ChkTeX v1.2

Jens T. Berger Thielemann

January 25, 1996

1 Introduction

This program was written in frustration of that some constructs in LaTeX are sometimes non-intuitive, and easy to forget. It is *not* a replacement for the built-in checker in LaTeX; however it catches some typographic errors LaTeX oversees. In other words, it is Lint for LaTeX.

While written on an Amiga, it is written in ANSI C, so you can use this at your UN*X/MSDOS/whatever site also. Full source included.

The program also supports output formats suitable for further processing by editors or other programs, making errors easy to catch. An ARexx script for interfacing with SCMSG (and via that, other editors) is included. A special script for CygnusED and an experimental script for GoldED is included.

The program itself does not have any machine requirements, an Amiga compiled version is included which should work on all Amigas. Of course, you'll need ARexx and SCMSG to benefit from the ARexx scripts, though.

2 Legal stuff

3 Installation

Amiga users should use the supplied Installer installation script.

If you are using a different platform, compile and link all .c files, and move the binary to a suitable place. The "`.chktexrc`" file should be moved to your

`$HOME` directory on UNIX; other platforms may put its directory path in a environment variable named `CHKTEXRC`.

You may wish to read section 7 first.

After doing this, you may enhance ChkTEX' behaviour by reading/editing the "`.chktexrc`" file.

4 New features

Changes from v1.1:

Added:

- Warns about '`"`' and '`´`' now — advices about use of better accenting characters.
- The user may now specify own patterns that will be searched and warned about.
- You may now make it dump comments; sometimes useful (default: off).
- Will now search for user-specified patterns.
- 5–6 other new warnings. We've got a total of 25 warnings now!
- Wildcard matching (Amiga only). Will now match file patterns internally, which should save a lot of work. This is, however, platform-specific code — on UNIX boxes this is done by the shell.
- "`chktex ?`" now gives help, too.
- Command-line options may be specified globally in the "`.chktexrc`" file.
- You may now use another string than a single colon to split fields in the `-v0` output format. This will solve problems on machines using colons in path names.
- VT100 and VT102 support added. `-v2` switch now works on such terminals, too. Checks `$TERM` environment variable.
- "Debugging" mode.
- An ARexx script for interfacing with SCMSG, the SAS/C message browser.
- A Perl script which transforms the `-v0` output into a UNIX-error like.

Modified:

- More intelligent "`.chktexrc`" format; it is now in a more shell/free-form style, thus more readable. Introduced C-like escape codes (using '`!`' as escape character instead of '\ ').
- More logical option names & better option parsing.
- A few more options, too...
- Dash-checking has been improved.
- Better error reporting of program errors; routines have been rewritten.
- ASCII version of the document is discontinued; users needing this product are assumed to possess LATEX (or at least, a DVI/PS viewer).
- Localization has also been discontinued... :-/ Too much fuzz when nobody is willing to write translations.

5 Usage

Unfortunately, the template has changed *slightly* from v1.1 to provide a more logical format. Thus, a *very* few scripts may be incompatible, more precisely those using either the `-g`, `-b`, `-x` or `-t` switches in peculiar ways (saying these twice on a commandline is now the same as not saying them at all). This problem is assumed to be neglible.

Template: A UNIX-compliant template format follows:

```
chktex  [-hiqr] [-v[0|1|2]] [-l <rcfile>]
        [-[w|e|m|n] <[1-25]>] [-d <number>]
        [-o <outputfile>] [-b[0|1]] [-t[0|1]]
        [-x[0|1]] [-g[0|1]]
```

Options: These are the options ChkTEX currently accepts. Please note that single-lettered options requiring a numerical or no argument may be concatenated. E.g. saying "`-v0qb0w2`" is the same as saying "`-v0 -q -b0 -w2`", except for being less to type.

Enough general talk; here's a rather detailed description of all options:

Misc. options: General options which aren't related to some specific subpart of ChkTEX.

- `-h [--help]` Gives you a command summary.
- `-i [--license]` Shows distribution information.
- `-l [--localrc]` Will read a resource-file formatted as the global resource-file "`.chktexrc`", in addition to the global resource-file. This option needs the name of the resource-file as a parameter. See also `-g`.
- `-r [--reset]` This will reset all settings to their defaults. This may be useful if you use the `CMDLINE` directive in your "`.chktexrc`" file, and wish to do something unusual.
- `-d [--debug]` Given a numeric argument above zero, it will output some status information during run-time.

Muting warning messages: Controls whether and in what form error messages will appear.

- `-w [--warnon]` Makes the message number passed as parameter a warning and turns it on.
- `-e [--erroron]` Makes the message number passed as parameter an error and turns it on.
- `-m [--msgon]` Makes the message number passed as parameter a message and turns it on. Messages are not counted.
- `-n [--nowarn]` Turns the warning/error number passed as a parameter off.

Output control flags: Determines the appearance and destination of the error reports.

- `-q [--quiet]` Shuts up about copyright information.
- `-o [--output]` Normally, all errors are piped to stdout. Using this option with a parameter, errors will be sent to the named file instead. Only information relative to the LATEX file will be sent to that file. Memory problems, etc., will as as always be sent to `stderr`. If a file with the name given already exists, it will be renamed to '`foobar.bak`', `foobar` being the name of the file. See also `-b`.

- `-v [--verbosity]` Specifies how much and how you wish the error reports to be displayed. This option currently accepts the following numerical levels:
  - **0** Will show the information in a way that should be suitable for further parsing by `awk`, `sed` or similar. The format is as follows:

    `File:Line:Column:Warning number:Warning message`

    The colons may be replaced with another string; use the `-s` switch for this.
    This format will also turn off any twirling baton's, and other things which make the program unsuitable for pipes and batch scripts. As the program does yet not output all errors in quite order, this output format is also suitable for piping through '`sort`'.
  - **1** Shows the information in a way which is more comprehensible for humans, but which still doesn't need anything but a glass tty.
  - **2** Shows the information in a fancy way, using escape codes and stuff. It is the indeed most readable of all modes; however, it needs proper set up of the '`chktex.h`' file.

  The default is mode 1, using `-v` without any parameter will give you mode 2.
- `-s [--splitchar]` String to use instead of the colons when doing `-v0`; e.g. this string will be output between the fields.

Boolean switches: Common for all of these are that they take an optional parameter. If it is `0`, the feature will be disabled, if it is `1`, it will be enabled. All these features are on by default; and are toggled if you don't give any parameter.

- `-b [--backup]` If you use the `-o` switch, and the named outputfile exists, it will be renamed to `filename.bak`.
- `-g [--globalrc]` Read in the global resource file. This switch may be useful together with the `-l` option.
- `-t [--tictoc]` Display a twirling baton, to show that we're working. `-v0` does an `-t0`, too, as it assumes that the user then uses the program non-interactively.
- `-x [--wipeverb]` Ignore the '`\verb`' command found within the LaTeX file and its argument is completely by the checking routines. This is done by simply overwriting them. If you somehow don't like that (for instance, you would like to count brackets inside those commands, too), use this switch.

The parameters `-m`, `-w`, `-e` and `-n` may need some further explanation. With these it is possible to control muting of certain warnings. Currently, there is no real difference between errors and warnings, except that they are counted separately.

If you don't specify any input LaTeX-files on the commandline, we'll read from `stdin`. To abort `stdin` input on the Amiga, press `Ctrl` + `\`. By default, we're using the 1994 version of GNU's `getopt()` routine. For those of you who have only used the Amiga's ReadArgs(), here are some quick instructions:

- Options may be given in any order; the names of the LaTeX-files do not have to be the last arguments. This behaviour may be turned off by creating an environment variable named '`POSIXLY_CORRECT`'.

- The special argument '`--`' forces an end of option-scanning.
- Long-named options begin with '`--`' instead of '`-`'. Their names may be abbreviated as long as the abbreviation is unique or is an exact match for some defined option. If they have an argument, it follows the option name in the argument, separated from the option name by a '=', or else the in next argument.

You should also take a look at the "`.chktexrc`" file. The method for finding has been changed, it is now done as follows:

1. Look for it in the current directory.
2. Look for it in the directory pointed to by the environment variable `CHKTEXRC`.
3. Look in the following directories:

Machine	**Directory**
2.04+ Amiga	`ENV:`
1.3 Amiga	`S:`
UNIX	`$HOME`

It should be rather self-explanatory, and should help you customize the program to your own needs.

6 Explanation of error messages

Warning/error 1: `Command terminated with space.`

You tried to terminate a command with a blank space. Usually, this is an error as these are ignored by LaTeX. In most cases, you would like to have a real space there.

Example:

```
\LaTeX_is a typesetter.
```
LaTeXis a typesetter.
```
\LaTeX\ is a typesetter.
```
LaTeX is a typesetter.

Warning/error 2: `Non-breaking space ('~') should have been used.`

When reading a document, it is not very pretty when references are split across lines. If you use the ~ character, LaTeX will assign a very high penalty for splitting a line at that point. ChkTeX issues this warning if you have forgot to do this.

Example:

```
Please refer to figure_\ref{foo}.
```
Please refer to figure 11.
```
Please refer to figure~\ref{foo}.
```
Please refer to figure 11.

Warning/error 3: `You should enclose the previous parenthesis with '{}'.`

This is a warning which you may ignore, but for maximum aestethic pleasure, you should enclose your bracket characters with '{}'s.

Example:

```
$_[(ab)^{-1}]_\^{-2}$
```
$[(ab)^{-1}]^{-2}$

```
${[{(ab)}^{-1}]}^{-2}$
```
$${[{(ab)}^{-1}]}^{-2}$$

Warning/error 4: `Italic correction ('\/') found in non-italic buffer.`

If you try to use the `\/` command when ChkTEX believes that the buffer is not outputted as italic, you'll get this warning.

Example:

```
This is an\/ example
```
This is an example.

```
This is an example.
```
This is an example.

Warning/error 5: `Italic correction ('\/') found more than once.`

If the buffer is italic, and you try to use the `\/` command more than once, you'll get this warning.

Example:

```
This {\it example\/\/} is not amusing.
```
This *example* is not amusing.

```
This {\it example\/} is not amusing.
```
This *example* is not amusing.

Warning/error 6: `No italic correction ('\/') found.`

You get this error if ChkTEX believes that you are switching from italic to non-italic, and you've forgot to use the `\/` command to insert that extra little spacing. If you use the `em` option, you may ignore this warning.

Example:

```
This {\it example_} is not amusing, either.
```
This *example* is not amusing, either.

```
This {\it example\/} is not amusing, either.
```
This *example* is not amusing, either.

Warning/error 7: `Accent command 'command' needs use of 'command'.`

If you're using accenting commands, 'i' and 'j' should lose their dots before they get accented. This is accomplished by using the `\i`, `\j`, `\imath` and `\jmath` command.

Example:

```
This is an example of use of accents: \'{i}.
```
This is an example of use of accents: í.

```
This is an example of use of accents: \'{\i}.
```
This is an example of use of accents: í.

Warning/error 8: `Wrong length of dash may have been used.`

This warning suggests that a wrong number of dashes may have been used. This check has now gotten somewhat better than previous version. It will now check the both the character in front and after the dashes. If they are of the same type, ChkTEX will use the table below to determine how many dashes there should be; if not, it will shut up and accept that it doesn't know.

Character type	**# of dashes**
Space	3
Number	2
Alphabetic character	1

This is more or less correct, according to my references. Hopefully this check can be even more improved (suggestions?).

Example:

```
It wasn't anything - just a 2---3 star--shots.
```
It wasn't anything - just a 2—3 star–shots.

```
It wasn't anything --- just a 2--3 star-shots
```
It wasn't anything — just a 2–3 star-shots.

Warning/error 9: `'character' expected, found 'character'.`

You get this warning when you try to mix brackets — ChkTEX expect to find matching brackets in the same order as their opposites were found. While not an explicit error, it is usually a sign that something is wrong.

Warning/error 10: `Solo 'character' found.`

This warning is triggered if we find a single, *closing* bracket. While not an explicit error, it is usually a sign that something is wrong.

Warning/error 11: `You should use '\dots' to achieve an ellipsis.`

Simply typing three '.' in a row will not give a perfect spacing between the '.'s. The `\dots` is much more suitable for this.

Example:

`I don't know quite...I think it's wrong.`
I don't know quite...I think it's wrong.

`I don't know quite\dots I think it's wrong.`
I don't know quite...I think it's wrong.

Warning/error 12: `Interword spacing ('\ ') should been used.`

One of the specified abbreviations were found. Unless you have previously said `\frenchspacing`, you'll have incorrect spacing, which one should avoid if possible.

Example:

`This is an example, i.e._an demonstration.`
This is an example, i.e. an demonstration.

`This is an example, i.e.\ an demonstration.`
This is an example, i.e. an demonstration.

Warning/error 13: `Intersentence spacing ('\@') should perhaps been used.`

LaTeX' detection of whether a period ends a sentence or not, is only based upon the character in front of the period. If it's uppercase, it assumes that it does not end a sentence. While this may be correct in many cases, it may be incorrect in others. ChkTeX thus outputs this warning in every such case.

Example:

`I've seen an UFO!_Right over there!`
I've seen an UFO! Right over there!

`I've seen an UFO\@! Right over there!`
I've seen an UFO! Right over there!

Warning/error 14: `Could not find argument for command 'command'.`

ChkTeX will in some cases need the argument of a function to detect an error. As ChkTeX currently processes the LaTeX file on a line-by-line basis, it won't find the argument if the command which needed it was on the previous line. On the other hand, this *may* also be an error; you ought to check it to be safe.

Example:

`$\hat$`
This will give a LaTeX error...

`$\hat{a}$`
$\hat{a}$

Warning/error 15: `No match found for character 'character'.`

This warning is triggered if we find a single, *opening* bracket. While not an explicit error, it is usually a sign that something is wrong.

Warning/error 16: `Mathmode still on at end of LaTeX file.`

This error is triggered if you at some point have turned on mathmode, and ChkTEX couldn't see that you remembered to turn it off.

Warning/error 17: `Number of 'character' doesn't match the number of 'character'!`

Should be self-explanatory. ChkTEX didn't find the same number of an opening bracket as it found of a closing bracket.

Warning/error 18: `` You should use either `` or '' as an alternative to `"'. ``

Self-explanatory. Look in the example, and you'll understand why.

Example:

`This is an "example"`
This is an "example"

``This is an ``example''``
This is an “example”

Warning/error 19: `You should use "'" (ASCII 39) instead of "´" (ASCII 180).`

On some keyboards you might get the wrong quote. This quote looks, IMHO, *ugly* compared to the standard quotes, it doesn't even come out as a quote! Just see in the example.

Example:

``` ``There´s quotes and there´s quotes´´ ```
“There´s quotes and there´s quotes´´

``` ``There's quotes and there's quotes'' ```
“There's quotes and there's quotes”

Warning/error 20: `User-specified pattern found.`

A keyword you've specified using `USERWARN` in the "`.chktexrc`" file, has been found.

Warning/error 21: `This command might not be intended.`

I implemented this because a friend of mine kept on making these mistakes. Easily done if you haven't gotten quite into the syntax of LaTeX.

Example:

```
\LaTeX\ is an extension of \TeX\. Right?
```

LaTeX is an extension of TeXṘight?

```
\LaTeX\ is an extension of \TeX. Right?
```

LaTeX is an extension of TeX. Right?

Warning/error 22: `Comment displayed.`

This warning is by default off. If you turn it on, ChkTeX will dump all comments it finds, which in some cases is useful. I usually keep all my notes in the comments, and like to review them before I ship the final version. For commenting out parts of the document, the `comment` environment is better suited.

Warning/error 23: `Either ''\,' or '\,'' will look better.`

This error is generated whenever you try to typeset three quotes in a row; this will not look pretty, and one of them should be separated from the rest.

Example:

```
```Hello', I heard him said'', she remembered.
```

"'Hello', I heard him said", she remembered.

```
``\,`Hello', I heard him said'', she remembered.
```

" 'Hello', I heard him said", she remembered.

Warning/error 24: `Delete this space to maintain correct pagereferences.`

This message, issued when a space is found in front of a `\index`, `\label` or similar command (can be set in the "`.chktexrc`" file). Sometimes, this space may cause that the word and the index happens on separate pages, if a pagebreak happens just there.

You might also use this warning to warn you about spaces in front of footnotes; however, the warning text may not be entirely correct then.

Example:

```
Indexing text_\index{text} is fun!
Indexing text\index{text} is fun!
```

Warning/error 25: `You might wish to put this between a pair of '{}'`
```

This warning is given whenever ChkTEX finds a '^' or a _ followed by either two or more numberic digits or two or more alphabetic characters. In most situations, this means that you've forgotten some {}'s.

Example:

`$5\cdot10^10$`

$5 \cdot 10^1 0$

`$5\cdot10^{10}$`

$5 \cdot 10^{10}$

7 Porting

Porting should be rather easy. However, to ensure that the porting proceeds painless, read the first lines of the '`ChkTeX.h`' file. Not very much to do, and it will make life easier later. By the way: Set tabsize to 4.

To compile it, just `cd` to the `source` directory, and compile and link all `.c` files. On UNIX, type the following in a shell: `gcc -o chktex *.c`

You may also wish to modify some of the routines in `OpSys.c` — unless you compiling the program on a UNIX or Amiga box.

8 Bugs

No fatal ones, I think, but the program currently has some problems when a LaTeX command/parameter stretch over a two lines — some extra spaces may be inserted into the input. I regard the program as fairly well tested; using the SAS/C `cover` utility I was able to make sure that approximately 95% of the code has actually been run successfully in the final version. Which leaves about 140 possibly buggy lines, most of these are procedure terminating brackets or 'can't happen' lines, though.

The detection of whether we're in math mode is not perfect. The program does not trust this check very much, either. The only codes which are recognized for getting in/out of math mode are: '`$`', '`$$`', '`\(`', '`\)`', '`\[`' and '`\]`'. Environments like `\begin{displaymath}`, user-defined commands, etc., do thus not work. This is usually no problem, as it just creates a bit more warnings than needed.

We've got some of the same problems when isolating the arguments of a command. Although improved, it will certainly fail in certain cases. Currently, this should cause nothing but a few extra warnings.

Before submitting a bug report, please first see whether the problem can be solved by editing the "`.chktexrc`" file appropiately.

9 Future plans

In a somewhat prioritized sequence, this is what I'd like to put into the think — if I have the time.

- A `configure` script for Unix boxes.
- An Emacs-hook which does much the same as the ARexx script.

- Probably some more warnings/errors; just have to think them out first. Suggestions are appreciated. I'll also see what I can implement from `lacheck`, a similar program. Currently, I can think of the following:
  - Quote nesting plus advices on which way quotes should be put.
  - `\ldots` where `\cdots` should have been used, and vice versa.
  - Space before footnotes (although this could be accomplished by editing the `POSTLINK` field of the "`.chktexrc`" file).
  - Whitespace before punctuation.
  - Support for `\input` — read the file!
  - Emacs compile mode compatible output (this can also be achieved by an appropiate filter, though).
  - Support for some environments, like `verbatim`, etc.
- Get indentation consistent. I used to favour Allman style when doing C, however, I was forced to use K&R style by Emacs' Perl-mode. And, as I must admit, it is more suitable once you've gotten into the braces.

  A result of this is that old code uses Allman style, while new code uses K&R.
- Amiga Workbench support. It would be nice to put shift-clicking into the program; until this is implemented by someone, use the included ARexx program.
- Put some garbage collecting into the thing. We're buffering almost one half of the lines in the file for the length of the program. 99% of these can be expunged *much* earlier. However, if you're dealing with so large files that you get out-of-mem errors, please consider splitting and using `\input` instead.

  Besides; if ChkTEX runs out of memory when processing your files, LATEX will indeed do the same!
- Rewrite the thing in Perl? Get regexp's for free, at least.

10 Notes

10.1 Wish to help?

As most other living creatures, I have only a limited amount of time. If you like ChkTEX and would like to help improving it, here's a few things I would like to receive. Think of it as giftware — if you like it, you help improving it. The following ideas are given:

- If somebody would like to write an Emacs interface for ChkTEX, it would be *very* welcomed. I don't know Emacs-LISP, and haven't got the time to get into it. You should be able to model a gcc interface to suit your needs.

  To a certain degree, however, this is possible already. In Emacs, using AUC TeX, you can type `M-x compile RET chktex.pl texfile.tex`, and you should be able to browse through the error messages. Then, `` C-x ` `` to parse the messages.

- If you have access to a dictionary listing English abbreviations, I would appreciate an updated version of the "`.chktexrc`" file with these filled in. In fact, if you update the "`.chktexrc`" file in anyway that is not strictly local, I would appreciate to receive your updated version.
- ARexx interfaces for other editors are also welcomed; these should be rather fast to write. They should to the following:
  1. Get the filename of the active file.
  2. If possible, save the file to disk if there has been any changes.
  3. Call the program '`ChkTeX.rexx`' with the filename as the only parameter.
- If somebody out there actually possess (and uses) GoldED, it would be nice if they checked whether the ARexx script included actually work. If not, please send me a fixed copy; perhaps also one which supports point 2 above, too.

  I don't have GoldEd in my possession; the script was just modelled after Juergen Zeschky's, (`<juergen@sokrates.nbg.de>`) PGP↔GoldED interface.

Of course, people doing any of this will be mentioned in the doc and readme, and thus receive eternal glory and appreciation.

10.2 Caps and stuff

This program uses the `getopt()` routine, as supplied from GNU. The source included in this distribution has been modified slightly. To make the use of C2LOCAL easier, portions which were `#ifdef`'ed out, have now been commented out.

Where trademarks have been used, the author is aware of that they belong to someone, and has tried to stick to the original caps.

11 About the author

`#ifdef EGO_TRIP`

> A quick summary of who I am and what I do:
>
> I'm 20 years old, and live in Oslo, the capital of Norway. I'm currently studying maths and computer science at the University of Oslo; planning to get a degree within mathematical modelling, with a dash of physics and emphasing the computer part of the study.
>
> At home I now possess 4 computers, of which 1 is regular use: A vanilla Amiga 1200, expanded only by a HD. The others are a `80286` PC and an Amiga 500, both semi-out-of-order. The last one is a Commodore VIC-20, which for some peculiar reason never seems to be used.
>
> Most of the time in front of these computers (including SGI Indy's and SPARC stations at our university) is spent on C and shell programming, plus some textprocessing. I am also involved in writing the document for Isaac — Interactive Simulation as an Alternative to Advanced Calculations. This is planned to help newcomers

to physics, by providing a computer program which enables one to simulate most experiments relating to classical mechanics.

C and shell programming is not my only knowledge areas regarding computers, however. I write the following languages more or less: Perl, Motorola `68000` assembly code, ARexx, Simula, C++, LaTeX, HTML, Amos Basic and Installer LISP. Once I also mastered Commodore Basic V2 (the one included with my VIC-20 :-)).

However, I also try to not to end up as a computer nerd. Thus, in addition to the obligatory (?) interest for computers, I am a scout. Still running into the woods, climbing the trees, falling down and climbing up once more, in other words. To be more specific, I am a now a troop leader for 'Ulven' scoutgroup; Norwegian Scouts Association. I am also a active rover in 'Vålerenga' scoutgroup.

Certainly a lot more to tell; but I'll stop here before you fall asleep...

```
#endif
```

12 Thanks

The author wishes to thank Kasper B. Graversen, for lots of creative suggestions and improvements. Thanks also goes to Frank Luithle, being the only one writing a translation for v1.0. Unfortunately, he remained unreachable after that...:-/

13 Contacting the author

I am currently hunting for a rather powerful wildcard pattern-matching/regular expressions routine; preferably one more or less equivalent to the one of `csh`, AmigaOS' `MatchPattern()` or similar. If you have access to a freely distributable version of such a routine, I would be pleased if you would mind contacting me at the below address. It is meant to be used in a future version of ChkTeX.

If you wish to contact me for whatever reason (more tests, bug reports, suggestions, hellos, smiley's, etc.) or would like to participate in the development of ChkTeX, write to:

Jens Berger
Spektrumvn. 4
N-0666 Oslo
Norway
E-mail: `<jensthi@ifi.uio.no>`

Any signs of intelligent life is welcomed; that should exclude piracy.
Have fun.