TXT2TeX Copyright (C) 1998 — 2008 Kalvis M. Jansons

This perl script (which is part of the KalTeX package) converts plain text into something with a little LaTeX formatting. If you are reading a LaTeXed version of this "readme" file, it was made from the comments in the code of txt2tex using txt2tex to format them; if you are reading the plain text version, try running it through txt2tex (you can use "txt2tex –demo" for this on a unix system).

Written by Kalvis M. Jansons (email address `k@kalvis.com`), but based on txt2html by Seth Golub (email address `seth@aigeek.com`). So if you like it, send an email to both of us, but thank Seth the most; if you have any problems or suggestions send an email to me (Kalvis).

By default, much of LaTeX's fine structure is disabled by definitions in the .tex file header. If you need to edit the LaTeX you may need to remove or change some of these statements; or you may need to rerun txt2tex in a lower escaping mode, to add more complex structures, like tables and complex equations. I did it this way as I will use txt2tex myself mainly for non-mathematical documents, and for those, I like to be able to type % for percent etc., and paste in emails without worrying too much about all the strange symbols. Set the "-ec" flag if you want to "escape" all of LaTeX's special functions, and kill the "\", which is often the safest setting for "unknown" document formats.

DO YOU WANT A DEMONSTRATION? IF SO, SEE BELOW.

- For a trivial demo of txt2tex, type "txt2tex –info |txt2tex -ec".
  - For a nicer copy of this readme file, try "txt2tex –info |txt2tex -ec -ns -10pt".
  - Or maybe you will like the look of this better: "txt2tex –info |txt2tex -tf -ec -ns -10pt".
    - Remember, to see the nice output, type something like: "txt2tex –info |txt2tex -tf -ec -r off > readme.tex" followed by "latex readme.tex; xdvi readme.dvi".
  - On a unix or linux system try "txt2tex –demo".
- The best test is clearly to try it on one of your own plain text files.

Paper size

The paper size is set to "a4paper", but if you would like a different paper size I suggest finding the line with "a4paper" in txt2tex and changing it once and for all. This can also be changed using the "–doctype" option.

Tag syntax

In the options in the next section, the term "tag" is often used. I have used this term for many types of LaTeX mark-up instruction. The syntax for using tags with txt2tex is easy. For a simple tag, which puts a heading into a LaTeX subsection form, the tag is just "subsection". For more complex, or nested, tags the syntax is a little more complex. If, for example, you wanted all section headings to be centered, the tag to do it with would be "section{\center". You could also add a "clearpage" so each section is on a new page, and a "*" so the sections are not numbered; the tag would then be "clearpage\section*{\center". Also remember when using tags on a command line, you must take account of the normal shell escaping conventions.

Some important command line options

Note that any command line option name can contain any number of "_" to make the command line more readable, and, in fact, you only need a single "-" for any of the names listed with "–".

[(-dt|–doctype) <doctype>]

Used to set the LaTeX documentclass or documentstyle. It can be set to "null" for no doctype, which is useful if you want to add some LaTeX definitions above the definitions in the txt2tex header. For an example, see the definition of "–switch slides" at the end of txt2tex.

[-10pt|-11pt|-12pt]

Used to set the LaTeX font size. The default is 12pt. The "pt" can be dropped.

[(-up|–usepackage) <name>|off]

Sets a LaTeX "usepackage" definition. No default packages loaded.

[(-lh|–latexhooks) <name|mode>]

Used to add LaTeX instructions from files. Given a "name", it tells LaTeX to read (if they exist) the files name-HeadB, name-HeadE, name-BodyB, name-BodyE (with or without a suffix .tex); these files are read in to the beginning and end of the HEAD and the beginning and end of the BODY. Given a number, it sets the "latex-hook" mode, which controls which LaTeX input statements are added; these are 1,2,4,8 for the above files, which are bitwise ORed. If a new LaTeX-hook name is given, the mode is set to 15, i.e. all bits set. If a mode is given, and no name has been set, the default name "\jobname" is used as the name. Hooks are off by default.

Remember in LaTeX the basename of the LaTeX file is stored in the LaTeX variable "\jobname", so by using this as the base part of your LaTeX hooks, you would not have to change the LaTeX itself if you wanted to use a different set of hook files, as you would need only to change the name of the main LaTeX file.

[(-ec|–escapechars) [<mode>]]

Used to set the escape mode. The options (which can be bitwise ORed) are:

```
  1 --- escape \
  2 --- escape $
  4 --- escape ^ and _
  8 --- escape < and >
 16 --- escape &
 32 --- escape |
 64 --- escape #
128 --- escape ~
512 --- escape %
1024 --- escape "
```

(The above list shows what txt2tex does with complex formatting in the plain text document, namely puts it in a LaTeX verbatim block, at least in the LaTeX version of the documentation.) The default mode is 2046, so the LaTeX backslash is still active. Using "-ec" without a following number will escape everything, and "-ec 0" will escape nothing. Note that mode 1 also fixes a problem with a line that begins with white space and has "[" as the first non-space character.

[-bm|–batchmode]

Makes LaTeX run in its non-stopping mode, i.e. ignores any LaTeX warnings about over-full boxes etc.. Off by default.

[-nv|–noverbatim]

Stops any output being put in verbatim blocks even if it looks like it is "preformatted". This sometimes gives other subroutines a chance to format the data. Off by default.

[-sv|–splitverbatim]

Use this if verbatim blocks can be split by page breaks; the default is that they cannot.

[(-pb|–prebegin) <num>]

Sets the number of preformatted-looking lines (2 by default) needed to begin a verbatim block. The options are:

- 0 — put the entire document in a verbatim block.
- 1 — one trigger line, so even a single line can be put in verbatim.
- 2 — two trigger lines.
- 3 — same as 1, but verbatim blocks can start only after a blank line.

Less than 0 is set to 0 and more than 3 is set to 3.

[(-pe|–preend) <num>]

Sets the number of non-preformatted-looking lines (2 by default) needed to end a verbatim block. The options are from 0 to 3, with less than 0 set to 0 and more than 3 set to 3. Option 3 has the special meaning of ending the verbatim block on a blank line.

NOTE for –prebegin and –preend: If only one is zero, the other is ignored. If both are zero, the entire document is put in a verbatim block.

[(-p|–preformat) <num[,num[,num]]>]

This option sets the values of the following variables:

- $verbatim_white_min (6),
- $verbatim_min (6),
- $verbatim_post_min (3),

where the numbers in () are the defaults. If only one number is given, it sets $verbatim_white_min and $verbatim_min to this value, otherwise it sets the variables in order. A line is considered to be preformatted if either there is a non-space character followed by $verbatim_min non-word characters, or if there are at least $verbatim_white_min spaces after the start of the line and the line contains a non-space character followed by $verbatim_post_min non-word characters.

Note that tabs are expanded before these tests.

[-ns|–nosectionnumbers]

Do not number LaTeX sections. They may already have numbers, for example, or you may feel that the document looks better without them. In fact, all this really does is add a "*" at the end of the headings tags, so if you have changed these tags, be sure that "-ns" still makes sense for your tags.

[-np|–nopagenumbers]

Do not number LaTeX pages, i.e. set the pagestyle to empty.

[(-lm|–listmode) <mode>]

Sets the list mode; the bitwise ORed options are:

- 0 — automatically number and label lists, renumbering what appear to be lists with errors. Use standard LaTeX numbering and labelling.
- 1 — keep the original numbers (or letters) on enumerated lists, but put standard labels on itemized lists.
- 2 — turn itemized lists into enumerated lists.
- 4 — hrules end all active lists.
- 8 — easy start. Enumerated lists need not start with 1, A, etc., which is useful for documents that have headings, diagrams etc. in lists. You would normally use this with list mode 1, to avoid renumbering.
- 16 — turn LaTeX description environments into enumerate; this may sound a strange thing to do, but leads to nice results. Try it!
- 32 — do not nest description environments. Normally a new description starts for every new level of indentation, but this mode switches this feature off.

Using "-lm" without a following number sets the default mode 0.

[(-de|–description) <regexp>|off]

Sets the regular expressions to identify lines that should be put in a LaTeX "description" environment. Only the "first match" in the regular expression will be used as the "name" in the "description", and the rest is deleted. So, if you do not want to delete anything, put your regular expression in "()". This is off by default, and the default can be reset with the command line option "-de off". See the definitions of "-sw remind" and "-sw dict" for examples.

[(-s|–shortline) <[-]num>]

Sets the upper bound of the length of a "short line" (40 by default), which is assumed to be intentionally this short, so must be kept broken. If the number given is negative, leading spaces are not ignored when determining if a line is "short". The default is that leading spaces are ignored.

[(-ss|–shortlineskip) <length>]

Sets the vertical skip after a "short line", for example try "-ss 1ex". The default is a normal line

break. The default can be restored by setting it to "null".

[(-r|–hrule) <num>|off]

If given a number, sets the minimum number of "===" etc. for a horizontal rule. The default is 4. If given "off", sets $hrules_on = 0, and any hrules found are not printed.

[(-sm|–smallmargins) [<mode>]]

LaTeX defaults to large margins, but I like small (1in) margins. The bitwise ORed options are:

- 0 — standard LaTeX margins.
- 1 — 1in X margins.
- 2 — 1in Y margins.
- 3 — 1in X and Y margins.

The default is 0. If "-sm" is not followed by a valid number, then option 3 is set.

[(-t|–title) <title>]

You can specify a title to be placed at the top of the document.

[(-tt|–titletag) <tag>]

Used to set the title tag. The default tag is "centerline{\LARGE\bf".

[-tf/+tf] | [–titlefirst/–notitlefirst]

Use the first non-blank line as the title of the document. Off by default.

[(-pi|–parindent) <num>]

Sets the minumum number of spaces indented in first line of a paragraph. This is used only when there's no blank line preceding the paragraph. The default is 3.

[(-c|–caps) <num>]

Sets the minimum sequential CAPS for a "caps line", which is then put in a special font. For the full definition of a caps line, see the code. The default is 3.

[(-ct|–capstag) <tag>|off]

Sets the tag to put around "caps lines". Set it to "off" for no caps lines, but note that some of these lines could then be marked as solo lines; if you want to avoid this, set it to "null", which is turned into the empty tag. The default tag is "subsubsection*".

[(-st|–solotag) <tag>|off]

Sets the tag for "solo lines", i.e. lines that have a blank line before and after, and have the "right" important-looking ending (see "sub solo" for the full definition). The default tag for solo lines is "subsubsection*{\textit". Set it to "off" for no solo lines.

[(-m|–mail) [<mode>]]

Used to set the mail mode. The bitwise ORed options are:

- 1 — deal with mail headers and mail quoted text.
- 2 — add half-line width right-flushed hrules at the beginning of new messages. Strange, but easy to see!
- 4 — add a LaTeX "clearpage" before each new message.
- 8 — do not print the mail header.

"-m" without a following number sets the default mail mode of 1. (Any non-zero option also includes option 1.)

[-u/+u] | [–unhyphenate/–nounhyphenate]

Enables unhyphenation of the raw text, so we can leave hyphenation to LaTeX. On by default.

[(-ul|–ulength) <num>]

Sets the underline tolerance for plain text headings, i.e. how much longer or shorter than the text can underlines be and still be underlines. The default is 1.

[(-uo|–uoffset) <num>]

Sets the offset tolerance for underlines of plain text headings. The default is 1.

[(-tw|–tabwidth) <num>]

Sets the width of a tab. The default is 8.

[-e/+e] | [–extract/–noextract]

Sets extract mode for making inserts for other LaTeX documents. Off by default.

[(-rs|–ruleset) <file>]

[+rs|–noruleset]

By default reads the ruleset in ".txt2tex-ruleset" (if it exists), but a different file can be given. When looking for a specified ruleset file, if it fails to find a direct match, it will then try "file-ruleset" and last of all "~/.txt2tex-file", where "file" is the given file name.

[-ro/+ro] | [–rulesetonly/–norulesetonly]

Do no escaping or marking up at all, except for processing the ruleset dictionary file and applying it. This is useful if you want to use txt2tex's rulesetting feature on a LaTeX document. If the LaTeX is a complete document (includes HEAD and BODY) then you will need to use the –extract option also. Off by default.

[(-H|–heading) <regexp>]

Used to set regular expressions to pick out custom headings in the plain text. For examples, see the "switch" options at the end of txt2tex, in particular "num". Header levels are assigned by regexp in the order seen; when a line matches a custom header regexp, it is tagged as a header. If it is the first time that particular regexp has matched, the next available header level is associated with it and applied to the line. Any later matches of that regexp will use the same header level. Therefore, if you want to match numbered header lines, you could use something like this:

-H '^ *\d+\. \w+' -H '^ *\d+\.\d+\. \w+' -H '^ *\d+\.\d+\.\d+\. \w+'

Then lines like:

2. Examples
2.1. More Examples
2.1.1. Even More Examples

would be marked as section, subsection, etc., assuming they were found in that order, and that no other header styles were found. If you prefer that the first heading specified always becomes "section", the second always becomes "subsection" etc., then use the –explicitheadings option. Also you would probably want the –nosectionnumbers option, to avoid getting two sets of numbers; this could also be fixed using the –trimheadings option (see the definition of "–switch n").

[(-HT|–headingtags) <tag1[,tag2...]>|shift|number]

[(-TH|–trimheadings) <regexp>]

The sequence of tags for the section headings can be set by something like: "-HT something,anotherthing,..." and the headings can be trimmed using "-TH <regexp>", i.e. whatever matches "regexp" is removed. Note that all headings are trimmed using the same regular expression and that the regular expression is applied after the heading tag and label have been added. The argument of "-HT" can also be "shift", which shifts the sequence of heading tags down by one, or "number", which tells txt2tex (rather than LaTeX) to number the headings (off by default). Remember not to ask LaTeX to number the headings too, if you use "number".

[-EH/+EH] | [–explicitheadings/–noexplicitheadings]

This tells txt2tex not to try to find any headings except the custom ones specified. Also, the custom headings will not be assigned levels in the order they are encountered in the document, but in the order they are specified on the command line. Off by default.

[(-db|–debug) <num>]

Debug mode for ruleset dictionaries. Bitwise OR what you want to see:

- 1 — the parsing of the dictionary.
- 2 — the code that will make the ruleset.

[(-tr|–trim) <num|regexp>]

Used to trim "n" characters from the beginning of each line longer than "n", or to trim using a regular expression. The default is 0.

[(-sw|–switch) <keyword>]

Used to add sets of command line options that are kept at the bottom of this file. For example "-sw num" will help pick out numbered section headings, and "-sw lynx" cleans up text files from the lynx browser. Take a look at the definition of "-sw num", and see if you can work out what all the options do. Then add some "-sw" options of your own. Also see the section on option sets below.

[-tc|–twocolumn]

Sets LaTeX's "twocolumn" option. Off by default. To see what this looks like with 1in margins, take a look at this "readme" file in this format by typing "txt2tex –demo" on a unix or linux machine.

[-ls|–landscape]

Sets LaTeX's "landscape" option. Off by default.

[-sp|–sloppy]

Sets LaTeX's "sloppy" option, which is particularly useful for slides. Off by default.

[-d|–draft]

Save the output in a file called draft.tex. Off by default.

[(-h|–help)/–info/–demo]

–help gives a short help message listing the options, –info gives a plain text version of the "readme" file, and –demo (on a standard unix or linux system) will run the plain text from –info through txt2tex to give a nice LaTeXed version of the "readme" file; note that the "demo" makes t2t_readme.txt, .tex, .dvi, .aux, and .log.

[-v|–version]

Prints the txt2tex version number.

Option sets

Below the "__END__" in txt2tex you can put lists of command line options after a "keyword"; these can then be loaded by putting "-sw keyword" on the command line. Note that "\" is a continuation character, so long options can be put on several lines. These include:

remind — turns the output of the unix remind program into nice LaTeX; call remind using "rem -n |sort".

num — picks out simple numbered headings.

n — a variant of the above.

plain — a very plain style, which is good for university work!

trim — removes leading spaces before txt2tex processes the line.

lynx — for lynx browser output.

noL — normally \014 produces a LaTeX "clearpage", but this option removes \014 before txt2tex sees the line.

HH — this is what I use to print the "Happy Hacker" newsletter.

man — useful for dealing with unix man pages, but could be better!

pagesec — each new section starts on a new page.

pagesubsec — each subsection starts on a new page.

slides — turns plain text into (very) simple slides. You might also want to set "noverbatim". Note that many of the standard options will not work with switch "slides" set.

handout — used for student handouts.

letter — used for writing letters, but you need to define your own letter-hook files with your address etc.

preview — not for LaTeXing, but marks up the file in a manner to show you what txt2tex was thinking; this can help in choosing the right tags etc. for the print run. It can be followed by other options, so you can see how that changes the mark up. It is also useful for debugging, but that is probably my job [:-)]

dict — turns a list of the form 'word: text' into a LaTeX description environment.

phone — turns a list of the form 'phrase: text' into a LaTeX description environment. I use this for a personal phone book.

fn — turns fancy numbered lists, with numbers like 1.1.1, into LaTeX description environments. Often useful for printing contracts off the net!

lpr — used as part of a fancy plain text printer filter.

lpn — used by the Lockpicker Network.

netrc — used to print a .netrc file.

A sample ruleset

Txt2tex by default tries to load a file called ".txt2tex-ruleset" from your home directory (assuming you are using a unix system). This file, if it exists, contains transformation rules that are executed AFTER all other txt2tex subroutines with the exception of "tidy" (which does a little cleaning up) and the escaping of "funny" characters. Strange behaviour can result from not keeping the time of execution in mind.

I most often use "rulesets" for writing my own documents in plain text, to be transformed later by txt2tex into LaTeX. So let us look at rules that help in such tasks. Each rule must be on a single line in the ruleset file.

/<<(.*?)>>/ -f-> $1

The "-f->" type rule, when the regular expression on the left matches, takes the expression on the right and turns it into a footnote, then removes the triggering text. So the above example transforms "Kalvis M. Jansons<<Mathematics, UCL>>" into "Kalvis M. Jansons\footnote{Mathematics, UCL}" in the LaTeX output.

Kalvis M. Jansons -Fo-> Email: kalvis\@jansons.org

The "-F->" type rules are the same as the "-f->" ones, but do not remove the triggering text. So the above rule adds a footnote with my email address to my name. So that this happens once only per document, I have added the "o" (for once) in the rule.

/txt2tex/ -oi-> TXT2TeX \\emph{(written by Kalvis)}

/pheonix/ —> phoenix

The above rules are simple transformations, the first is case insensitive, hence the "i", and is executed once only. The second corrects a common spelling error (every time it occurs).

/tagad/ -ie-> my $time = localtime(time); $time =~ s/\:\d\d\s.*//; $time

The "e" option means evaluate the righthand side as a perl expression. So the above expression turns "tagad" (the Latvian for "now") into the current date and time (and removes "tagad"). The "e" option can also be used to change the value of txt2tex parameters while running, by setting them when certain patterns are first encountered.

/*([a-z][a-z]*[a-z])*/ -ti-> emph

/*([a-z])*/ -ti-> emph

The "t" option is used to tag the text in (), so leads to a shorter rule than could be obtained using the above rules to do this job. The above rules put any sequence of letters and spaces which are between two stars in the LaTeX "emph" style. This use of "*" is often seen in plain text "readme" files.

/<*(.*?)*>/ -tfi-> textbf

Putting a few bits together, we can turn anything in <* ... *> into a "textbf" footnote, but I am sure you can think of a better application.

Saving the sample ruleset

If you want to save this sample ruleset to adapt for your own use, type "txt2tex -sampleruleset >

~/.txt2tex-ruleset",

or direct it into a different file if you do not want it to be the default.

Getting help

Please contact me (Kalvis) with any problems or suggestions.

Bugs

Send any bug reports to me, and I will do my best to fix them, but note that there is a limit to what txt2tex can be expected to do on poorly formatted text files. For such files, it is often better to fix the worst features before giving them to txt2tex; then there should not be the need to do much work, if any, on the LaTeX file produced.

Ensure that you are using the latest version, which can be obtained from any CTAN site.

Kalvis@Jansons.org