

1 The Esperanto language

The file `esperanto.dtx`¹ defines all the language-specific macros for the Esperanto language.

For this language the character `^` is made active. In table 1 an overview is given of its purpose. In Unicode engines, avoid shorthands to enter characters with hat – whenever possible, type them directly.

<code>^c</code>	gives \hat{c} with hyphenation in the rest of the word allowed, this works for c, C, g, G, H, J, s, S, z, Z
<code>^h</code>	gives \hat{h} , but with the attribute <code>lowhhat</code> , prevents it from becoming too tall
<code>^j</code>	gives \hat{j}
<code>^u</code>	gives \hat{u} , with hyphenation in the rest of the word allowed
<code>^U</code>	gives \hat{U} , with hyphenation in the rest of the word allowed
<code>^ </code>	inserts a <code>\discretionary{-}{-}{-}</code>

Table 1: The functions of the active character for Esperanto.

Change in 1.5. Previously `^h` gave always a lower hat. However, this rendering is not fully compatible with PDF tagging. Now, `^h` is just `\^h`, which works with LuaLaTeX and XeTeX, and an appropriate font. If you want the former behavior, you should activate the `lowhhat` attribute, either with `\languageattribute` or as modifier (i.e., calling `esperanto.lowhhat`). However, it is only correctly tagged with LaLaTeX, which is now the recommended engine.

The position can be adjusted with something like:

```
\renewcommand\esperantolowhhatadj{\kern.04em}
```

A *positive* value shifts the hat to the *left*. You can use L^AT_EX hooks to set different values for different font styles.

Code

The macro `\LdfInit` takes care of preventing that this file is loaded more than once, checking the category code of the `@` sign, etc.

```
1 \*code
2 \LdfInit{esperanto}\captionesperanto
```

When this file is read as an option, i.e. by the `\usepackage` command, `esperanto` will be an ‘unknown’ language in which case we have to make it known. So we check for the existence of `\l@esperanto` to see whether we have to do something here.

```
3 \ifx\l@esperanto\undefined
4 \nopatterns{Esperanto}
5 \addialect\l@esperanto0\fi
```

¹The file described in this section has version number v1.5a and was last revised on 2025/07/14. A contribution was made by Ruiz-Altaba Marti (`ruizaltb@cernvm.cern.ch`). Code from the file `esperant.sty` by Jörg Knappen (`knappen@vkpmzd.kph.uni-mainz.de`) was included.

The next step consists of defining commands to switch to the Esperanto language. The reason for this is that a user might want to switch back and forth between languages.

`\captionesperanto` The macro `\captionesperanto` defines all strings used in the four standard documentclasses provided with L^AT_EX.

```

6 \addto\captionesperanto{%
7   \def\prefacename{Anta\u{u}parolo}%
8   \def\refname{Cita\^{}oj}%
9   \def\abstractname{Resumo}%
10  \def\bibName{Bibliografio}%
11  \def\chaptername{{\^{}C}apitro}%
12  \def\appendixname{Apendico}%
13  \def\contentsname{Enhavo}%
14  \def\listfigurename{Listo de figuroj}%
15  \def\listtablename{Listo de tabeloj}%
16  \def\indexname{Indekso}%
17  \def\figurename{Figuro}%
18  \def\tablename{Tabelo}%
19  \def\partname{Parto}%
20  \def\enclname{Aldono(j)}%
21  \def\ccname{Kopie al}%
22  \def\headtoname{Al}%
23  \def\pagename{Pa\^{}go}%
24  \def\subjectname{Temo}%
25  \def\seenname{vidu} % au: vd.
26  \def\alsoname{vidu anka\u{u}} % au vd. anka\u{u}
27  \def\proofname{Pruvo}%
28  \def\glossaryname{Glosaro}%
29  }

```

`\dateesperanto` The macro `\dateesperanto` redefines the command `\today` to produce Esperanto dates.

```

30 \def\dateesperanto{%
31   \def\today{\number\day{--a}~de~\ifcase\month\or
32     januaro\or februaro\or marto\or aprilo\or majo\or junio\or
33     julio\or a\u{u}gusto\or septembro\or oktobro\or novembro\or
34     decembro\fi,\space \number\year}}

```

`\extrasesperanto` The macro `\extrasesperanto` performs all the extra definitions needed for the Esperanto language. The macro `\noextrasesperanto` is used to cancel the actions of `\extrasesperanto`.

First, some definitions for `lowhhat`, used below.

```

35 \def\esperanto@hhat{\^{}h}
36 \let\esperantolowhhatadj@empty
37 \bbl@declare@ttribute{\CurrentOption}{lowhhat}{%
38   \ifodd\bbl@engine
39     \def\esperanto@hhat{h\llap{%
40       \iffontchar\font{^^^0302^^^0302\else\^{}{}}\fi\esperantolowhhatadj}}%
41   \else
42     \def\esperanto@hhat{h\llap{\^{}{}}\esperantolowhhatadj}}%
43   \fi}

```

For Esperanto the $\hat{\ }$ character is made active. This is done once, later on its definition may vary.

```
44 \initiate@active@char{^}
```

Because the character $\hat{\ }$ is used in math mode with quite a different purpose we need to add an extra level of evaluation to the definition of the active $\hat{\ }$. It checks whether math mode is active; if so the shorthand mechanism is bypassed by a direct call of `\normal@char^`.

```
45 \addto\extrasesperanto{\languageshortands{esperanto}}
```

```
46 \addto\extrasesperanto{\bbl@activate{^}}
```

```
47 \addto\noextrasesperanto{\bbl@deactivate{^}}
```

In order to prevent problems with the active $\hat{\ }$ we add a shorthand on system level which expands to a `'normal ^`.

```
48 \declare@shorthand{system}{^}{\csname normal@char\string^ \endcsname}
```

And here are the uses of the active $\hat{\ }$:

```
49 \declare@shorthand{esperanto}{^c}{\^c\allowhyphens}
```

```
50 \declare@shorthand{esperanto}{^C}{\^C\allowhyphens}
```

```
51 \declare@shorthand{esperanto}{^g}{\^g\allowhyphens}
```

```
52 \declare@shorthand{esperanto}{^G}{\^G\allowhyphens}
```

```
53 \declare@shorthand{esperanto}{^h}{\esperanto@hhat\allowhyphens}
```

```
54 \declare@shorthand{esperanto}{^H}{\^H\allowhyphens}
```

```
55 \declare@shorthand{esperanto}{^j}{\^j\allowhyphens}
```

```
56 \declare@shorthand{esperanto}{^J}{\^J\allowhyphens}
```

```
57 \declare@shorthand{esperanto}{^s}{\^s\allowhyphens}
```

```
58 \declare@shorthand{esperanto}{^S}{\^S\allowhyphens}
```

```
59 \declare@shorthand{esperanto}{^u}{\u u\allowhyphens}
```

```
60 \declare@shorthand{esperanto}{^U}{\u U\allowhyphens}
```

```
61 \declare@shorthand{esperanto}{^|}{\discretionary{-}{-}{-}\allowhyphens}
```

`\Esper` In `esperant.sty` Jörg Knappen provides the macros `\esper` and `\Esper` that can `\esper` be used instead of `\alph` and `\Alph`. These macros are available in this file as well.

Their definition takes place in two steps. First the toplevel.

```
62 \def\esper#1{\@esper{\@nameuse{c@#1}}}
```

```
63 \def\Esper#1{\@Esper{\@nameuse{c@#1}}}
```

Then the second level.

```
64 \def\@esper#1{%
```

```
65   \ifcase#1\or a\or b\or c\or ^c\or d\or e\or f\or g\or ^g\or
```

```
66   h\or \esperanto@hhat\or i\or j\or ^j\or k\or l\or m\or n\or o\or
```

```
67   p\or r\or s\or ^s\or t\or u\or \u{u}\or v\or z\else\@ctrerr\fi}
```

```
68 \def\@Esper#1{%
```

```
69   \ifcase#1\or A\or B\or C\or ^C\or D\or E\or F\or G\or ^G\or
```

```
70   H\or ^H\or I\or J\or ^J\or K\or L\or M\or N\or O\or
```

```
71   P\or R\or S\or ^S\or T\or U\or \u{U}\or V\or Z\else\@ctrerr\fi}
```

`\hodiau` In `esperant.sty` Jörg Knappen provides two alternative macros for `\today`, `\hodiaun` `\hodiau` and `\hodiaun`. The second macro produces an accusative version of the date in Esperanto.

```
72 \addto\dateesperanto{\def\hodiau{la \today}}
```

```
73 \def\hodiaun{la \number\day --an~de~\ifcase\month\or
```

```
74 januaro\or februaro\or marto\or aprilo\or majo\or junio\or
75 julio\or a\u{u}gusto\or septembro\or oktobro\or novembro\or
76 decembro\fi, \space \number\year}
```

The macro `\ldf@finish` takes care of looking for a configuration file, setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value.

```
77 \ldf@finish{esperanto}
78 \endcode}
```