

# Babel support for Polish

current maintainer: Marcin Woliński

Version 1.3, 2022/03/12

## 1 Basic Usage

Typesetting Polish with  $\text{T}_{\text{E}}\text{X}$  is, surprisingly, more difficult than setting most other European languages. Polish uses several accented characters. Some of them, including  $\acute{c}$ ,  $\acute{n}$ ,  $\acute{o}$ ,  $\acute{s}$ ,  $\acute{z}$ ,  $\acute{z}$ ,  $\acute{C}$ ,  $\acute{N}$ ,  $\acute{O}$ ,  $\acute{S}$ ,  $\acute{Z}$ ,  $\acute{Z}$ , can be reasonably rendered by attaching “floating diacritics” to Latin letters. More problematic is the letter  $ł/L$ , but legend has it that Donald Knuth needed to include the name *Lukasiewicz* in his TAOCP, so he included a “diacritic” stroke in his Computer Modern fonts. Unfortunately, he had no reason to mention *Wałęsa* anywhere in TAOCP, and in result the “ogonek<sup>1</sup> accent” is not present in original  $\text{T}_{\text{E}}\text{X}$  fonts. A different set of fonts is necessary to properly render Polish characters with ogonek:  $ą$ ,  $ę$ ,  $Ą$ ,  $Ę$  (these characters are also used in Lithuanian along with  $į$ ,  $ų$ ,  $I$ ,  $U$ ).

In Unicode  $\text{T}_{\text{E}}\text{X}$  engines (Xe $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ , lua $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ), Latin Modern fonts are used by default and these include a rich repertoire of accented Latin characters. With these engines, it’s enough to load Babel with the Polish module:

```
\usepackage[polish]{babel}
```

The matter is more complicated in the case of classical  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  (including pdf $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ). To obtain correct Polish letters, it is necessary to switch the document to a suitable font encoding. One possibility is:

```
\usepackage[OT4]{fontenc}
\usepackage[polish]{babel}
```

This will result in a document typeset using the Polish adaptation of Knuth’s classical Computer Modern fonts.

A more flexible option is the European font encoding T1, which covers most of European languages written in the Latin script. Unfortunately, the default fonts in this encoding are bitmap based, which will result in sub-optimal rendering. So with T1 it is necessary to select a different font family as well. A safe option is to

---

<sup>1</sup>The Polish name seems to also be the official English name of the diacritic: <https://en.wikipedia.org/wiki/Ogonek>.

use Latin Modern (Latin Modern is practically identical to Computer Modern, a new name was needed to keep backward compatibility of Computer Modern):

```
\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[polish]{babel}
```

Many font switching packages work with T1. Besides Latin Modern, you can expect well designed Polish characters to be present in all fonts from the T<sub>E</sub>X Gyre collection (in particular the Times lookalike `tgtermes`, Bookmanish `tgbonum`, Century Schoolbook-like `tgschola`, etc.) and in other fonts originating from the Polish T<sub>E</sub>X Users Group GUST<sup>2</sup>. With Unicode engines you can also use any OpenType font present on your system (see the `fontspec` package).

It is also worth noting, that ogonek is not really an accent: it has to be separately designed for each base letter to get a typographically acceptable result. This is not the case, e.g., for the packages in the PSNFFS set, in particular `times.sty`. Although these support the T1 encoding, the ogonek protrudes from the characters at some random angle (this is particularly visible in the case of the letter `ę`):

T <sub>E</sub> X Gyre Termes	<i>gąbke</i>	<i>gąbke</i>
PSNFF Times	<i>gąbke</i>	<i>gąbke</i>

If quality is your goal, do not use `times.sty` for Polish, use `tgtermes.sty`.

Another element necessary for successful typesetting of Polish is a way to encode Polish accented characters in the source file. Fortunately, nowadays all L<sup>A</sup>T<sub>E</sub>X engines assume input files to be in Unicode, in its UTF-8 form, so if you are on a modern system, this will work by default. (If you are forced to use some old byte-based encoding like the standard ISO 8859-2 or its Windows counterpart CP1250, please consult the documentation of the `inputenc` package.)

## 2 The Functions of this Module

As of version 1.3 the code has been updated to the new syntax provided by Babel to declare language specific elements. In its current form, the module is pretty typical `.ldf` file: it defines Polish translation of language specific strings used by L<sup>A</sup>T<sub>E</sub>X, it ensures Polish hyphenation patterns are used, and activates the setting known as `\frenchspacing`.

Language modules of Babel typically check for the hyphenation patterns for the language in question. In their absence, a warning is issued and English hyphenation patterns are activated. We find this behaviour inappropriate. Polish hyphenation patterns are a part of standard T<sub>E</sub>X distributions for over 20 years.

---

<sup>2</sup><http://www.gust.org.pl/projects/e-foundry>

Inability to load them signals some configuration problem in the system. Moreover, the use of English patterns would lead to very badly chosen breaks in a Polish text. Thus, we have decided to block automatic hyphenation in Polish text in this case. In consequence, system configuration has to be corrected or hyphenation points selected manually.

American typography has a rule, that space after punctuation characters is larger than typical inter-word space. This rule is by default followed by  $\TeX$ . In Polish tradition all spaces should be of equal width, so we activate `\frenchspacing` for Polish text.

Babel provides a mechanism for language-specific short commands (shorthands) consisting of the character " followed by one more character. Table 1 lists the commands available for Polish. The shorthand "- is similar to the command \- but the former does not block automatic hyphenation in the rest of the word. The shorthand "= should be used for explicit hyphens in the text. According to Polish rules, when a break occurs at an explicit hyphen, the hyphen gets repeated at the beginning of the new line. So, e.g., Polish compound adjectives like *biało-czerwony* ‘white-and-red’ should be input as `biało"=czerwony`.

Table 1: Shorthands provided by the Polish module

"-	soft hyphen
"=	explicit hyphen repeated after the break
"	disable ligature at this position.
"‘	Polish left double quote: „
"’	Polish right double quote: ”
"<	French left double quote: « (used in Polish as second level quotes)
">	French right double quote: »

### 3 A Note on Compatibility

Incompatible changes were introduced in version 1.3 of this module. In case of problems with processing legacy documents, please try replacing the module invocation with the following:

```
\usepackage[polish-compat]{babel}
```

This causes an exact copy of the module version v1.2l of 2005/03/31 to be loaded. However, please be aware that this old code has known problems (see Section 4.1). In compatibility mode a different list of shorthands is available, they are listed in Table 2.

Table 2: Shortcuts and commands provided in the compatibility mode. Be aware that these do not provide typographically acceptable shapes of Polish accented characters.

"a	or \aob, for a-ogonek (rendered ą)
"A	or \Aob, for A-ogonek (rendered Ą)
"e	or \eob, for e-ogonek (rendered ę)
"E	or \Eob, for E-ogonek (rendered Ę)
"c	or \'c, for accented c (ć), same with uppercase letters and n, o, s
"l	or \lpb{ }, for l with stroke (ł)
"L	or \Lpb{ }, for L with stroke (Ł)
"r	or \zkb{ }, for z-dot-accent (ż), cf. pronunciation
"R	or \Zkb{ }, for Z-dot-accent (Ż)
"z	or \'z, for accented z (ź)
"Z	or \'Z, for accented Z (Ź)
"	disable ligature at this position.
"-	an explicit hyphen sign, allowing hyphenation in the rest of the word.
" "	like "-", but producing no hyphen sign (for compound words with hyphen, e.g. x-"y).
"‘	for Polish left double quote („).
"’	for Polish right double quote (”).
"<	for French left double quote («).
">	for French right double quote (»).

## 4 The code

```

1 {*code}
2 \LdfInit{polish}\captionspolish

The new code requires at least the version 3.63 of Babel where the macro
\babel@texpdf was first introduced.
3 \@ifpackagelater{babel}{2021/08/30}{%
4 }{\PackageError{polish.ldf}{Babel too old, at least version 2021/08/30
5   required}{}%
6 \providecommand\babel@texpdf[4]{#1}%
7 }
```

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

```

8 \ifx\l@polish\undefined
9 \nopatterns{Polish}
10 \adddialect\l@polish\l@unhyphenated\fi
```

`\captionspolish` The macro `\captionspolish` defines all strings used in the four standard documentclasses provided with L<sup>A</sup>T<sub>E</sub>X.

```

11 \StartBabelCommands*{polish}{captions}
12 [unicode, charset=utf8, fontenc=TU EU1 EU2]
13 \SetString{\prefacename}{Przedmowa}
14 \SetString{\refname}{Literatura}
15 \SetString{\abstractname}{Streszczenie}
16 \SetString{\bibname}{Bibliografia}
17 \SetString{\chaptername}{Rozdział}
18 \SetString{\appendixname}{Dodatek}
19 \SetString{\contentsname}{Spis treści}
20 \SetString{\listfigurename}{Spis rysunków}
21 \SetString{\listtablename}{Spis tabel}
22 \SetString{\indexname}{Skorowidz}
23 \SetString{\figurename}{Rysunek}
24 \SetString{\tablename}{Tabela}
25 \SetString{\partname}{Część}
26 \SetString{\enclname}{Załączniki}
27 \SetString{\ccname}{Do wiadomości}
28 \SetString{\headtoname}{Do}
29 \SetString{\pagename}{Strona}
30 \SetString{\seename}{zob.}
31 \SetString{\alsoname}{zob. także}
32 \SetString{\proofname}{Dowód}
33 \SetString{\glossaryname}{Słowniczek}
34
35 \StartBabelCommands*{polish}{date}
36 [unicode, charset=utf8, fontenc=TU EU1 EU2]
37 \SetStringLoop{month#1name}{%
38     styczenia, lutego, marca, kwietnia, maja, czerwca, %
39     lipca, sierpnia, września, października, listopada, grudnia}

```

And now, the generic branch, using the LICR and assuming T1.

```

40 \StartBabelCommands*{polish}{captions}
41 \SetString{\prefacename}{Przedmowa}
42 \SetString{\refname}{Literatura}
43 \SetString{\abstractname}{Streszczenie}
44 \SetString{\bibname}{Bibliografia}
45 \SetString{\chaptername}{Rozdział}
46 \SetString{\appendixname}{Dodatek}
47 \SetString{\contentsname}{Spis treści}
48 \SetString{\listfigurename}{Spis rysunków}
49 \SetString{\listtablename}{Spis tabel}
50 \SetString{\indexname}{Skorowidz}
51 \SetString{\figurename}{Rysunek}
52 \SetString{\tablename}{Tabela}
53 \SetString{\partname}{Część}
54 \SetString{\enclname}{Załączniki}
55 \SetString{\ccname}{Do wiadomości}
56 \SetString{\headtoname}{Do}
57 \SetString{\pagename}{Strona}
58 \SetString{\seename}{zob.}

```

```

59 \SetString{\alsoname}{zob. tak\ .ze}
60 \SetString{\proofname}{Dow\ 'od}
61 \SetString{\glossaryname}{S\l owniczek}
62
63 \StartBabelCommands*{polish}{date}
64 \SetStringLoop{month#1name}{%
65     styczenia, lutego, marca, kwietnia, maja, czerwca, %
66     lipca, sierpnia, wrze\ 'snia, pa\ 'zdziernika, listopada, grudnia}
67 \SetString\today{%
68     {\number\day}~\@nameuse{month\romannumeral\month name} {\number\year}}
69 \EndBabelCommands

```

`\extrapolish` The macro `\extrapolish` will perform all the extra definitions needed for the Polish language. The macro `\noextrapolish` is used to cancel the actions of `\extrapolish`.

```

70 \initiate@active@char{"}
71 \declare@shorthand{polish}{"-}{\bbl@hy@soft}
72 \declare@shorthand{polish}{"="}{\babel@texpdf{\bbl@hy@repeat}{-}{-}{\textminus}}
73 \declare@shorthand{polish}{"|}{\babel@texpdf{\discretionary{-}{-}{\kern.03em}}{-}{-}}
74 \declare@shorthand{polish}{"'}{\quotedblbase}
75 \declare@shorthand{polish}{"'}{\textquotedblright}
76 \declare@shorthand{polish}{"<}{\guillemotleft}
77 \declare@shorthand{polish}{">}{\guillemotright}

```

We specify that the polish group of shorthands should be used. These characters are ‘turned on’ once, later their definition may vary.

```

78 \addto\extrapolish{%
79     \bbl@frenchspacing
80     \languageshorthands{polish}%
81     \bbl@activate{}}

```

For Polish texts `\frenchspacing` should be in effect. We make sure this is the case and reset it if necessary.

```

82 \addto\noextrapolish{%
83     \bbl@nonfrenchspacing
84     \bbl@deactivate{}}

```

```

85 \ldf@finish{polish}
86 </code>

```

## 4.1 Compatibility code

This section contains the code and documentation of version v1.2l (2005/03/31) of `polish.ldf`, which is used for `polish-compat`. We keep it as a compatibility fallback, since (more or less) this version was available for almost 20 years and some documents may depend on its idiosyncratic features. There are known problems with this code:

- The "-shorthands for Polish characters do not use LICR, so even with modern fonts the precomposed characters from the font are not used (in the case of  $q$ ,  $\epsilon$ ,  $l$ ,  $z$ ).
- The substitute for ogonek accent defined for OT1 is not accessible with standard `\k` (besides being hideous).
- Wrong shape of  $\epsilon$  is used in the word *Część* generated by the command `\part`.
- The code is **evil**: it redefines `\selectfont`, which leads to conflicts with some standard L<sup>A</sup>T<sub>E</sub>X packages. Another known incompatibility is the macro `\lll`, which conflicts with AMS-L<sup>A</sup>T<sub>E</sub>X.

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

```
87 *compat
88 \LdfInit{polish}\captionspolish
```

This file is part of the babel system, it provides the source code for the Polish language definition file. It was developed out of Polish.tex, which was written by Elmar Schalueck and Michael Janich. Polish.tex was based on code by Leszek Holenderski, Jerzy Ryll and J. S. Bień from Faculty of Mathematics, Informatics and Mechanics of Warsaw University, except of Jerzy Ryll (Instytut Informatyki Uniwersytetu Warszawskiego).

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

```
89 \ifx\l@polish\@undefined
90 \@nopatterns{Polish}
91 \adddialect\l@polish0\fi
```

The next step consists of defining commands to switch to (and from) the Polish language.

`\captionspolish` The macro `\captionspolish` defines all strings used in the four standard documentclasses provided with L<sup>A</sup>T<sub>E</sub>X.

```
92 \addto\captionspolish{%
93 \def\prefacename{Przedmowa}%
94 \def\refname{Literatura}%
95 \def\abstractname{Streszczenie}%
96 \def\bibname{Bibliografia}%
97 \def\chaptername{Rozdział}%
98 \def\appendixname{Dodatek}%
99 \def\contentsname{Spis treści}%
100 \def\listfigurename{Spis rysunków}%
101 \def\listtablename{Spis tabel}%
102 \def\indexname{Indeks}%
103 \def\figurename{Rysunek}%
```

```

104 \def\tablename{Tablica}%
105 \def\partname{Cz\aeob{}}\s{c}%
106 \def\enclname{Za\l\ao{b}{c}znik}%
107 \def\ccname{Kopie:}%
108 \def\headtoname{Do}%
109 \def\pagename{Strona}%
110 \def\seename{Por\ownaj}%
111 \def\alsoname{Por\ownaj tak\ze}%
112 \def\proofname{Dow\od}%
113 \def\glossaryname{Glossary}% <-- Needs translation
114 }

```

`\datepolish` The macro `\datepolish` redefines the command `\today` to produce Polish dates.

```

115 \def\datepolish{%
116 \def\today{\number\day~\ifcase\month\or
117 stycznia\or lutego\or marca\or kwietnia\or maja\or czerwca\or lipca\or
118 sierpnia\or wrze\snia\or pa\zdziernika\or listopada\or grudnia\fi
119 \space\number\year}%
120 }

```

`\extrapolish` The macro `\extrapolish` will perform all the extra definitions needed for the Polish language. The macro `\noextrapolish` is used to cancel the actions of `\extrapolish`.

For Polish the " character is made active. This is done once, later on its definition may vary. Other languages in the same document may also use the " character for shorthands; we specify that the polish group of shorthands should be used.

```

121 \initiate@active@char{"}
122 \addto\extrapolish{\languageshorthands{polish}}
123 \addto\extrapolish{\bbl@activate{}}

```

Don't forget to turn the shorthands off again.

```

124 \addto\noextrapolish{\bbl@deactivate{}}

```

The code above is necessary because we need an extra active character. This character is then used as indicated in table 2.

If you have problems at the end of a word with a linebreak, use the other version without hyphenation tricks. Some TeX wizard may produce a better solution with forecasting another token to decide whether the character after the double quote is the last in a word. Do it and let us know.

In Polish texts some letters get special diacritical marks. Leszek Holenderski designed the following code to position the diacritics correctly for every font in every size. These macros need a few extra dimension variables.

```

125 \newdimen\pl@left
126 \newdimen\pl@down
127 \newdimen\pl@right
128 \newdimen\pl@temp

```



`\sob` The macro `\sob` is used to put the ‘ogonek’ in the right place.

```
129 \def\sob#1#2#3#4#5{%parameters: letter and fractions hl,ho,vl,vo
130 \setbox0\hbox{#1}\setbox1\hbox{$_\mathchar'454$}\setbox2\hbox{p}%
131 \pl@right=#2\wd0 \advance\pl@right by-#3\wd1
132 \pl@down=#5\ht1 \advance\pl@down by-#4\ht0
133 \pl@left=\pl@right \advance\pl@left by\wd1
134 \pl@temp=-\pl@down \advance\pl@temp by\dp2 \dp1=\pl@temp
135 \leavevmode
136 \kern\pl@right\lower\pl@down\box1\kern-\pl@left #1}
```

`\aob` The ogonek is placed with the letters ‘a’, ‘A’, ‘e’, and ‘E’.

```
\Aob 137 \DeclareTextCommand{\aob}{OT1}{\sob a{.66}{.20}{0}{.90}}
\Eob 138 \DeclareTextCommand{\Aob}{OT1}{\sob A{.80}{.50}{0}{.90}}
\Eob 139 \DeclareTextCommand{\eob}{OT1}{\sob e{.50}{.35}{0}{.93}}
140 \DeclareTextCommand{\Eob}{OT1}{\sob E{.60}{.35}{0}{.90}}
```

For the ‘new’ T1 encoding we can provide simpler definitions.

```
141 \DeclareTextCommand{\aob}{T1}{\k a}
142 \DeclareTextCommand{\Aob}{T1}{\k A}
143 \DeclareTextCommand{\eob}{T1}{\k e}
144 \DeclareTextCommand{\Eob}{T1}{\k E}
```

Construct the characters by default from the OT1 encoding.

```
145 \ProvideTextCommandDefault{\aob}{\UseTextSymbol{OT1}{\aob}}
146 \ProvideTextCommandDefault{\Aob}{\UseTextSymbol{OT1}{\Aob}}
147 \ProvideTextCommandDefault{\eob}{\UseTextSymbol{OT1}{\eob}}
148 \ProvideTextCommandDefault{\Eob}{\UseTextSymbol{OT1}{\Eob}}
```

`\spb` The macro `\spb` is used to put the ‘poprzeczka’ in the right place.

```
149 \def\spb#1#2#3#4#5{%
150 \setbox0\hbox{#1}\setbox1\hbox{\char'023}%
151 \pl@right=#2\wd0 \advance\pl@right by-#3\wd1
152 \pl@down=#5\ht1 \advance\pl@down by-#4\ht0
153 \pl@left=\pl@right \advance\pl@left by\wd1
154 \ht1=\pl@down \dp1=-\pl@down
155 \leavevmode
156 \kern\pl@right\lower\pl@down\box1\kern-\pl@left #1}
```

`\skb` The macro `\skb` is used to put the ‘kropka’ in the right place.

```
157 \def\skb#1#2#3#4#5{%
158 \setbox0\hbox{#1}\setbox1\hbox{\char'056}%
159 \pl@right=#2\wd0 \advance\pl@right by-#3\wd1
160 \pl@down=#5\ht1 \advance\pl@down by-#4\ht0
161 \pl@left=\pl@right \advance\pl@left by\wd1
162 \leavevmode
163 \kern\pl@right\lower\pl@down\box1\kern-\pl@left #1}
```

`\textpl` For the ‘poprzeczka’ and the ‘kropka’ in text fonts we don’t need any special coding, but we can (almost) use what is already available.

```

164 \def\textpl{%
165   \def\lpb{\p111}%
166   \def\Lpb{\pLLL}%
167   \def\zkb{\.z}%
168   \def\Zkb{\.Z}}

```

Initially we assume that typesetting is done with text fonts.

```

169 \textpl
170 \let\111=\1 \let\LLL=\L
171 \def\p111{\111}
172 \def\pLLL{\LLL}

```

`\telepl` But for the ‘teletype’ font in ‘OT1’ encoding we have to take some special actions, involving the macros defined above.

```

173 \def\telepl{%
174   \def\lpb{\spb 1{.45}{.5}{.4}{.8}}%
175   \def\Lpb{\spb L{.23}{.5}{.4}{.8}}%
176   \def\zkb{\skb z{.5}{.5}{1.2}{0}}%
177   \def\Zkb{\skb Z{.5}{.5}{1.1}{0}}

```

To activate these codes the font changing commands as they are defined in  $\text{\LaTeX}$  are modified. The same is done for plain  $\text{\TeX}$ ’s font changing commands.

When `\selectfont` is undefined the current format is supposed to be either plain (based) or  $\text{\LaTeX}$  2.09.

```

178 \ifx\selectfont\undefined
179   \ifx\prm\undefined \addto\rm{\textpl}\else \addto\prm{\textpl}\fi
180   \ifx\pit\undefined \addto\it{\textpl}\else \addto\pit{\textpl}\fi
181   \ifx\pbf\undefined \addto\bf{\textpl}\else \addto\pbf{\textpl}\fi
182   \ifx\psl\undefined \addto\sl{\textpl}\else \addto\psl{\textpl}\fi
183   \ifx\psf\undefined \else \addto\psf{\textpl}\fi
184   \ifx\psc\undefined \else \addto\psc{\textpl}\fi
185   \ifx\ptt\undefined \addto\tt{\telepl}\else \addto\ptt{\telepl}\fi
186 \else

```

When `\selectfont` exists we assume  $\text{\LaTeX}$  2 $\epsilon$ .

```

187 \expandafter\addto\csname selectfont \endcsname{%
188   \csname\f@encoding @pl\endcsname}
189 \fi

```

Currently we support the OT1 and T1 encodings. For T1 we don’t have to make a difference between typewriter fonts and other fonts, they all have the same glyphs.

```

190 \expandafter\let\csname T1@pl\endcsname\textpl

```

For OT1 we need to check the current font family, stored in `\f@family`. Unfortunately we need a hack as `\ttdefault` is defined as a `\long` macro, while `\f@family` is not.

```

191 \expandafter\def\csname OT1@pl\endcsname{%
192   \long\edef\curr@family{\f@family}%
193   \ifx\curr@family\ttdefault
194     \telepl

```

```

195 \else
196 \textpl
197 \fi}

```

`\dq` We save the original double quote character in `\dq` to keep it available, the math accent `\'` can now be typed as `"`.

```

198 \begingroup \catcode'\''12
199 \def\x{\endgroup
200 \def\dq{"}}
201 \x

```

Now we can define the doublequote macros for diacritics,

```

202 \declare@shorthand{polish}{"a}{\textormath{\aob}{\ddot a}}
203 \declare@shorthand{polish}{"A}{\textormath{\Aob}{\ddot A}}
204 \declare@shorthand{polish}{"c}{\textormath{\'c}{\acute c}}
205 \declare@shorthand{polish}{"C}{\textormath{\'C}{\acute C}}
206 \declare@shorthand{polish}{"e}{\textormath{\eob}{\ddot e}}
207 \declare@shorthand{polish}{"E}{\textormath{\Eob}{\ddot E}}
208 \declare@shorthand{polish}{"l}{\textormath{\lpb}{\ddot l}}
209 \declare@shorthand{polish}{"L}{\textormath{\Lpb}{\ddot L}}
210 \declare@shorthand{polish}{"n}{\textormath{\'n}{\acute n}}
211 \declare@shorthand{polish}{"N}{\textormath{\'N}{\acute N}}
212 \declare@shorthand{polish}{"o}{\textormath{\'o}{\acute o}}
213 \declare@shorthand{polish}{"O}{\textormath{\'O}{\acute O}}
214 \declare@shorthand{polish}{"s}{\textormath{\'s}{\acute s}}
215 \declare@shorthand{polish}{"S}{\textormath{\'S}{\acute S}}

```

`\polishrz` The command `\polishrz` defines the shorthands `"r`, `"z` and `"x` to produce pointed `z`, accented `z` and `"x`. This is the default as these shorthands were defined by this language definition file for quite some time.

```

216 \newcommand*{\polishrz}{%
217 \declare@shorthand{polish}{"r}{\textormath{\zkb}{\ddot r}}%
218 \declare@shorthand{polish}{"R}{\textormath{\Zkb}{\ddot R}}%
219 \declare@shorthand{polish}{"z}{\textormath{\'z}{\acute z}}%
220 \declare@shorthand{polish}{"Z}{\textormath{\'Z}{\acute Z}}%
221 \declare@shorthand{polish}{"x}{\dq x}%
222 \declare@shorthand{polish}{"X}{\dq X}%
223 }
224 \polishrz

```

The command `\polishzx` switches to a different set of shorthands, `"z`, `"x` and `"r` to produce pointed `z`, accented `z` and `"r`; a different shorthand notation also in use.

```

225 \newcommand*{\polishzx}{%
226 \declare@shorthand{polish}{"z}{\textormath{\zkb}{\ddot z}}%
227 \declare@shorthand{polish}{"Z}{\textormath{\Zkb}{\ddot Z}}%
228 \declare@shorthand{polish}{"x}{\textormath{\'z}{\acute x}}%
229 \declare@shorthand{polish}{"X}{\textormath{\'Z}{\acute X}}%
230 \declare@shorthand{polish}{"r}{\dq r}%

```

```

231 \declare@shorthand{polish}{"R}{\dq R}%
232 }

```

Then we define access to two forms of quotation marks, similar to the german and french quotation marks.

```

233 \declare@shorthand{polish}{"‘}{%
234 \textormath{\quotedblbase}{\mbox{\quotedblbase}}}
235 \declare@shorthand{polish}{"’}{%
236 \textormath{\textquotedblright}{\mbox{\textquotedblright}}}
237 \declare@shorthand{polish}{"<}{%
238 \textormath{\guillemotleft}{\mbox{\guillemotleft}}}
239 \declare@shorthand{polish}{">}{%
240 \textormath{\guillemotright}{\mbox{\guillemotright}}}

```

then we define two shorthands to be able to specify hyphenation breakpoints that behave a little different from \-.

```

241 \declare@shorthand{polish}{"-}{\nobreak-\bbl@allowhyphens}
242 \declare@shorthand{polish}{"~}{\hskip\z@skip}

```

And we want to have a shorthand for disabling a ligature.

```

243 \declare@shorthand{polish}{"|}{%
244 \textormath{\discretionary{-}{-}{\kern.03em}}{}}

```

`\mdqon` All that’s left to do now is to define a couple of commands for reasons of compatibility with `polish.tex`.

```

245 \def\mdqon{\shorthandon{}}
246 \def\mdqoff{\shorthandoff{}}

```

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.

```

247 \ldf@finish{polish}
248 </compat>

```