If nil, just check the current file. Used when checking if any files have changed.

When performing spell checking on a document or a region (invoked through AUCTEX's 'Spell' command or `M-x ispell RET`), you want the spell checking program to skip certain macro arguments and environments, most notably the arguments of referencing macros and the contents of verbatim environments. The skipped parts are controlled by variable `ispell-tex-skip-alists` provided by `ispell.el`. AUCTEX has a library which can be added to this variable depending on the value of `TeX-ispell-extend-skip-list` which is set to `t` by default.

`TeX-ispell-extend-skip-list`   [User Option]
> This boolean option controls whether AUCTEX activates its extension for skipping certain macro arguments and environments when spell checking.
>
> When non-`nil`, AUCTEX loads the file `tex-ispell.el` and adds its content to `ispell-tex-skip-alists`. This library can and will never be complete, but the interface can be used to add selected and private macro names within your init file or on a file local basis.
>
> `ispell-tex-skip-alists` has the following structure:
>
> ```
> (defvar ispell-tex-skip-alists
>   '((;; First list
>      ("\\\\addcontentsline"        ispell-tex-arg-end 2)
>      ("\\\\\\\\([aA]lph\\|arabic\\)"  ispell-tex-arg-end)
>      ("\\\\makebox"                ispell-tex-arg-end 0)
>      ("\\\\documentclass" . "\\\\begin{document}"))
>     (;; Second list
>      ("\\(figure\\|table\\)\\*?"  ispell-tex-arg-end 0)
>      ("list"                      ispell-tex-arg-end 2)
>      ("verbatim\\*?" . "\\\\end{verbatim\\*?}")))
>   "*Lists of regions to be skipped in TeX mode.
> First list is used raw.
> Second list has key placed inside \\begin{}.")
> ```
>
> Each item is an alist and the structure of it is described in `ispell-skip-region-alist`:
>
> ```
> (defvar ispell-skip-region-alist
>   '((...))
>   "Alist expressing beginning and end of regions not to spell check.
> The alist key must be a regular expression.
> Valid forms include:
>   (KEY) - just skip the key.
>   (KEY . REGEXP) - skip to the end of REGEXP.
>                     REGEXP may be string or symbol.
>   (KEY REGEXP) - skip to end of REGEXP.  REGEXP must be a string.
>   (KEY FUNCTION ARGS) - FUNCTION called with ARGS
>                           returns end of region.")
> ```
>
> Let's go through the first list of `ispell-tex-skip-alists` line by line:
>
> ```
> ("\\\\addcontentsline"        ispell-tex-arg-end 2)
> ```

`KEY` is the string `"\\\\\addcontentsline"`, `FUNCTION` is `ispell-tex-arg-end` called with `ARGS`, here 2. `ispell-tex-arg-end` is a function provided by `ispell.el` which skips as many subsequent optional arguments in square brackets as it sees and then skips `ARGS` number of mandatory arguments in braces. Omitting `ARGS` means skip 1 mandatory argument. In practice, when you have something like this in your document:

```
\addcontentsline{toc}{chapter}{Some text}
```

The first two arguments are left out and 'Some text' will be spell checked. For the next line

```
("\\\\\\\([aA]lph\\|arabic\\)"  ispell-tex-arg-end)
```

the name of the counter as argument is skipped. Next line is

```
("\\\\\makebox"                    ispell-tex-arg-end 0)
```

where only optional arguments are skipped, the first mandatory argument is checked, e.g.

```
\makebox[0pt][l]{Some text}
```

Finally, the next line

```
("\\\\\documentclass" . "\\\\\begin{document}"))
```

ensures that the entire preamble of a document is discarded. Second list works the same; it is more convenient for environments since `KEY` is wrapped inside `\begin{}`.

AUCTeX provides two functions to add items to car and cdr of `ispell-tex-arg-end`, namely `TeX-ispell-skip-setcar` and `TeX-ispell-skip-setcdr`. The argument of these functions is exactly as in `ispell-tex-skip-alists`. Additions can be done via init file, e.g.:

```
(eval-after-load "tex-ispell"
  '(progn
     (TeX-ispell-skip-setcar
      '(("\\\\\mymacro" ispell-tex-arg-end)))
     (TeX-ispell-skip-setcdr
      '(("myverbatim" . "\\\\\end{myverbatim}")))))
```

Another possibility is to use file local additions at the end of your TeX file, e.g.:

```
%%% Local Variables:
%%% mode: latex
%%% TeX-master: t
%%% eval: (TeX-ispell-skip-setcar '(("\\\\\mymacro" . "{[-0-9]+}")))
%%% End:
```

Finally, AUCTeX provides a function called `TeX-ispell-tex-arg-end` which sees more arguments than `ispell-tex-arg-end`. Refer to its doc string for more information.

4.1.3 Options for TeX Processors

There are some options you can customize affecting which processors are invoked or the way this is done and which output they produce as a result. These options control if DVI or PDF output should be produced, if TeX should be started in interactive or nonstop mode, if

Function Index

Variable Index

A

C

D

F

J

Concept Index