Title

stata.com

putdocx begin — Create an Office Open XML (.docx) file

Description

`putdocx begin` creates an Office Open XML (`.docx`) file. This is the active document that the remaining `putdocx` commands modify.

`putdocx describe` describes the active `.docx` file.

`putdocx save` saves and closes the `.docx` file.

`putdocx clear` closes the `.docx` file without saving.

`putdocx append` appends the contents of one or more `.docx` files to another `.docx` file.

Quick start

Create a document in memory onto which subsequent contents are added

```
putdocx begin
```

Create a document with page numbers in the footer named `footer1`, using uppercase Roman numerals

```
putdocx begin, pagenum(upper_roman) footer(footer1)
```

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

```
putdocx save myfile
```

Append the contents of `filename2.docx` and `filename3.docx` to the end of the contents in `filename1.docx`

```
putdocx append filename1 filename2 filename3
```

As above, but save the resulting document in a file named `filename4.docx`

```
putdocx append filename1 filename2 filename3, saving(filename4)
```

Syntax

Create document for export

```
putdocx begin [ , begin_options ]
```

Describe active document

```
putdocx describe
```

Save and close document

```
putdocx save filename [ , save_options ]
```

Close without saving

```
putdocx clear
```

Append contents of documents

```
putdocx append filename1 filename2 [filename3 [ ... ]] [ , append_options ]
```

begin_options	Description
`pagesize(`*psize*`)`	set document page size
`landscape`	change document orientation to landscape
`font(`*fspec*`)`	set font, font size, and font color for the document
`pagenum(`*pnspec*`)`	set page number format
`header(`*hname*`)`	add a header
`footer(`*fname*`)`	add a footer

save_options	Description
`replace`	replace *filename* with the active document
`append`	append the active document to the end of *filename*
`append(`*ap_opts*`)`	append active document to *filename* and change page break, header, and footer settings
`nomsg`	suppress message with a link to *filename*

Only one of `replace`, `append`, or `append(`*ap_opts*`)` may be specified.

append_options	Description
saving(*filename* [, replace])	save the document to *filename*; use replace to overwrite existing *filename*
pagebreak	begin each appended file on a new page
headsrc(first | last | own)	specify the document from which headers and footers are to be used; default is headsrc(first)
pgnumrestart	restart page numbering on the first page of each appended document
nomsg	suppress message with a link to resulting document

Options

Options are presented under the following headings:

Options for putdocx begin
Options for putdocx save
Options for putdocx append

Options for putdocx begin

pagesize(*psize*) sets the page size of the document. *psize* may be letter, legal, A3, A4, or B4JIS. 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 valid font installed on the user's computer. If *fontname* includes spaces, then it must be enclosed in double quotes. If *fontname* is not specified, the computer's default font will be used.

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 putdocx**; 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 the font size only. Use font("", "", *color*) to specify the font color only.

pagenum(*pnformat* [, *start* [, *chapStyle* [, *chapSep*]]]) specifies the format and starting page for page numbers.

pnformat specifies the page number format, such as decimals enclosed in parentheses or uppercase Roman numerals. The default is decimal. For a complete list of page number formats, see *Page number formats* of [RPT] **Appendix for putdocx**.

start specifies the starting page number and must be an integer greater than or equal to 0. The default is 1.

chapStyle specifies the style used for chapter headings. For a complete list of chapter styles, see *Chapter styles* of [RPT] **Appendix for putdocx**.

chapSep specifies the symbol used to separate chapter numbers and page numbers. *chapSep* may be colon, hyphen, em_dash, en_dash, or period. The default is hyphen.

This option is not required for including page numbers in your document, unless you want to include chapter numbers as well. To include chapter numbers, specify the style used to indicate chapters (*chapStyle*), and optionally, the symbol used to separate chapter and page numbers.

When specifying *chapStyle* and *chapSep*, chapter numbers will be reported along with the page numbers. To activate these options, you must specify a multilevel list style in Word that includes headings.

`header(`*hname*`)` adds the header named *hname* to the document. The content of *hname*, including page numbers, can be defined with either `putdocx paragraph` or `putdocx table`. *hname* must be a valid name according to Stata's naming conventions; see **[U] 11.3 Naming conventions**.

`footer(`*fname*`)` adds the footer named *fname* to the document. The content of *fname*, including page numbers, can be defined with either `putdocx paragraph` or `putdocx table`. *fname* must be a valid name according to Stata's naming conventions; see **[U] 11.3 Naming conventions**.

Options for putdocx save

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

`append` specifies to append the contents of the document in memory to the end of *filename*.

`append(`*ap_opts*`)` specifies to append the contents of the document in memory to *filename* and indicates whether new content will be added on a new page, which header and footer to use, and whether to restart page numbering for each document. *ap_opts* are `pagebreak`, `headsrc()`, and `pgnumrestart`.

`pagebreak` appends the active document beginning on a new page.

`headsrc(file | active | own)` specifies the file whose header and footer will be used in the document.

`headsrc(file)` is the default; it applies the header and footer from *filename* throughout the document.

`headsrc(active)` applies the header and footer from the active document throughout the document.

`headsrc(own)` specifies that each document, *filename* and the active document, use its own header and footer. If the active document does not have a header or footer, it will inherit the header or footer from *filename*.

`pgnumrestart` restarts page numbering in each document; `pagebreak` must be specified with `pgnumrestart`.

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

Options for putdocx append

`saving(`*filename* [`, replace`]`)` specifies to append the contents of the existing document $filename_2$ to the end of $filename_1$ and then write the result to the new document *filename*. If *filename* already exists, it can be overwritten by specifying `replace`. By default, $filename_1$ is overwritten with the document created by appending content from $filename_2$.

If more than two files are specified, the contents are appended in the order in which the files are listed. For example, $filename_2$ is appended to $filename_1$, $filename_3$ is appended to the result of the first append, and so forth.

pagebreak begins each appended file on a new page.

headsrc(first | last | own) specifies the document from which headers and footers are to be used.

headsrc(first), the default, specifies that the header and footer from the first document being appended, *filename*$_1$, be applied throughout the document.

headsrc(last) specifies that the header and footer from the last document be applied throughout the document. headsrc(last) may not be specified with pgnumrestart.

headsrc(own) specifies that each file being appended use its own header and footer. If any file does not have a header or footer, it will inherit the header or footer from the previous document.

pgnumrestart restarts page numbering on the first page of each appended document. The pagebreak option must be specified with pgnumrestart. This option may not be specified with headsrc(last).

nomsg suppresses the message that contains a link to the resulting document.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Creating and formatting a .docx file

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

```
. putdocx 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] **putdocx paragraph**. For information on adding tables to the document, see [RPT] **putdocx table**.

Because we did not include any options in the above putdocx begin command, it creates a letter-size document with pages in portrait orientation. It uses our computer's default font type and font color in 11-point size. We can specify other formats for the document as a whole by using the options available with putdocx begin. We can specify the page size, page orientation, and font properties for the document. We can also include page numbers, headers, and footers.

For example, to create a document with letter-size pages in landscape orientation and 11-point Garamond font, we type

```
. putdocx begin, landscape font(Garamond)
```

The page size and orientation set with putdocx begin remain in effect until a section break is added.

The font properties specified with putdocx begin remain in effect throughout the document, but we can change these properties for each paragraph (or each sentence or even each word) with the options available in [RPT] **putdocx paragraph**. When specified without options, putdocx paragraph and putdocx text will default to the font properties specified in putdocx begin. Any text added with putdocx textblock will also default to the font properties specified in putdocx begin, unless we specify some other format by using dynamic tags within the text block.

Headers and footers remain in effect throughout the document by default, but they can be changed when a new section is added to the document.

Including headers and footers

To create a document that includes a header, footer, or page numbers requires multiple steps. First, we need to include either the `header()` or `footer()` option in our `putdocx begin` command, and if desired, we can customize the page numbers using the `pagenum()` option. Then we use either `putdocx paragraph` or `putdocx table` to add content, including page numbers, to the header or footer.

For instance, suppose we are creating an academic progress report for a local high school, and we want to include a footer with the page number and school name. We can create our document as follows:

```
. putdocx begin, pagenum(decimal) footer(npage)
```

In this command, we specified the decimal format for the page numbers, and we added a blank footer to our document.

Now, we can define the contents of the footer `npage` by using `putdocx paragraph`.

```
. putdocx paragraph, tofooter(npage)
. putdocx text ("Mountain High Report: ")
. putdocx pagenumber
```

This creates a paragraph with the school name and page number. The `tofooter()` option directed the content of this paragraph to the footer, `npage`, rather than including it in the body of the document.

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.

```
. putdocx describe
```

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

Saving or clearing the .docx file

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

```
. putdocx save myfile, replace
```

The `replace` option specifies that we wish to overwrite the contents of an existing `myfile.docx`.

If the current document is a continuation of work stored in an existing file—say, `oldfile.docx`—then we can instead append the current document's contents to the existing file by typing

```
. putdocx save oldfile, append
```

This command adds the content of the active document that we have been working on to `oldfile.docx`, beginning directly below its original content. The new combined document is saved as `oldfile.docx`.

Issuing the `putdocx save` command automatically clears the active document from memory and closes it after it is saved.

Sometimes, we may want to clear the active document from memory without saving it. Perhaps we accidentally added a table with the wrong numeric formatting or we forgot to label a graph. We can type

```
. putdocx clear
```

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

`putdocx clear` is especially important if we are running a do-file with a series of `putdocx` commands repeatedly. Maybe we have `putdocx begin` at the top of our do-file and when we try to run it again, we get an error message. We must either save our work or close the document with `putdocx clear` before we can issue `putdocx begin` again.

Appending .docx files

Sometimes, it is easiest to create a final document by appending multiple `.docx` files. Perhaps you create parts of a report directly within Word and create other parts using `putdocx`. Perhaps you prefer to work on your document out of order and then compile the parts at the end. In these cases, we can use the `putdocx append` command.

Say we have four files that we are now ready to merge: `part1.docx`, `part2.docx`, `part3.docx`, and `part4.docx`. We can type

```
. putdocx append part1 part3 part2 part4, saving(allparts)
```

`putdocx append` appends files in the order in which they are specified, so the above command will append `part3` to the end of `part1`, `part2` to the end of `part3`, and `part4` to the end of `part2`. We also specified to save the resulting document in a new file called `allparts.docx`.

Oops! `part2` should have come before `part3` in the merged file. We can fix that by typing

```
. putdocx append part1 part2 part3 part4, saving(allparts, replace)
```

By including `replace` in the `saving()` option, we request that the resulting document overwrite the existing contents of `allparts.docx`.

If we do not use the `saving()` option, for example,

```
. putdocx append part1 part2 part3 part4
```

then the merged document is saved in the first-listed file, namely, `part1.docx`.

Also see

[RPT] **putdocx intro** — Introduction to generating Office Open XML (.docx) files

[RPT] **putdocx pagebreak** — Add breaks to an Office Open XML (.docx) file

[RPT] **putdocx paragraph** — Add text or images to an Office Open XML (.docx) file

[RPT] **putdocx table** — Add tables to an Office Open XML (.docx) file

[RPT] **Appendix for putdocx** — Appendix for putdocx entries