

**Dynamic documents intro** — Introduction to dynamic documents[Description](#)[Remarks and examples](#)[Also see](#)

## Description

Stata's dynamic document commands create text files, Word (.docx) documents, and HTML files that include Stata results. With these commands, you can create documents that combine text with summary statistics, regression results, graphs, and other Stata results. You can include the full output of Stata commands or incorporate individual values from the results of commands. Word documents and HTML files can easily be customized using the Markdown text-formatting language.

See the following manual entries for details on dynamic documents:

[RPT] [Dynamic tags](#) Dynamic tags for text files

[RPT] [dyndoc](#) Convert dynamic Markdown document to HTML or Word (.docx) document

[RPT] [dyntext](#) Process Stata dynamic tags in text file

[RPT] [markdown](#) Convert Markdown document to HTML file or Word (.docx) document

These documents are dynamic because, as your data change, you simply rerun the `dyndoc` or `dyntext` command that creates your dynamic document, and the HTML file, Word document, or text file is updated with the new results.

## Remarks and examples

stata.com

Creating files with Stata output is straightforward with `dyntext` and `dyndoc`. The former processes Stata commands within a plain text file to create a text file that includes Stata output. The latter converts a text file with Markdown text and Stata commands into a formatted HTML file or Word document file with Stata output. The Stata commands are processed according to the [dynamic tags](#) that indicate how commands, output, results of expressions, and graphs should be inserted in a document. Markdown, a simple markup language with a formatting syntax based on plain text, is processed by `dyndoc`. This allows you to include headings, subheadings, bold and italic font, text boxes, bulleted lists, and more in the HTML files and Word documents generated by this command.

Below, we briefly overview and demonstrate the dynamic document creation commands. See the individual entries for the syntax and additional examples.

### ► Example 1: Using dynamic tags

Whether you want to create a text file, Word document, or an HTML file, you will use dynamic tags in your source file to embed Stata output in your destination file. Different tags are available to include Stata output, expressions, and graphs in the destination file. With dynamic tags, you can control whether the Stata command, the output, or both are included in the final document. For example, in the following text file we load `auto.dta` and then `summarize mpg`:

begin example.txt

```
Using Stata dynamic tags
=====
```

```
I am going to examine fuel efficiency using -auto.dta-. First, I load the dataset:
```

```
<<dd_do>>
sysuse auto, clear
<</dd_do>>
```

```
Now I -summarize mpg-, but I only display the output, not the command:
```

```
<<dd_do: nocommands>>
summarize mpg
<</dd_do>>
```

end example.txt

You can type

```
. copy http://www.stata-press.com/data/r16/reporting/example.txt .
```

to copy `example.txt` to your current working directory.

We use the `<<dd_do>>` dynamic tag to run the `sysuse` command. However, when summarizing `mpg`, we specify the `nocommands` attribute to suppress the command in the output file. Attributes, which can be thought of as options, modify the tag's behavior. See [RPT] [Dynamic tags](#) for a full list of dynamic tags.

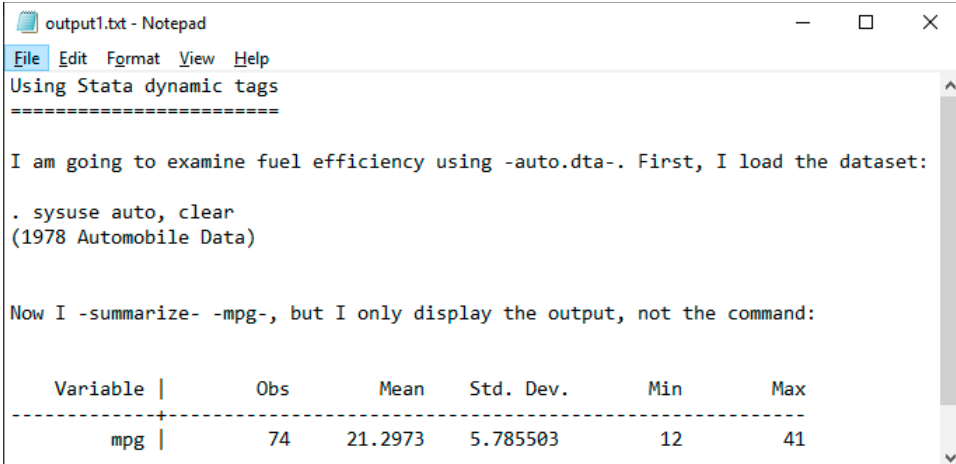


### ► Example 2: Create a text file with Stata output

Having enclosed the Stata commands within the dynamic tags in our text file, we can use `dyntax` to embed the output from those commands in a text file:

```
. dyntax example.txt, saving(output1.txt)
```

This command produces the following:



```
output1.txt - Notepad
File Edit Format View Help
Using Stata dynamic tags
=====

I am going to examine fuel efficiency using -auto.dta-. First, I load the dataset:

. sysuse auto, clear
(1978 Automobile Data)

Now I -summarize- -mpg-, but I only display the output, not the command:

-----
Variable |      Obs      Mean   Std. Dev.   Min     Max
-----+-----
      mpg |       74   21.2973   5.785503    12     41
-----
```

This `output1.txt` file is available at <http://www.stata-press.com/data/r16/reporting/>.

For a more detailed example of including Stata output in text files, see [RPT] [dyntext](#).

◀

### ▷ Example 3: Create an HTML file with Stata output

Suppose that now we want to convert `example.txt` to an HTML file. We could use `dyndoc` instead of `dyntext`, but the output would not be formatted nicely; you can try it for yourself to see the difference. In fact, the beauty of `dyndoc` is that it can process Markdown-formatted text and embed Stata output in the destination file. Let's see `dyndoc` in its full potential by adding Markdown text to our previous file. We still need to use dynamic tags to process the Stata code, but rather than using “-” to indicate Stata command names, variable names, etc., we will use two asterisks around each command and variable name. We also enclose Stata commands and tags in sets of four tildes to display the content in plain text. Our modified text file is shown here:

begin `example2.txt`

```
Using Stata dynamic tags
=====

I am going to examine fuel efficiency using auto.dta. First, I load the
dataset:
~~~~
<<dd_do>>
sysuse auto, clear
<</dd_do>>
~~~~

Now I summarize mpg, but I only display the output, not the command:
~~~~
<<dd_do: nocommands>>
summarize mpg
<</dd_do>>
~~~~
```

end `example2.txt`

We copy this file to our current working directory and then convert it to an HTML file with `dyndoc`:

```
. copy http://www.stata-press.com/data/r16/reporting/example2.txt .
. dyndoc example2.txt
```

This produces the following:



We could have also included a graph or Stata expression in the HTML file; see [RPT] [dyndoc](#) for a more detailed example.

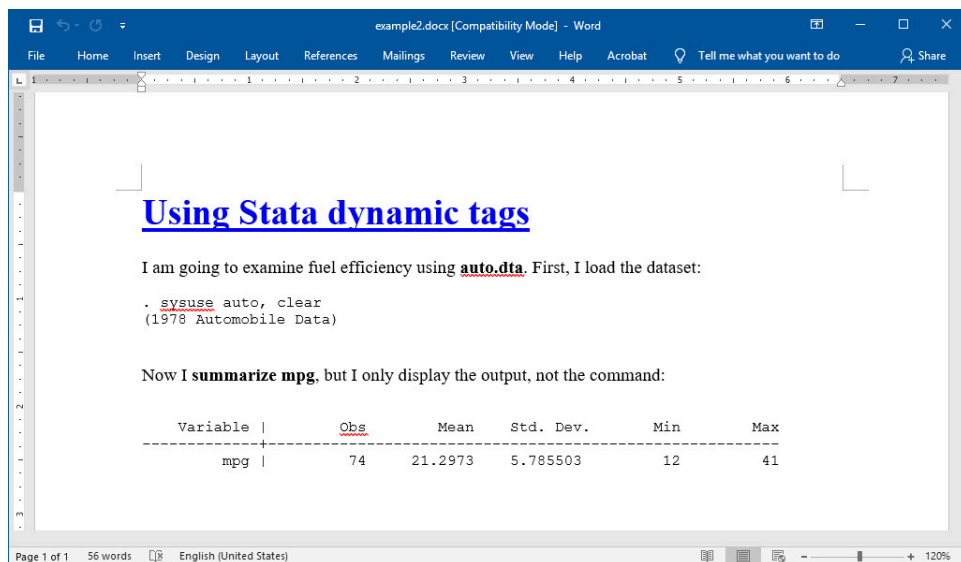
◀

### ► Example 4: Create a Word document with Stata output

By default, `dyndoc` will create an HTML output file, but we could just as easily convert `example.txt` to a Word (`.docx`) document instead by specifying the `docx` option:

```
. dyndoc example2.txt, docx
```

This produces `example2.docx`, which looks like this:



Once you have created a Word document, you can also convert it to a PDF file; see [\[RPT\] docx2pdf](#).

The output files, `example2.html` and `example2.docx`, are available at <http://www.stata-press.com/data/r16/reporting/>.

Above, we used `dyndoc` to embed the output from commands and to process Markdown-formatted text when we created an HTML file and a Word document. `dyndoc` uses the `markdown` command to convert the Markdown-formatted text to the HTML format. `markdown` can also be used independently to create HTML files and Word documents when the source file does not contain Stata dynamic tags.

## Also see

[\[RPT\] Dynamic tags](#) — Dynamic tags for text files

[\[RPT\] dyndoc](#) — Convert dynamic Markdown document to HTML or Word (.docx) document

[\[RPT\] dyntext](#) — Process Stata dynamic tags in text file

[\[RPT\] markdown](#) — Convert Markdown document to HTML file or Word (.docx) document