`latexindent.pl`

`Version 3.0`

Chris Hughes [*]

January 22, 2017

Abstract

`latexindent.pl` is a `Perl` script that indents `.tex` (and other) files according to an indentation scheme that the user can modify to suit their taste. Environments, including those with alignment delimiters (such as `tabular`), and commands, including those that can split braces and brackets across lines, are *usually* handled correctly by the script. Options for `verbatim`-like environments and indentation after headings (such as `chapter`, `section`, etc) are also available. The script also has the ability to modifiy line breaks, and add comment symbols.

Contents

Listings

[*] and contributors! (See **??** on page ??.) For all communication, please visit [**latexindent-home**].

1 The -m (modifylinebreaks) switch

All features described in this section will only be relevant if the -m switch is used.

modifylinebreaks: ⟨*fields*⟩

One of the most exciting features of Version 3.0 is the -m switch, which permits latexindent.pl to modify line breaks, according to the specifications in the modifyLineBreaks field. *The settings in this field will only be considered if the -m switch has been used.* A snippet of the default settings of this field is shown in Listing 1.

LISTING 1: modifyLineBreaks

```
modifyLineBreaks:
    preserveBlankLines: 1
    condenseMultipleBlankLinesInto: 1
   ...
```

Having read the previous paragraph, it should sound reasonable that, if you call latexindent.pl using the -m switch, then you give it permission to modify line breaks in your file, but let's be clear:

> If you call latexindent.pl with the -m switch, then you are giving it permission to modify line breaks. By default, the only thing that will happen is that multiple blank lines will be condensed into one blank line; many other settings are possible, discussed next.

All YAML-based details in this section only apply if the -m switch is active.

preserveBlankLines: **0|1**

This field is directly related to *poly-switches*, discussed below. By default, it is set to 1, which means that blank lines will be protected from removal; however, regardless of this setting, multiple blank lines can be condensed if condenseMultipleBlankLinesInto is greater than 0, discussed next.

condenseMultipleBlankLinesInto: ⟨*integer* ≥ 0⟩

Assuming that this switch takes an integer value greater than 0, latexindent.pl will condense multiple blank lines into the number of blank lines illustrated by this switch. As an example, Listing 2 shows a sample file with blank lines; upon running

```
cmh:~$ latexindent.pl myfile.tex -m
```

the output is shown in Listing 3; note that the multiple blank lines have been condensed into one blank line, and note also that we have used the -m switch!

LISTING 2: mlb1.tex

```
before blank line



after blank line


after blank line
```

LISTING 3: mlb1.tex out output

```
before blank line

after blank line

after blank line
```

1.1 Poly-switches

Every other field in the modifyLineBreaks field uses poly-switch, and can take one of four integer values[1]:

−1 *remove mode*: line breaks before or after the *<part of thing>* can be removed (assuming that preserveBlankLines is set to 0);

0 *off mode*: line breaks will not be modified for the *<part of thing>* under consideration;

1 *add mode*: a line break will be added before or after the *<part of thing>* under consideration, assuming that there is not already a line break before or after the *<part of thing>*;

2 *comment then add mode*: a comment symbol will be added, followed by a line break before or after the *<part of thing>* under consideration, assuming that there is not already a comment and line break before or after the *<part of thing>*.

All poly-switches are *off* by default; latexindent.pl searches first of all for per-name settings, and then followed by global per-thing settings.

1.2 modifyLineBreaks for environments

We start by viewing a snippet of defaultSettings.yaml in Listing 4; note that it contains *global* settings (immediately after the environments field) and that *per-name* settings are also allowed – in the case of Listing 4, settings for equation* have been specified. Note that all poly-switches are *off* by default.

LISTING 4: environments

```
    environments:
        BeginStartsOnOwnLine: 0
        BodyStartsOnOwnLine: 0
        EndStartsOnOwnLine: 0
        EndFinishesWithLineBreak: 0
        equation*:
            BeginStartsOnOwnLine: 0
            BodyStartsOnOwnLine: 0
            EndStartsOnOwnLine: 0
            EndFinishesWithLineBreak: 0
```

1.2.1 Adding line breaks (poly-switches set to 1 or 2)

Let's begin with the simple example given in Listing 5; note that we have annotated key parts of the file using ♠, ♡, ◇ and ♣, these will be related to fields specified in Listing 4.

[1] visual learners might like to associate one of the four circles in the logo with one of the four given values

LISTING 5: env-mlb1.tex

```
before words♠ \begin{myenv}♡body of myenv◇\end{myenv}♣ after words
```

Let's explore BeginStartsOnOwnLine and BodyStartsOnOwnLine in Listings 6 and 7, and in particular, let's allow each of them in turn to take a value of 1.

LISTING 6: env-mlb1.yaml

```
modifyLineBreaks:
    environments:
        BeginStartsOnOwnLine: 1
```

LISTING 7: env-mlb2.yaml

```
modifyLineBreaks:
    environments:
        BodyStartsOnOwnLine: 1
```

After running the following commands,

```
cmh:~$ latexindent.pl -m env-mlb.tex -l env-mlb1.yaml
cmh:~$ latexindent.pl -m env-mlb.tex -l env-mlb2.yaml
```

the output is as in Listings 8 and 9.

LISTING 8: env-mlb.tex using Listing 6

```
before words
\begin{myenv}body of myenv\end{myenv} after words
```

LISTING 9: env-mlb.tex using Listing 7

```
before words \begin{myenv}
  body of myenv\end{myenv} after words
```

There are a couple of points to note:

- in Listing 8 a line break has been added at the point denoted by ♠ in Listing 5; no other line breaks have been changed;
- in Listing 9 a line break has been added at the point denoted by ♡ in Listing 5; furthermore, note that the *body* of myenv has received the appropriate (default) indentation.

Let's now change each of the 1 values in Listings 6 and 7 so that they are 2 and save them into env-mlb3.yaml and env-mlb4.yaml respectively (see Listings 10 and 11).

LISTING 10: env-mlb3.yaml

```
modifyLineBreaks:
    environments:
        BeginStartsOnOwnLine: 2
```

LISTING 11: env-mlb4.yaml

```
modifyLineBreaks:
    environments:
        BodyStartsOnOwnLine: 2
```

Upon running commands analogous to the above, we obtain Listings 12 and 13.

LISTING 12: env-mlb.tex using Listing 10

```
before words%
\begin{myenv}body of myenv\end{myenv} after words
```

LISTING 13: env-mlb.tex using Listing 11

```
before words \begin{myenv}%
  body of myenv\end{myenv} after words
```

Note that line breaks have been added as in Listings 8 and 9, but this time a comment symbol has been added before adding the line break; in both cases, trailing horizontal space has been stripped before doing so.

Let's explore EndStartsOnOwnLine and EndFinishesWithLineBreak in Listings 14 and 15, and in particular, let's allow each of them in turn to take a value of 1.

LISTING 14: env-mlb5.yaml

```
modifyLineBreaks:
    environments:
        EndStartsOnOwnLine: 1
```

LISTING 15: env-mlb6.yaml

```
modifyLineBreaks:
    environments:
        EndFinishesWithLineBreak: 1
```

After running the following commands,

```
cmh:~$ latexindent.pl -m env-mlb.tex -l env-mlb5.yaml
cmh:~$ latexindent.pl -m env-mlb.tex -l env-mlb6.yaml
```

the output is as in Listings 16 and 17.

LISTING 16: env-mlb.tex using Listing 14

```
before words \begin{myenv}body of myenv
\end{myenv} after words
```

LISTING 17: env-mlb.tex using Listing 15

```
before words \begin{myenv}body of myenv\end{myenv}
after words
```

There are a couple of points to note:

- in Listing 16 a line break has been added at the point denoted by ♢ in Listing 5 on page 4; no other line breaks have been changed and the `\end{myenv}` statement has *not* received indentation (as intended);
- in Listing 17 a line break has been added at the point denoted by ♣ in Listing 5 on page 4.

Let's now change each of the `1` values in Listings 14 and 15 so that they are `2` and save them into `env-mlb7.yaml` and `env-mlb8.yaml` respectively (see Listings 18 and 19).

LISTING 18: env-mlb7.yaml

```
modifyLineBreaks:
    environments:
        EndStartsOnOwnLine: 2
```

LISTING 19: env-mlb8.yaml

```
modifyLineBreaks:
    environments:
        EndFinishesWithLineBreak: 2
```

Upon running commands analogous to the above, we obtain Listings 20 and 21.

LISTING 20: env-mlb.tex using Listing 18

```
before words \begin{myenv}body of myenv%
\end{myenv} after words
```

LISTING 21: env-mlb.tex using Listing 19

```
before words \begin{myenv}body of myenv\end{myenv}%
after words
```

Note that line breaks have been added as in Listings 16 and 17, but this time a comment symbol has been added before adding the line break; in both cases, trailing horizontal space has been stripped before doing so.

If you ask `latexindent.pl` to add a line break (possibly with a comment) using a poly-switch value of 1 (or 2), it will only do so if necessary. For example, if you process the file in Listing 7 on page 4 using any of the YAML files presented so far in this section, it will be left unchanged.

LISTING 22: env-mlb2.tex

```
before words
\begin{myenv}
  body of myenv
\end{myenv}
after words
```

LISTING 23: env-mlb3.tex

```
before words
\begin{myenv}  %
  body of myenv%
\end{myenv}%
after words
```

In contrast, the output from processing the file in Listing 23 will vary depending on the poly-switches used; in Listing 24 you'll see that the comment symbol after the `\begin{myenv}` has been moved to the next line, as `BodyStartsOnOwnLine` is set to `1`. In Listing 25 you'll see that the comment has been accounted for correctly, and that, because `BodyStartsOnOwnLine` has been set to `2`, the comment symbol has *not* been moved to its own line. You're encouraged to experiment with Listing 23 and by setting the other poly-switches considered so far to `2` in turn.

LISTING 24: env-mlb3.tex using Listing 7 on page 4

```
before words
\begin{myenv}
  %
  body of myenv%
\end{myenv}%
after words
```

LISTING 25: env-mlb3.tex using Listing 11 on page 4

```
before words
\begin{myenv}  %
  body of myenv%
\end{myenv}%
after words
```

The details of the discussion in this section have concerned *global* poly-switches in the `environments` field; each switch can also be specified on a *per-name* basis, which would take priority over the global values; with reference to Listing 4 on page 3, an example is shown for the `equation*` environment.

1.2.2 Removing line breaks (poly-switches set to −1)

Setting poly-switches to −1 tells `latexindent.pl` to remove line breaks of the <*part of the thing*>, if necessary. We will consider the example code given in Listing 26, noting in particular the positions of the line break highlighters, ♠, ♡, ◇ and ♣, together with the associated YAML files in Listings 27 to 30.

LISTING 26: `env-mlb4.tex`

```
before words♠
\begin{myenv}♡
body of myenv◇
\end{myenv}♣
after words
```

LISTING 27: `env-mlb9.yaml`

```
modifyLineBreaks:
    environments:
        BeginStartsOnOwnLine: -1
```

LISTING 28: `env-mlb10.yaml`

```
modifyLineBreaks:
    environments:
        BodyStartsOnOwnLine: -1
```

LISTING 29: `env-mlb11.yaml`

```
modifyLineBreaks:
    environments:
        EndStartsOnOwnLine: -1
```

LISTING 30: `env-mlb12.yaml`

```
modifyLineBreaks:
    environments:
        EndFinishesWithLineBreak: -1
```

After running the commands

```
cmh:~$ latexindent.pl -m env-mlb4.tex -l env-mlb9.yaml
cmh:~$ latexindent.pl -m env-mlb4.tex -l env-mlb10.yaml
cmh:~$ latexindent.pl -m env-mlb4.tex -l env-mlb11.yaml
cmh:~$ latexindent.pl -m env-mlb4.tex -l env-mlb12.yaml
```

we obtain the output in Listings 31 to 34.

LISTING 31: `env-mlb4.tex` using Listing 27

```
before words\begin{myenv}
  body of myenv
\end{myenv}
after words
```

LISTING 32: `env-mlb4.tex` using Listing 28

```
before words
\begin{myenv}body of myenv
\end{myenv}
after words
```

LISTING 33: `env-mlb4.tex` using Listing 29

```
before words
\begin{myenv}
  body of myenv\end{myenv}
after words
```

LISTING 34: `env-mlb4.tex` using Listing 30

```
before words
\begin{myenv}
  body of myenv
\end{myenv}after words
```

Notice that in

- Listing 31 the line break denoted by ♠ has been removed;
- Listing 32 the line break denoted by ♡ has been removed;
- Listing 33 the line break denoted by ◇ has been removed;

- Listing 34 the line break denoted by ♣ has been removed.

We examined each of these cases separately for clarity of explanation, but you can combine all of the YAML settings in Listings 27 to 30 into one file; alternatively, you could tell `latexindent.pl` to load them all by using the following command, for example

```
cmh:~$ latexindent.pl -m env-mlb4.tex -l env-mlb9.yaml,env-mlb10.yaml,env-mlb11.yaml,env-mlb12.yaml
```

which gives the output in Listing 5 on page 4.

About trailing horizontal space Recall that on **??** we discussed the YAML field `removeTrailingWhitespace`, and that it has two (binary) switches to determine if horizontal space should be removed `beforeProcessing` and `afterProcessing`. The `beforeProcessing` is particularly relevant when considering the `-m` switch; let's consider the file shown in Listing 35, which highlights trailing spaces.

Listing 35: `env-mlb5.tex`

```
before␣words␣␣␣♠
\begin{myenv}␣␣␣␣␣␣␣␣␣␣␣♡
body␣of␣myenv␣␣␣␣␣␣♢
\end{myenv}␣␣␣␣␣♣
after␣words
```

Listing 36: `removeTWS-before.yaml`

```
removeTrailingWhitespace:
    beforeProcessing: 1
```

The output from the following commands

```
cmh:~$ latexindent.pl -m env-mlb5.tex -l env-mlb9.yaml,env-mlb10.yaml,env-mlb11.yaml,env-mlb12.yaml
cmh:~$ latexindent.pl -m env-mlb5.tex -l
    env-mlb9.yaml,env-mlb10.yaml,env-mlb11.yaml,env-mlb12.yaml,removeTWS-before.yaml
```

is shown, respectively, in Listings 37 and 38; note that the trailing horizontal white space has been preserved (by default) in Listing 37, while in Listing 38, it has been removed using the switch specified in Listing 36.

Listing 37: `env-mlb5.tex` using Listings 31 to 34

```
before words   \begin{myenv}            body of myenv      \end{myenv}     after words
```

Listing 38: `env-mlb5.tex` using Listings 31 to 34 *and* Listing 36

```
before words\begin{myenv}body of myenv\end{myenv}after words
```

Blank lines Now let's consider the file in Listing 39, which contains blank lines.

Listing 39: `env-mlb6.tex`

```
before words♠


\begin{myenv}♡


body of myenv♢


\end{myenv}♣

after words
```

Listing 40: `UnpreserveBlankLines.yaml`

```
modifyLineBreaks:
    preserveBlankLines: 0
```

Upon running the following commands

```
cmh:~$ latexindent.pl -m env-mlb6.tex -l env-mlb9.yaml,env-mlb10.yaml,env-mlb11.yaml,env-mlb12.yaml
cmh:~$ latexindent.pl -m env-mlb6.tex -l
    env-mlb9.yaml,env-mlb10.yaml,env-mlb11.yaml,env-mlb12.yaml,UnpreserveBlankLines.yaml
```

we receive the respective outputs in Listings 41 and 42. In Listing 41 we see that the multiple blank lines have each been condensed into one blank line, but that blank lines have *not* been removed by the poly-switches – this is because, by default, `preserveBlankLines` is set to `1`. By contrast, in Listing 42, we have allowed the poly-switches to remove blank lines because, in Listing 40, we have set `preserveBlankLines` to `0`.

LISTING 41: `env-mlb6.tex` using Listings 31 to 34

```
before words

\begin{myenv}

  body of myenv

\end{myenv}

after words
```

LISTING 42: `env-mlb6.tex` using Listings 31 to 34 *and* Listing 40

```
before words\begin{myenv}body of myenv\end{myenv}after words
```

1.3 Poly-switches for other code blocks

Rather than repeat the examples shown for the environment code blocks (in Section 1.2 on page 3), we choose to detail the poly-switches for all other code blocks in Table 1; note that each and every one of these poly-switches is *off by default*.

TABLE 1: Poly-switch mappings for all code-block types

Code block	Sample	Poly-switch mapping
environment	`before words`♠ `\begin{myenv}`♡ `body of myenv`♢ `\end{myenv}`♣ `after words`	♠ BeginStartsOnOwnLine ♡ BodyStartsOnOwnLine ♢ EndStartsOnOwnLine ♣ EndFinishesWithLineBreak
ifelsefi	`before words`♠ `\if...`♡ `body of if statement`★ `\else`□ `body of else statement`♢ `\fi`♣ `after words`	♠ IfStartsOnOwnLine ♡ BodyStartsOnOwnLine ★ ElseStartsOnOwnLine □ ElseFinishesWithLineBreak ♢ FiStartsOnOwnLine ♣ FiFinishesWithLineBreak

optionalArguments	...♠ [♡ body of opt arg♢]♣ ...	♠ LSqBStartsOnOwnLine[2] ♡ OptArgBodyStartsOnOwnLine ♢ RSqBStartsOnOwnLine ♣ RSqBFinishesWithLineBreak
mandatoryArguments	...♠ {♡ body of mand arg♢ }♣ ...	♠ LCuBStartsOnOwnLine[3] ♡ MandArgBodyStartsOnOwnLine ♢ RCuBStartsOnOwnLine ♣ RCuBFinishesWithLineBreak
commands	before words♠ \mycommand♡ ⟨*arguments*⟩	♠ CommandStartsOnOwnLine ♡ CommandNameFinishesWithLineBreak
namedGroupingBraces Brackets	before words♠ myname♡ ⟨*braces/brackets*⟩	♠ NameStartsOnOwnLine ♡ NameFinishesWithLineBreak
keyEqualsValuesBraces	before words♠ key●=♡ ⟨*braces/brackets*⟩	♠ KeyStartsOnOwnLine ● EqualsStartsOnOwnLine ♡ EqualsFinishesWithLineBreak
items	before words♠ \item♡ ...	♠ ItemStartsOnOwnLine ♡ ItemFinishesWithLineBreak
specialBeginEnd	before words♠ \[♡ body of special♢ \]♣ after words	♠ SpecialBeginStartsOnOwnLine ♡ SpecialBodyStartsOnOwnLine ♢ SpecialEndStartsOnOwnLine ♣ SpecialEndFinishesWithLineBreak

[2] LSqB stands for Left Square Bracket

[3] LCuB stands for Left Curly Brace