

markdoc v. 5.0 package vignette

E. F. Haghish

June 2, 2019

Contents

I	markdoc package commands	4
1	markdoc	5
1.1	Syntax	5
1.1.1	Options	5
1.2	Additional commands	6
1.3	Installation	7
1.4	Description	7
1.5	Software Installation	7
1.6	Set file paths permanently	8
1.7	Software troubleshoot	8
1.8	Calling Pandoc	8
1.9	Writing mathematical notation	8
1.10	Inserting an image or figure in the document	8
1.11	Writing dynamic text	9
1.12	Creating dynamic tables with tbl command	9
1.13	Markers	9
1.14	Markers examples	10
1.14.1	Example of riting text in the do-file	10
1.15	Dynamic Document Examples	10
1.16	Dynamic Slide Examples	11
1.17	Supported markup languages	11
1.18	Remarks	11
1.19	Author	12
1.20	See also	12
1.21	License	12
2	mini	12
2.1	Syntax	12
2.2	Description	12
2.3	Examples	12
2.4	Author	12
2.5	License	13
3	pandoc	13
3.1	Syntax	13
3.2	Description	13
3.3	Examples	13
3.4	Author	13
3.5	License	13

4	wkhtmltopdf	14
4.1	Syntax	14
4.2	Description	14
4.3	Example	14
4.4	Author	14
4.5	License	14
II	Additional commands	15
5	txt	16
5.1	Syntax	16
5.2	Description	16
5.3	Display directives	17
5.4	Mathematical notations	17
5.5	Examples	17
5.6	Author	18
6	img	18
6.1	Syntax	18
6.1.1	Options	18
6.2	Description	19
6.3	Examples	19
6.4	Author	19
7	tbl	19
7.1	Syntax	20
7.1.1	Options	21
7.2	Description	21
7.3	Remarks	21
7.4	Examples	21
7.5	Author	22

Part I

markdoc package commands

1 markdoc

markdoc is a general-purpose literate programming package for Stata that produces *dynamic analysis documents*, *presentation slides*, as well as Stata package help files and package vignettes in various formats such as **pdf**, **docx**, **html**, and **sthlp**. for further information see:

- markdoc homepage
- journal article
- manual on Github wiki
- release notes on Github
- examples on GitHub
- please ask your questions on statalist.org

1.1 Syntax

the syntax of the package is simple and can be summarized as:

markdoc *filename* [, *options*]

where *filename* can be:

Document format	Description
.do	executes <i>do</i> file, examines the analysis reproducibility, and produces a dynamic document
.smcl	converts <i>smcl</i> file to a dynamic document without testing the code reproducibility
.ado or .mata	generates <i>sthlp</i> help files or package vignettes from Markdown documentation

1.1.1 Options

the main options are the following:

Option	Description
mini	runs markdoc independent of any third-party software
statax	activates the built-in syntax highlighter
replace	replace the exported file if already exists
export(name)	document format; it can be md , html , docx , pdf , slide , tex , or sthlp
numbered	numbers Stata commands in the dynamic document
date	adds the current date in the document
title(str)	specify the title of the document
author(str)	specify the author of the document

Option	Description
affiliation (<i>str</i>)	specify the author affiliation in the document
address (<i>str</i>)	specify the author's contact information in the document
summary (<i>str</i>)	specify the summary of the document
style (<i>name</i>)	specify the style of the document; it can be simple , stata , or formal

and the following options are for communicating with the third-party software (not required in the **mini** mode)

Option	Description
install	installs the pandoc and wkhtmltopdf automatically, if they are not found
unc	specify that markdoc is being accessed from a Windows server with UNC file paths
pandoc (<i>str</i>)	specify the path to Pandoc software on the operating system
printer (<i>str</i>)	specify the path to pdf driver on the operating system
markup (<i>name</i>)	specify the markup language used for notation; the default is Markdown
master	automatically creates the layout for LaTeX and HTML documents
template (<i>str</i>)	renders the document using an external style sheet file or example docx file
test	examines markdoc third-party software with an example
toc	creates table of content

1.2 Additional commands

markdoc includes a few commands that greatly simplify writing the workflow. you may use the additional commands to:

1. write dynamic text, while applying any display directive supported by the display command:

```
txt [code] [display_directive] [display_directive [...]]
```

2. capture and include the current image in the document automatically (if *filename* is missing)

```
img [filename][, markup(str) title(str) width(int) height(int) markup(str) left center  
]
```

3. create and style a dynamic table in Markdown documents

```
tbl ([#, #... ] [ #[, #... ] [ [...]] ) [, title(str)
```

furthermore, you can also call **pandoc** or **wkhtmltopdf** within Stata i.e. :

```
pandoc command
```

```
wkhtmltopdf command
```

1.3 Installation

The latest release as well as archived older versions of **markdoc** are hosted on GitHub website. the recommended method for installing the package is through **github** package package, which is used for building, searching, installing, and managing Stata packages hosted on GitHub. you can install **github** by typing:

```
. net install github, from("https://haghish.github.io/github/")
```

next, install the latest stable **markdoc** release along with its Stata dependencies by typing:

```
. github install haghish/markdoc
```

1.4 Description

markdoc is a general-purpose literate programming package for Stata that produces *dynamic analysis documents*, *package vignette documentation*, *dynamic presentation slides*, as well as dynamic *Stata package help files*. to improve applications of the package for developing educational materials and encouraging university lecturers to ask students to practice literate programming for taking notes or doing their semester projects, **markdoc** was programmed to include unique features. These features make the package a complete tool for documenting data analysis, Stata packages, as well as a tool for producing educational materials within Stata Do-file editor. For example:

- it includes a syntax highlighter
- it recognizes markdown, html, and latex markup languages
- it can render LaTeX mathematical notations in **pdf**, **docx**, **html**, **odt**, and **tex** documents
- it can automatically capture graphs from Stata and include them in the document
- it creates dynamic tables and supports writing dynamic text for interpreting the analysis
- it includes a user-friendly GUI interface (try **db markdoc**) to make using **markdoc** easier for newbies.

1.5 Software Installation

REMEMBER, **markdoc** v. 4.2 forth can run without any software dependencies, using the **mini** engine.

Without applying the **mini** option, which uses the light-weight engine, the markdoc package requires additional software which can be installed manually or automatically. The required software are Pandoc and wkhtmltopdf. They are both opensource freeware, supported for any common operating system such as Microsoft Windows, Macintosh, and Linux. Naturally, users who wish to use LaTeX markup for writing the documentation, will need a pdfLaTeX.

After a manual installation, the path to executable Pandoc should be specified in **pandoc** option. similarly, the path to executable wkhtmltopdf or pdfLaTeX should be given to **printer** option. Note that the **printer** option is only needed for compiling PDF document.

With automatic installation (i.e. using the **install** option), Pandoc and Wkhtmltopdf are downloaded and placed in Weaver directory which is located in **/ado/plus/Weaver/** on your machine. To find the location of **ado/plus/** directory on your machine use the sysdir command which returns the system directories.

1.6 Set file paths permanently

After manual installation, the paths to the executable Pandoc, wkhtmltopdf, and pdfLaTeX can be permanently set using **weave setup** command. This command will open *weaversetup.ado* document, where you can define the files paths as global macros.

1.7 Software troubleshoot

As mentioned, the required software can be installed manually or automatically. The optional automatic installation is expected to work properly in Microsoft Windows **XP**, Windows **7**, and Windows **8.1**, Macintosh

OSX 10.9.5, Linux **Mint 17 Cinnamon** (32bit & 64bit), Ubuntu **14** (64bit), and **CentOS 7** (64bit). Other operating systems may require manual software installation.

However, if for some technical or permission reasons markdoc fails to download, access, or run Pandoc, install it manually and provide the file path to Pandoc using **PANDOC** option.

1.8 Calling Pandoc

PANDOC commands can also be executed from Stata. This command takes the path to the executable Pandoc from markdoc and allows you to use Pandoc seamlessly for converting files within Stata. FOR EXAMPLE:

```
. pandoc ./example.tex -o ./example.html
```

1.9 Writing mathematical notation

markdoc can render LaTeX mathematical notations not only when the document is exported to LaTeX **tex**, but also when the document is exported to **pdf** document or **slide**, Microsoft Office **docx**, OpenOffice and LibreOffice **odt**, and **html**.

mathematical notations can be inline a text paragraph or on a separate line. For writing inline notations, place the notation between single dollar signs, for example: $a^2 + b^2 = c^2$ For including notation on a separate line, place the notations between double dollar signs:

$$a^2 + b^2 = c^2$$

1.10 Inserting an image or figure in the document

Any of the supported markup languages can be used to insert a figure in the document. In general, there are two ways for inserting an image in the document. First, you can use Markdown, HTML, or LaTeX syntax for inserting an image - that is already saved in your hard drive - in the document. The other solution is using **img** command. **img** command can take the *filename* of exsisting image on the hard drive and print the markup code (Markdown, HTML, or LaTeX. the default is Markdown) int he document. **img** command can also auto-export the current graph and import it in the document. For more information in this regard see the **img**.

1.11 Writing dynamic text

the **txt** command is borrowed from weaver package to print dynamic text in the the exported dynamic document. It can be used for interpreting the analysis results or dynamically referring to values of scalars or macros in the dynamic document. Writing dynamic text allows the content of the text to change by altering analysis codes and thus is the desirable way for explaining the analysis results. The text and macros can be styled using any of the supported markup languages in markdoc which are *markdown*, *LaTeX*, and *HTML*. This command is fully documented on [GitHub Wiki](#).

1.12 Creating dynamic tables with tbl command

the **tbl** command also belongs to weaver package. The syntax of this command is similar to **matrix input**, however, it can include *string*, *digits*, *scalars*, and *macros* to create a dynamic table. This command is fully documented on [GitHub Wiki](#).

1.13 Markers

markdoc also introduces a few handy markers for annotating the document, regardless of the markup language you use to write the document (Markdown, Tex, HTML). These markers can be used to specify what parts of the code should or should not appear in the dynamic document. The table below provides a brief summary of these annotating markers. in general, comments - unless they appear after a command - will be ignored in the dynamic document. However, the markers mentioned below are that will influence the markdoc process.

1. Creating text block

```
. /**
creates a block of comments in the smcl file that will be interpreted
in the dynamic document. this sign is distinguished from the normal comment
signs, with one star.
. ***/
```

2. Hiding command or output

```
. /**/ only include the output in the dynamic document and hide the comment
. /**/ only include the command in the dynamic document and hide the output
```

3. Hiding a section. Anything placed after **//OFF** until **//ON** markers will be *ignored* in the dynamic document

```
. //OFF
Stata code
. //ON
```

4. Appending external text file (Markdown, HTML, LaTeX) to the dynamic document

```
. //IMPORT filename
```

Apart from the text block markers, the other markers *are not supported within loops*. Nonetheless, writing markup text within the loop is not recommended either because it only gets printed once. For active writing within the loop or a program, see the **txt** command.

1.14 Markers examples

1.14.1 Example of riting text in the do-file

As noted, **markdoc** package allows writing and styling text as a comment in the do-file, using special comment signs. here is an example:

```
. /***
Text heading
=====

subheading
-----

When you write a dynamic document in markdoc, place text between
the "/***" and "*/" signs. But they should be placed on separate lines,
as shown in this example.
. ***/
```

1.15 Dynamic Document Examples

qui log using example, replace

```
. /***
Introduction to markdoc (heading 1)
=====

Using Markdown (heading 2)
-----

Writing with __markdown__ syntax allows you to add text and graphs to
_smcl_ logfile and export it to a editable document format. I will demonstrate
the process by using the __Auto.dta__ dataset.

### Get started with markdoc (heading 3)
I will open the dataset, list a few observations, and export a graph.
Then I will export the logfile to Microsoft Office docx format.
. ***/

. /***/ sysuse auto, clear
. /**/ list in 1/5
. histogram price
. graph export graph.png, width(400) replace

. /***
Adding a graph or image in the report
=====

Adding a graph using Markdown
-----

In order to add a graph using Markdown, I export the graph in PNG format.
You can explain the graph in the "brackets" and define the file path in parentheses

![explain the graph](./graph.png)
. ***/
```

When the log file is created, we can translate it with **markdoc**

```
. markdoc example, replace export(html) install
. markdoc example, replace export(docx)
. markdoc example, replace export(tex) master
. markdoc example, replace export(pdf)
. markdoc example, replace export(epub)
```

1.16 Dynamic Slide Examples

this is an example of generating dynamic slides with **markdoc**

```
. qui log using example, replace

. /***
---
title:markdoc Dynamic Slides
author: E. F. Haghish
---

Slide 1
=====

- Writing with __markdown__ syntax allows you to add text and graphs
to _smcl_ logfile and export it to a editable document format. I will demonstrate
the process by using the __Auto.dta__ dataset.

- I will open the dataset, list a few observations, and export a graph.
Then I will export the logfile to Microsoft Office docx format.

Adding commands and output
=====
. ***/

. sysuse auto, clear
. histogram price
. graph export graph.png, width(400) replace

. /***
Adding image in a slide
=====

![Histogram of the price variable](./graph.png)
. ***/

. qui log c
. markdoc example, replace export(slide) install printer("/usr/texbin/pdflatex")
```

1.17 Supported markup languages

markdoc supports three markup languages which are Markdown, HTML, and LaTeX. Markup languages should not be used together in one document because **markdoc** process each markup language differently.

1.18 Remarks

If the log-file is closed exactly using `qui` command, **markdoc** automatically removes this command from the end of the file. Similarly, markdoc removes `from` from the logfile. Therefore `and` can be used to separate codes in the dofile that are not wanted in the dynamic document, but still required in for the analysis. Nonetheless, this is not a proper practice and can harm the transparency of the analytic session. The log-file should include as much information about the history of the analysis as possible. Use the `for` for hiding sections of the log-file in the dynamic document.

1.19 Author

E. F. Haghish
University of Göttingen
haghish@med.uni-goettingen.de
<https://github.com/haghish>

1.20 See also

- Statax, JavaScript & LaTeX syntax highlighter for Stata

1.21 License

MIT License

This help file was dynamically produced by MarkDoc Literate Programming package
version 4.8

2 mini

mini is a simplified command to call **markdoc** in light-weight *mini* mode, without requiring any dependencies

2.1 Syntax

mini *filename* [, *options*]

where *options* are identical to the markdoc options

2.2 Description

mini is just a wrapper for **markdoc** with the *mini* option to run **markdoc** independent of any third-party dependencies.

2.3 Examples

execute a do-file and produce a word, pdf, and html files

```
. mini "filename.do" , export(docx)
. mini "filename.do" , export(pdf)
. mini "filename.do" , export(html)
```

2.4 Author

E. F. Haghish
University of Göttingen
haghish@med.uni-goettingen.de
<https://github.com/haghish>

2.5 License

MIT License

This help file was dynamically produced by MarkDoc Literate Programming package
v. 1.1

3 pandoc

pandoc executing Pandoc from Stata

3.1 Syntax

pandoc *anything*

3.2 Description

Pandoc is a document convertor freeware. The **markdoc** package uses this application to produce dynamic documents, slides, and package documentation. This program is a supplementary command that allows using this application for other purposes, outside **markdoc**.

3.3 Examples

executing Pandoc command

```
. pandoc _filename_ -o _filename_
```

adding more Pandoc arguments

```
. pandoc -s -S _filename_ -o _filename_
```

3.4 Author

E. F. Haghish
Center for Medical Biometry and Medical Informatics
University of Freiburg, Germany
haghish@imbi.uni-freiburg.de
<http://www.haghish.com/stat>

3.5 License

MIT License

This help file was dynamically produced by MarkDoc Literate Programming package
v. 1.1

4 wkhtmltopdf

wkhtmltopdf renders **html** documents to **pdf** within Stata

4.1 Syntax

wkhtmltopdf [*options*] *filename.html filename.pdf*

See the *options*, which is a link to the **wkhtmltopdf** manual, explaining the arguments you can add to adjust the pdf output.

4.2 Description

if the **mini** engine is not used, **markdoc** requires the wkhtmltopdf software to convert **html** to **pdf** without requiring installing LaTeX. Moreove, **markdoc** provides automatic installation of wkhtmltopdf, if desired.

However, **markdoc** is not the only software that deals with documents in Stata and many users show interest to create dynamic documents in their own way. to help them create **pdf** documents, this command was created to convert their **html** documents to **pdf**.

4.3 Example

convert html file to pdf

```
. wkhtmltopdf myfile.html myfile.pdf
```

4.4 Author

E. F. Haghish
University of Göttingen
haghish@med.uni-goettingen.de
<https://github.com/haghish>

4.5 License

MIT License

This help file was dynamically produced by MarkDoc Literate Programming package

Part II

Additional commands

5 txt

txt prints string and values of scalar expressions or macros in dynamic document. By default, the command writes a text paragraph. The primary purpose of the command is writing dynamic text to interpret analysis results in the dynamic document. This command belongs to weaver package, but it also supports the markdoc package.

The syntax for both packages is similar, but the **txt** command behaves differently based on which package is in use. If Weaver log is on, **txt** functions for Weaver package only. This document only describes the **txt** command in Weaver package. For using the command in markdoc package see the MarkDoc documentation on GitHub wiki

5.1 Syntax

txt [**code**] [*display_directive* [*display_directive* [...]]]

The **code** subcommand is optional and if specified, it prints the output in the dynamic document.

where the *display_directive* is:

- Weaver Markup
-
-
-
- [%fmt] [=]exp
- _skip(#)
- _column(#)
- _newline[(#)]
- _dup(#)
- ,
- ”

5.2 Description

txt prints dynamic text i.e. strings and values of scalar expressions or macros in the Weaver or the smcl log-file. **txt** also prints output from the user-written Stata programs. Any of the supported markup languages can be used to alter the string and scalar expressions. This command is to some extent similar to **display** command in Stata. For example, it can be used to carry out a mathematical calculation by typing:

```
. txt 1+1
```

It also supports many of the display directives as well.

Note that in contrast to the **display** command that prints the **scalar** unformatted, the **txt** command uses the default **%10.2f** format for displaying the scalar. This feature

helps the users avoid specifying the format for every scalar, due to popularity of this format. However, specifying the format expression can overrule the default format. For example:

```
. scalar num = 10.123
. txt "The value of the scalar is " %5.1f num
```

The example above will print the scalar with only 1 decimal number. This feature only supports scalar interpretation and does not affect the **macro** contents.

5.3 Display directives

The supported *display_directives* are used in do-files and programs to produce formatted output. The directives are

Directives	Description
Weaver Markup	A simplified markup language for annotating the content of the HTML log displays the string without the quotes allows embedded quotes
<code>[%fmt] [=] exp</code>	allows results to be formatted
<code>skip(#) skips # columns</code>	skips to the #th column
<code> column(#)</code>	
<code>newline goes to a new line</code>	skips # lines
<code>line newline(#)</code>	
<code>dup(#) repeats the next directive #- times</code>	
<code>,</code>	displays one blank between two directives
<code>„(#)</code>	places no blanks between two directives

5.4 Mathematical notations

The **txt** command can be used for writing mathematical notations in Weaver package, both in HTML and LaTeX log files. Writing mathematical notations in the HTML log is made possible by including MathJax engine, a JavaScript-based engine for rendering LaTeX notations in HTML format. To do so, notations should begin with and end with `for` for rendering notations within the text and double dollar sign or alternatively, the `and` for rendering notations in a separate line. For more information in this regard, see mathematical notations documentation.

When Weaver package is running, the **code** subcommand appends the dynamic text to the

5.5 Examples

As a hand calculator:

```
. txt 2 * 2
```

As might be used in do-files and programs:

```
. sysuse auto
. summarize price
. txt "mean of Price variable is " r(mean) " and SD is " %9.3f r(sd)
```

If the text only includes string and macro, the double quotations can be ignored. The **txt** command will interpret all of the *display_directives* and scalars as string (so it's not recommended):

```
. local n 9.9
. txt Not recommended, but you may also print the value of without double quote
```

5.6 Author

E. F. Haghish
University of Göttingen
haghish@med.uni-goettingen.de
<https://github.com/haghish>

This help file was dynamically produced by MarkDoc Literate Programming package
v. 2.0

6 img

img captures and imports images and graphs into the dynamic document. This command belongs to **Weaver** package but it also supports the **markdoc** package. This document only describes **txt** in Weaver package. For using the command in markdoc package, read the markdoc manual.

6.1 Syntax

Import graphical files in the dynamic document

```
img [using filename] [, options ]
```

Automatically include the current graph from Stata in the dynamic document

```
img [, options ]
```

6.1.1 Options

Option	Description
title (str)	specify a header string (title) for the figure
width (int)	define the width of the figure. This option must be used with <i>height</i> option
height (int)	define the hight of the figure. This option must be used with <i>width</i> option
left	aligns the figure to the left-side of the dynamic document (default)
center	aligns the figure to the center of the document

6.2 Description

The **img** command imports images and graphs into the dynamic document. Any graphical file that is compatible with a web-browser can be inserted in the html log. This command belongs to Weaver package but it also supports the markdoc package. The syntax for both packages is the same but the **img** command behave differently based on which of the packages is in use. If Weaver html log and smcl log are open at the same time, the command only functions for Weaver and not for markdoc. In contrast, when Weaver html log is not open and smcl log is on, it will function for markdoc package.

6.3 Examples

You have created a graph in Stata. Before importing in the HTML log, you should export it in a format that can be interpreted in html. Such as PNG which is recommended because it is lossless format and the same file can be used for publication.

```
. sysuse auto
. histogram price
. graph export price.png, replace

. img using price.png
. img using price.png, title("Histogram of the Price variable")
. img using price.png, w(300) h(200) center
```

Alternatively, the image can be obtained from Stata automatically

```
. histogram mpg
. img, title("Histogram of the MPG variable")
```

6.4 Author

E. F. Haghish
University of Göttingen
haghish@med.uni-goettingen.de
<https://github.com/haghish>

This help file was dynamically produced by MarkDoc Literate Programming package
v. 2.0

7 tbl

tbl creates a dynamic table in **HTML**, **LaTeX**, or **Markdown**. It can also align each column to left, center, or right, and also create multiple-columns for hierarchical tables. This command belongs to **Weaver** package, but it also supports the **markdoc** package. For using the command in **markdoc** package see the MarkDoc documentation on GitHub wiki

7.1 Syntax

`\tbl _(*[,*...] [*[,*...] [[...]]])_ [, options]`

where the `--*` represents a *display directive* which is

Directives	Description
	displays the string without the quotes
	allows embedded quotes
<code>[%fmt] [=] exp</code>	allows results to be formatted
<code>,</code>	separates the directives of each column of the table
<code>{l}</code>	if placed before any directives, create a left-aligned column
<code>{c}</code>	creates a center-aligned column
<code>{r}</code>	creates a right-aligned column
<code>{col #}</code>	this directive will create a multi-column by merging # number of columns

7.1.1 Options

Option	Description
title (str)	specify a header string (title) for the table
width (int)	define the width of the table. This option must be used with <i>height</i> option
height (int)	define the height of the table. This option must be used with <i>width</i> option
left	aligns the table to the left-side of the dynamic document (default)
center	aligns the table to the center of the document

7.2 Description

tbl is a command in **Weaver** package that creates a dynamic table in HTML or LaTeX, depending on the markup language used in Weaver log. If Weaver HTML is in use, **tbl** will be able to interpret the Weaver Markup codes as well as Weaver mathematical notations. In other words, Weaver Markups and Weaver mathematical notations can be used as a display directives within the **tbl** command to alter other directives or display mathematical signs and formulas. Advanced users can also use HTML code to alter the table.

If LaTeX markup is used for creating the Weaver log, then **tbl** command creates a LaTeX table. However, neither Weaver Markup nor Weaver mathematical notations are not supporting LaTeX. Instead, LaTeX mathematical notations can be used for writing mathematical notations or altering the table.

7.3 Remarks

Note that the **tbl** command parses the rows using the backslash symbol. Therefore, to include LATEX notations in a dynamic table that begin with a backslash such as **beta** sign, double backslash should be used to avoid conflict with the parsing syntax (e.g. `\beta` or `95\%`)

7.4 Examples

creating a simple 2x3 table with string and numbers

```
. tbl ("Column 1", "Column 2", "Column 3" \ 10, 100, 1000 )
```

creating a table that includes scalars and aligns the columns to left, center, and right respectively

```
. tbl ({l}"Left", {c}"Centered", {r}"Right" \ c(os), c(machine_type), c(username))
```

write mathematical notations

```
. tbl ("\" \\beta \)", "\" \\epsilon \"" \ "\" \\sum \)", "\" \\prod \)")
```

7.5 Author

E. F. Haghish
University of Göttingen
haghish@med.uni-goettingen.de
<https://github.com/haghish>

This help file was dynamically produced by MarkDoc Literate Programming package