a PDF file

Before we can write to a PDF file using `putpdf`, we need to create an active PDF document in memory by using the `putpdf begin` command. We can simply type

```
. putpdf begin
```

to create a document. We could now add content to this document. For information on adding text or images to the document, see [RPT] [putpdf paragraph](#). For information on adding tables to the document, see [RPT] [putpdf table](#).

Because we did not include any options in the above `putpdf begin` command, it creates a letter-size document with pages in portrait orientation. We can specify other formats for the document as a whole by using the options available with `putpdf begin`. We can specify the page size, page orientation, and font properties for the document.

For example, below we create a legal-sized PDF file in memory with the font Courier.

```
. putpdf begin, pagesize(legal) font("Courier")
```

The page size and orientation specified with `putpdf begin` will remain in effect until a [section break](#) is added. On the other hand, the font specification can be modified for every [paragraph](#) and addition of [text](#).

## Describing the document

To view a document, we must first save the file. However, if we have been adding text, images, and tables to a document, we can describe the contents of the active document without saving it.

```
. putpdf describe
```

This reports the number of paragraphs and tables that have been added to the document.

## Saving or clearing the PDF file

When we have finished adding content to our document, we can save it under a given filename, say, `myfile.pdf`.

```
. putpdf save myfile.pdf
```

If the file already exists in the saving directory, we will need to specify the `replace` option, which will overwrite the existing contents in the file.

Suppose we have mistakenly added an image with the wrong dimensions. Rather than saving the document in this situation, we can clear the active document from memory without saving it by typing

```
. putpdf clear
```

This command clears the document in memory and automatically closes the document without saving.

## References

- Jann, B. 2016. [Creating L<sup>A</sup>T<sub>E</sub>X documents from within Stata using texdoc](#). *Stata Journal* 16: 245–263.
- Rodríguez, G. 2017. [Literate data analysis with Stata and Markdown](#). *Stata Journal* 17: 600–618.
- Weinreb, M. D., and J. Trinitapoli. 2022. [printcase: A command for visualizing single observations](#). *Stata Journal* 22: 958–968.

## Also see

- [RPT] [putpdf intro](#) — Introduction to generating PDF files
- [RPT] [putpdf collect](#) — Add a table from a collection to a PDF file
- [RPT] [putpdf pagebreak](#) — Add breaks to a PDF file
- [RPT] [putpdf paragraph](#) — Add text or images to a PDF file
- [RPT] [putpdf table](#) — Add tables to a PDF file
- [RPT] [Appendix for putpdf](#) — Appendix for `putpdf` entries

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).