

1 The Mongolian language

The file `mongolian.dtx`¹ defines all the language-specific macros for the Mongolian language. It needs the file `cyrkod` for success documentation with Mongolian encodings (see below).

For this language the character " is made active. In table 1 an overview is given of its purpose.

"	disable ligature at this position.
"-	an explicit hyphen sign, allowing hyphenation in the rest of the word.
"---	Cyrillic emdash in plain text.
"--~	Cyrillic emdash in compound names (surnames).
"--*	Cyrillic emdash for denoting direct speech.
""	like "-", but producing no hyphen sign (for compound words with hyphen, e.g. x-"y or some other signs as "disable/enable").
"~	for a compound word mark without a breakpoint.
"=	for a compound word mark with a breakpoint, allowing hyphenation in the composing words.
" ,	thinspace for initials with a breakpoint in following surname.
"‘	for German left double quotes (looks like „).
"’	for German right double quotes (looks like “).
"<<	for French left double quotes (looks like <<).
">>	for French right double quotes (looks like >>).

Table 1: The extra definitions made by `mongolian`

The quotes in table 1 (see, also table ??) can also be typeset by using the commands in table 2 (see, also table ??).

<code>\cdash---</code>	Cyrillic emdash in plain text.
<code>\cdash--~</code>	Cyrillic emdash in compound names (surnames).
<code>\cdash--*</code>	Cyrillic emdash for denoting direct speech.
<code>\glqq</code>	for German left double quotes (looks like „).
<code>\grqq</code>	for German right double quotes (looks like “).
<code>" \flqq</code>	for French left double quotes (looks like <<).
<code>\frqq</code>	for French right double quotes (looks like >>).
<code>\dq</code>	the original quotes character (").

Table 2: More commands which produce quotes, defined by `babel`

The French quotes are also available as ligatures ‘<<’ and ‘>>’ in 8-bit Cyrillic font encodings (LCY, X2, T2*) and as ‘<’ and ‘>’ characters in 7-bit Cyrillic font encodings (OT2 and LWN).

¹The file described in this section has version number v1.2. This file was derived from the `russianb.dtx` version 1.1r.

The quotation marks traditionally used in Mongolian and Russian languages were borrowed from other languages (e.g. French and German) so they keep their original names.

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{mongolian}{captionsmongolian}
```

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

```
3 \ifx\l@mongolian\@undefined
4   \nopatterns{Mongolian}
5   \adddialect\l@mongolian0
6 \fi
```

`\latinencoding` We need to know the encoding for text that is supposed to be which is active at the end of the `babel` package. If the `fontenc` package is loaded later, then... too bad!

```
7 \let\latinencoding\cf@encoding
```

The user may choose between different available Cyrillic encodings—e.g., `X2`, `LCY`, or `LWN`. Hopefully, `X2` will eventually replace the two latter encodings (`LCY` and `LWN`). If the user wants to use another font encoding than the default (`T2A`), he has to load the corresponding file *before* `mongolian.sty`. This may be done in the following way:

```
% override the default X2 encoding used in Babel
\usepackage[cpctt,OT1]{fontenc}
\usepackage[english,mongolian]{babel}
```

Note: for the Mongolian language, the `T2A` encoding is better than `X2`, because `X2` does not contain Latin letters, and users should be very careful to switch the language every time they want to typeset a Latin word inside a Mongolian phrase or vice versa.

We parse the `\cdp@list` containing the encodings known to `LATEX` in the order they were loaded. We set the `\cyrillicencoding` to the *last* loaded encoding in the list of supported Cyrillic encodings: `OT2`, `X2`, `T2B`, `T2A`, `CTT`, `MNK`, `MLS`, if any.

```
8 \def\reserved@a#1#2{%
9   \edef\reserved@b{#1}%
10  \edef\reserved@c{#2}%
11  \ifx\reserved@b\reserved@c
12    \let\cyrillicencoding\reserved@c
13  \fi}
14 \def\cdp@elt#1#2#3#4{%
15   \reserved@a{#1}{X2}%
16   \reserved@a{#1}{T2B}%
17   \reserved@a{#1}{T2A}}
18 \cdp@list
```

Now, if `\cyrillicencoding` is undefined, then the user did not load any of supported encodings. So, we have to set `\cyrillicencoding` to some default value. We test the presence of the encoding definition files in the order from less preferable to more preferable encodings. We use the lowercase names (i.e., `lcyenc.def` instead of `LCYenc.def`).

```
19 \ifx\cyrillicencoding\undefined
20 \IfFileExists{x2enc.def}{\def\cyrillicencoding{X2}}\relax
21 \IfFileExists{t2benc.def}{\def\cyrillicencoding{T2B}}\relax
22 \IfFileExists{t2aenc.def}{\def\cyrillicencoding{T2A}}\relax
```

If `\cyrillicencoding` is still undefined, then the user seems not to have a properly installed distribution. A fatal error.

```
23 \ifx\cyrillicencoding\undefined
24 \PackageError{babel}%
25 {No Cyrillic encoding definition files were found}%
26 {Your installation is incomplete.\MessageBreak
27 You need at least one of the following files:\MessageBreak
28 \space\space
29 x2enc.def, t2aenc.def, t2benc.def, x2enc.def.}%
30 \else
```

We avoid `\usepackage[\cyrillicencoding]{fontenc}` because we don't want to force the switch of `\encodingdefault`.

```
31 \lowercase
32 \expandafter{\expandafter\input\cyrillicencoding enc.def\relax}%
33 \fi
34 \fi
```

```
\PackageInfo{babel}
{Using '\cyrillicencoding' as a default Cyrillic encoding}%
```

```
35 \DeclareRobustCommand{\Mongolian}{%
36 \fontencoding\cyrillicencoding\selectfont
37 \let\encodingdefault\cyrillicencoding
38 \expandafter\set@hyphenmins\mongolianhyphenmins
39 \language\l@mongolian}%
40 \DeclareRobustCommand{\English}{%
41 \fontencoding\latinencoding\selectfont
42 \let\encodingdefault\latinencoding
43 \expandafter\set@hyphenmins\englishhyphenmins
44 \language\l@english}%
45 \let\Mon\Mongolian
46 \let\Eng\English
47 \let\cyrillictext\Mongolian
48 \let\cyr\Mongolian
```

Since the X2 encoding does not contain Latin letters, we should make some redefinitions of L^AT_EX macros which implicitly produce Latin letters.

```
49 \expandafter\ifx\csname T@X2\endcsname\relax\else
```

We put `\latinencoding` in braces to avoid problems with `\@alph` inside minipages (e.g., footnotes inside minipages) where `\@alph` is expanded and we get for example `'\fontencoding OT1'` (`\fontencoding` is robust).

```

50 \def\@alph#1{\fontencoding{\latinencoding}\selectfont
51 \ifcase#1\or
52   a\or b\or c\or d\or e\or f\or g\or h\or
53   i\or j\or k\or l\or m\or n\or o\or p\or
54   q\or r\or s\or t\or u\or v\or w\or x\or
55   y\or z\else\@ctrerr\fi}}%
56 \def\@Alph#1{\fontencoding{\latinencoding}\selectfont
57 \ifcase#1\or
58   A\or B\or C\or D\or E\or F\or G\or H\or
59   I\or J\or K\or L\or M\or N\or O\or P\or
60   Q\or R\or S\or T\or U\or V\or W\or X\or
61   Y\or Z\else\@ctrerr\fi}}%

```

Unfortunately, the commands `\AA` and `\aa` are not encoding dependent in \LaTeX (unlike e.g., `\oe` or `\DH`). They are defined as `\r{A}` and `\r{a}`. This leads to unpredictable results when the font encoding does not contain the Latin letters ‘A’ and ‘a’ (like X2).

```

62 \DeclareTextSymbolDefault{\AA}{OT1}
63 \DeclareTextSymbolDefault{\aa}{OT1}
64 \DeclareTextCommand{\aa}{OT1}{\r a}
65 \DeclareTextCommand{\AA}{OT1}{\r A}
66 \fi

```

The following block redefines the character class of uppercase Greek letters and some accents, if it is equal to 7 (variable family), to avoid incorrect results if the font encoding in some math family does not contain these characters in places of OT1 encoding. The code was taken from `amsmath.dtx`. See comments and further explanation there.

```

67 % \begingroup\catcode'\="=12
68 % % uppercase greek letters:
69 % \def\@tempa#1{\expandafter\@tempb\meaning#1\relax\relax\relax\relax
70 %   "0000\@nil#1}
71 % \def\@tempb#1"#2#3#4#5#6\@nil#7{%
72 %   \ifnum"#2=7 \count@"1#3#4#5\relax
73 %     \ifnum\count@<"1000 \else \global\mathchardef#7="0#3#4#5\relax \fi
74 %   \fi}
75 % \@tempa\Gamma\@tempa\Delta\@tempa\Theta\@tempa\Lambda\@tempa\Xi
76 % \@tempa\Pi\@tempa\Sigma\@tempa\Upsilon\@tempa\Phi\@tempa\Psi
77 % \@tempa\Omega
78 % % some accents:
79 % \def\@tempa#1#2\@nil{\def\@tempc{#1}}\def\@tempb{\mathaccent}
80 % \expandafter\@tempa\hat\relax\relax\@nil
81 % \ifx\@tempb\@tempc
82 %   \def\@tempa#1\@nil{#1}%
83 %   \def\@tempb#1{\afterassignment\@tempa\mathchardef\@tempc=}
84 %   \def\do#1"#2{}
85 %   \def\@tempd#1{\expandafter\@tempb#1\@nil
86 %     \ifnum\@tempc>"FFF
87 %       \xdef#1{\mathaccent"\expandafter\do\meaning\@tempc\space}%
88 %     \fi}
89 % \@tempd\hat\@tempd\check\@tempd\tilde\@tempd\acute\@tempd\grave
90 % \@tempd\dot\@tempd\ddot\@tempd\breve\@tempd\bar
91 % \fi
92 % \endgroup

```

The user must use the `inputenc` package when any 8-bit Cyrillic font encoding is used, selecting one of the Cyrillic input encodings. We do not assume any default input encoding, so the user should explicitly call the `inputenc` package by `\usepackage{inputenc}`. We also removed `\AtBeginDocument`, so `inputenc` should be used before `babel`.

```

93 \@ifpackageloaded{inputenc}{}{}%
94 \def\reserved@a{LWN}%
95 \ifx\reserved@a\cyrillicencoding\else
96 \def\reserved@a{OT2}%
97 \ifx\reserved@a\cyrillicencoding\else
98 \PackageWarning{babel}%
99 {No input encoding specified for Mongolian language}
100 \fi\fi}

```

Now we define two commands that offer the possibility to switch between Cyrillic and Roman encodings.

`\cyrillictext` The command `\cyrillictext` will switch from Latin font encoding to the Cyrillic font encoding, the command `\latintext` switches back. This assumes that the ‘normal’ font encoding is a Latin one. These commands are *declarations*, for shorter peaces of text the commands `\textlatin` and `\textcyrillic` can be used.

```

101 %\DeclareRobustCommand{\latintext}{}%
102 % \fontencoding{\latinencoding}\selectfont
103 % \def\encodingdefault{\latinencoding}}
104 \let\lat\latintext

```

`\textcyrillic` These commands take an argument which is then typeset using the requested font encoding.
`\textlatin`

```

105 \DeclareTextFontCommand{\textcyrillic}{\cyrillictext}
106 %\DeclareTextFontCommand{\textlatin}{\latintext}

```

We make the \TeX

```

107 %\ifx\ltxTeX\undefined\let\ltxTeX\TeX\fi
108 %\ProvideTextCommandDefault{\TeX}{\textlatin{\ltxTeX}}

```

and \LaTeX logos encoding independent.

```

109 %\ifx\ltxLaTeX\undefined\let\ltxLaTeX\LaTeX\fi
110 %\ProvideTextCommandDefault{\LaTeX}{\textlatin{\ltxLaTeX}}

```

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

`\captionsmongolian` The macro `\captionsmongolian` defines all strings used in the four standard document classes provided with \LaTeX . The two commands `\cyr` and `\lat` activate Cyrillic resp. Latin encoding.

```

111 \addto\captionsmongolian{%
112 \def\prefacename{\cyr\CYROTLD\cyr\cyrn\cyrotld\cyrh
113 \ \cyr\cyrg}}%
114 \def\refname{%
115 {\cyr\CYRA\cyrsh\cyri\cyrg\cyr1\cyr\cyr\cyr\cyr\cyrn
116 \ \cyrn\cyro\cyr}}%
117 \def\abstractname{\cyr\CYRU\cyrd\cyri\cyrr\cyrt\cyrg\cyr\cyr1}}%

```

```

118 \def\bibName{{\cyr\CYRN\cyro\cyr\cyrm\cyrz\cyry\cyrishrt}}%
119 \def\chaptername{{\cyr\CYRB\cyry\cyrl\cyrerev\cyrg}}%
120 \def\appendixname{{\cyr\CYRH\cyra\cyrv\cyrs\cyrr\cyra\cyrl\cyrt}}%
121 \@ifundefined{thechapter}%
122   {\def\contentsname{{\cyr\CYRA\cyrg\cyru\cyru\cyrl\cyrg\cyra}}}%
123   {\def\contentsname{{\cyr\CYRG\cyra\cyrr\cyrch\cyri\cyrg}}}%
124 \def\listfigurename{{\cyr\CYZ\cyru\cyrr\cyrg\cyri\cyrishrt\cyrn
125   \ \cyrzh\cyra\cyrg\cyrs\cyra\cyra\cyrl\cyrt}}%
126 \def\listtablename{%
127   {\cyr\CYRH\cyry\cyrs\cyrn\cyrerev\cyrg\cyrt\cyri\cyrishrt\cyrn
128   \ \cyrzh\cyra\cyrg\cyrs\cyra\cyra\cyrl\cyrt}}%
129 \def\indexname{{\cyr\CYRT\cyro\cyrv\cyrhdsn\cyro\cyrg}}%
130 \def\authorname{{\cyr\CYRN\cyrerev\cyrr\cyri\cyrishrt\cyrn
131   \ \cyrh\cyrerev\cyrl\cyrh\cyrerev\cyrerev}}%
132 \def\figurename{{\cyr\CYZ\cyru\cyrr\cyra\cyrg}}%
133 \def\tablename{{\cyr\CYRH\cyry\cyrs\cyrn\cyrerev\cyrg\cyrt}}%
134 \def\partname{{\cyr\CYRH\cyrerev\cyrs\cyrerev\cyrg}}%
135 \def\enclname{{\cyr\CYRI\cyrsh\cyrl\cyrerev\cyrl}}%
136 \def\ccname{{\cyr\cyrerev.\cyrs.}}%
137 \def\headtoname{}%
138 \def\pagename{{\cyr\cyrt\cyra\cyrl}}%
139 \def\seename{{\cyr\cyrt\cyra\cyrl\cyrd\ \cyry\cyrz}}%
140 \def\alsiname{}%
141 {\cyr\cyr\cyrotld\cyrn\ \cyrt\cyra\cyrl\cyrd\ \cyry\cyrz}}%
142 \def\proofname{{\cyr\CYRB\cyra\cyrt\cyra\cyrl\cyrg\cyra\cyra}}%
143 \def\glossaryname{{\cyr\CYRT\cyra\cyrishrt\cyrl\cyrb\cyra\cyrr}}%
144 }

```

`\datemongolian` The macro `\datemongolian` redefines the command `\today` to produce Mongolian dates.

```

145 \def\datemongolian{%
146   \def\today{\number\year~\cyro\cyrn\cyrery\ \ifcase\month\or
147   1-\cyrr\or 2-\cyrr\or 3-\cyrr\or 4-\cyrr\or 5-\cyrr\or 6-\cyrr\or
148   7-\cyrr\or 8-\cyrr\or 9-\cyrr\or 10-\cyrr\or 11-\cyrr\or 12-\cyrr\fi
149   ~\cyrs\cyra\cyrr\cyrery\cyrn\ \number\day}}

```

`\extrasmongolian` The macro `\extrasmongolian` will perform all the extra definitions needed for the Mongolian language. The macro `\noextrasmongolian` is used to cancel the actions of `\extrasmongolian`.

The first action we define is to switch on the selected Cyrillic encoding whenever we enter ‘mongolian’.

```

150 \addto\extrasmongolian{\cyrillictext}

```

When the encoding definition file was processed by \LaTeX the current font encoding is stored in `\latinencoding`, assuming that \LaTeX uses T1 or OT1 as default. Therefore we switch back to `\latinencoding` whenever the Mongolian language is no longer ‘active’.

```

151 \addto\noextrasmongolian{\latintext}

```

Next we must allow hyphenation in the Mongolian words with apostrophe whenever we enter ‘mongolian’. This solution was proposed by Vladimir Volovich <vvv@vvv.vsu.ru>

```

152 \addto\extrasmongolian{\lccode'\=''}
153 \addto\noextrasmongolian{\lccode'\=0}

```

`\verbatim@font` In order to get both Latin and Cyrillic letters in verbatim text we need to change the definition of an internal L^AT_EX command somewhat:

```
154 %\def\verbatim@font{%
155 % \let\encodingdefault\latinencoding
156 % \normalfont\ttfamily
157 % \expandafter\def\csname\cyrillicencoding-cmd\endcsname##1##2{%
158 % \ifx\protect\@typeset@protect
159 % \begingroup\UseTextSymbol\cyrillicencoding##1\endgroup
160 % \else\noexpand##1\fi}}
```

The category code of the characters ‘:’, ‘;’, ‘!’, and ‘?’ is made `\active` to insert a little white space.

For Mongolian (as well as for Russian and German) the " character also is made active.

Note: It is *very* questionable whether the Russian typesetting tradition requires additional spacing before those punctuation signs. Therefore, we make the corresponding code optional. If you need it, then define the `frenchpunct` docstrip option in `babel.ins`.

Borrowed from french. Some users dislike automatic insertion of a space before ‘double punctuation’, and prefer to decide themselves whether a space should be added or not; so a hook `\NoAutoSpaceBeforeFDP` is provided: if this command is added (in file `mongolian.cfg`, or anywhere in a document) `mongolian` will respect your typing, and introduce a suitable space before ‘double punctuation’ *if and only if* a space is typed in the source file before those signs.

The command `\AutoSpaceBeforeFDP` switches back to the default behavior of `mongolian`.

```
161 <frenchpunct>
162 \initiate@active@char{:}
163 \initiate@active@char{;}
164 </frenchpunct>
165 <frenchpunct | spanishlig>
166 \initiate@active@char{!}
167 \initiate@active@char{?}
168 </frenchpunct | spanishlig>
169 \initiate@active@char{"}
```

The code above is necessary because we need extra active characters. The character " is used as indicated in table 1.

We specify that the Mongolian group of shorthands should be used.

```
170 \addto\extrasmongolian{\languageshorthands{mongolian}}
```

These characters are ‘turned on’ once, later their definition may vary.

```
171 \addto\extrasmongolian{%
172 <frenchpunct> \bbl@activate{:}\bbl@activate{;}%
173 <frenchpunct | spanishlig> \bbl@activate{!}\bbl@activate{?}%
174 \bbl@activate{"}
175 \addto\noextrasmongolian{%
176 <frenchpunct> \bbl@deactivate{:}\bbl@deactivate{;}%
177 <frenchpunct | spanishlig> \bbl@deactivate{!}\bbl@deactivate{?}%
178 \bbl@deactivate{"}}
```

The X2 and T2* encodings do not contain `spanish_shriek` and `spanish_query` symbols; as a consequence, the ligatures ‘?’ and ‘!’ do not work with them (these characters are useless for Cyrillic texts anyway). But we define the shorthands to emulate these ligatures (optionally).

We do not use `\latinencoding` here (but instead explicitly use OT1) because the user may choose T2A to be the primary encoding, but it does not contain these characters.

```
179 <*spanishligs>
180 \declare@shorthand{mongolian}{?'}{\UseTextSymbol{OT1}\textquestiondown}
181 \declare@shorthand{mongolian}{!''}{\UseTextSymbol{OT1}\textexclamdown}
182 </spanishligs>
```

`\mongolian@sh@;` We have to reduce the amount of white space before `;`, `:` and `!`. This should only happen in horizontal mode, hence the test with `\ifhmode`.

```
\mongolian@sh@:
183 <*frenchpunct>
184 \declare@shorthand{mongolian}{:}{;%
185 \ifhmode
```

In horizontal mode we check for the presence of a ‘space’, ‘unskip’ if it exists and place a 0.1em kerning.

```
186 \ifdim\lastskip>\z@
187 \unskip\nobreak\kern.1em
188 \else
```

If no space has been typed, we add `\FDP@thinspace` which will be defined, up to the user’s wishes, as an automatic added thinspace, or as `\@empty`.

```
189 \FDP@thinspace
190 \fi
191 \fi
```

Now we can insert a ‘;’ character.

```
192 \string;}
```

The other definitions are very similar.

```
193 \declare@shorthand{mongolian}{:}{;%
194 \ifhmode
195 \ifdim\lastskip>\z@
196 \unskip\nobreak\kern.1em
197 \else
198 \FDP@thinspace
199 \fi
200 \fi
201 \string:}
202 \declare@shorthand{mongolian}{!''}{;%
203 \ifhmode
204 \ifdim\lastskip>\z@
205 \unskip\nobreak\kern.1em
206 \else
207 \FDP@thinspace
208 \fi
209 \fi
210 \string!}
```



```

211 \declare@shorthand{mongolian}{?}{%
212   \ifhmode
213     \ifdim\lastskip>\z@
214       \unskip\nobreak\kern.1em
215     \else
216       \FDP@thinspace
217     \fi
218   \fi
219   \string?}

```

`\AutoSpaceBeforeFDP` `\FDP@thinspace` is defined as unbreakable spaces if `\AutoSpaceBeforeFDP` is activated or as `\@empty` if `\NoAutoSpaceBeforeFDP` is in use. The default is `\FDP@thinspace` `\AutoSpaceBeforeFDP`.

```

220 \def\AutoSpaceBeforeFDP{%
221   \def\FDP@thinspace{\nobreak\kern.1em}}
222 \def\NoAutoSpaceBeforeFDP{\let\FDP@thinspace\@empty}
223 \AutoSpaceBeforeFDP

```

`\FDPon` The next macros allow to switch on/off activeness of double punctuation signs.

```

\FDPoff 224 \def\FDPon{\bbl@activate{;}%
225   \bbl@activate{;}%
226   \bbl@activate{;}%
227   \bbl@activate{!}}
228 \def\FDPoff{\bbl@deactivate{;}%
229   \bbl@deactivate{;}%
230   \bbl@deactivate{;}%
231   \bbl@deactivate{!}}

```

`\system@sh@:` When the active characters appear in an environment where their Mongolian behaviour is not wanted they should give an ‘expected’ result. Therefore we define shorthands at system level as well.

```

\system@sh@; 232 \declare@shorthand{system}{:}{\string;}
233 \declare@shorthand{system}{;}{\string;}
234 \frenchpunct
235 \*frenchpunct&!spanishlig)
236 \declare@shorthand{system}{!}{\string!}
237 \declare@shorthand{system}{?}{\string?}
238 \frenchpunct&!spanishlig)

```

To be able to define the function of “”, we first define a couple of ‘support’ macros.

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

```

239 \begingroup \catcode'\ "12
240 \def\reserved@a{\endgroup
241   \def\@SS{\mathchar"7019 }
242   \def\dq{"}}
243 \reserved@a

```

Now we can define the doublequote macros: german and french quotes. We use definitions of these quotes made in `babel.sty`. The french quotes are contained in the T2* encodings.

```

244 \declare@shorthand{mongolian}{" '}{\glqq}
245 \declare@shorthand{mongolian}{" '}{\grqq}
246 \declare@shorthand{mongolian}{"<}{\flqq}
247 \declare@shorthand{mongolian}{">}{\frqq}

```

Some additional commands:

```

248 \declare@shorthand{mongolian}{""}{\hskip\z@skip}
249 \declare@shorthand{mongolian}{"~}{\textormath{\leavevmode\hbox{-}}{-}}
250 \declare@shorthand{mongolian}{"=}{\nobreak-\hskip\z@skip}
251 \declare@shorthand{mongolian}{"|}{%
252   \textormath{\nobreak\discretionary{-}{-}{\kern.03em}%
253     \allowhyphens}{}}

```

The next two macros for "-" and "--- are somewhat different. We must check whether the second token is a hyphen character:

```

254 \declare@shorthand{mongolian}{"-}{%

```

If the next token is '-', we typeset an emdash, otherwise a hyphen sign:

```

255   \def\mongolian@sh@tmp{%
256     \if\mongolian@sh@next-\expandafter\mongolian@sh@emdash
257     \else\expandafter\mongolian@sh@hyphen\fi
258   }%

```

TeX looks for the next token after the first '-': the meaning of this token is written to \mongolian@sh@next and \mongolian@sh@tmp is called.

```

259   \futurelet\mongolian@sh@next\mongolian@sh@tmp}

```

Here are the definitions of hyphen and emdash. First the hyphen:

```

260 \def\mongolian@sh@hyphen{%
261   \nobreak-\bbl@allowhyphens}

```

For the emdash definition, there are the two parameters: we must 'eat' two last hyphen signs of our emdash. . . :

```

262 \def\mongolian@sh@emdash#1#2{\cdash-#1#2}

```

`\cdash` . . . these two parameters are useful for another macro: `\cdash`:

```

263 %\ifx\cdash\undefined % should be defined earlier
264 \def\cdash#1#2#3{\def\tempx@{#3}%
265 \def\tempa@{-}\def\tempb@{~}\def\tempc@{*}%
266 \ifx\tempx@\tempa@\@Acdash\else
267 \ifx\tempx@\tempb@\@Bcdash\else
268 \ifx\tempx@\tempc@\@Ccdash\else
269 \errmessage{Wrong usage of cdash}\fi\fi\fi}

```

second parameter (or third for `\cdash`) shows what kind of emdash to create in next step

"--- ordinary (plain) Cyrillic emdash inside text: an unbreakable thinspace will be inserted before only in case of a *space* before the dash (it is necessary for dashes after display maths formulae: there could be lists, enumerations etc. started with "— where *a* is ..." i.e., the dash starts a line). (Firstly there were planned rather soft rules for user: he may put a space before the dash or not. But it is difficult to place this thinspace automatically, i.e., by checking modes because after display formulae TeX uses horizontal mode. Maybe there is a misunderstanding? Maybe there is another way?) After a dash a breakable thinspace is always placed;

```

270 % What is more grammatically: .2em or .2\fontdimen6\font ?
271 \def\@Acdash{\ifdim\lastskip>\z@\unskip\nobreak\hskip.2em\fi
272 \cyrdash\hskip.2em\ignorespaces}%

"--~   emdash in compound names or surnames (like Mendeleev–Klapeiron); this
dash has no space characters around; after the dash some space is added
\exhyphenalty

273 \def\@Bcdash{\leavevmode\ifdim\lastskip>\z@\unskip\fi
274 \nobreak\cyrdash\penalty\exhyphenpenalty\hskip\z@skip\ignorespaces}%

"--*   for denoting direct speech (a space like \enskip must follow the emdash);

275 \def\@Ccdash{\leavevmode
276 \nobreak\cyrdash\nobreak\hskip.35em\ignorespaces}%
277 \fi

```

`\cyrdash` Finally the macro for “body” of the Cyrillic emdash. The `\cyrdash` macro will be defined in case this macro hasn’t been defined in a fontenc file. For T2* fonts, `cyrdash` will be placed in the code of the English emdash thus it uses ligature ---.

```

278 % Is there an IF necessary?
279 \ifx\cyrdash\undefined
280 \def\cyrdash{\hbox to.8em{--\hss--}}
281 \fi

```

Here a really new macro—to place thinspace between initials. This macro used instead of `\`, allows hyphenation in the following surname.

```

282 %\declare@shorthand{mongolian}{",,}{\nobreak\hskip.2em\ignorespaces}

```

`\mdqon` All that’s left to do now is to define a couple of commands for “.

```

\mdqoff 283 \def\mdqon{\bbl@activate{""}
284 \def\mdqoff{\bbl@deactivate{""}

```

The Mongolian hyphenation patterns can be used with `\lefthyphenmin` and `\righthyphenmin` set to 2.

```

285 \providehyphenmins{\CurrentOption}{\tw@\tw@}
286 % temporary hack:
287 \ifx\englishhyphenmins\undefined
288 \def\englishhyphenmins{\tw@\thr@@}
289 \fi

```

Now the action `\extrasmongolian` has to execute is to make sure that the command `\frenchspacing` is in effect. If this is not the case the execution of `\noextrasmongolian` will switch it off again.

```

290 \addto\extrasmongolian{\bbl@frenchspacing}
291 \addto\noextrasmongolian{\bbl@nonfrenchspacing}

```

Next we add a new enumeration style for Mongolian manuscripts with Cyrillic letters, and later on we define some math operator names in accordance with Mongolian and Russian typesetting traditions.

`\Useg` We begin by defining `\Useg` which works like `\Alph`, but produces (uppercase) Cyrillic letters instead of Latin ones. The letters `CYRGUP`, and `SFTSN` are skipped, as usual for such enumeration.

```
292 \def\Useg#1{\expandafter\@Useg\csname c@#1\endcsname}
293 \def\@Useg#1{\ifcase#1\or
294   \CYRA\or\CYRB\or\CYRV\or\CYRG\or\CYRD\or\CYRE\or\CYRYO\or\CYRZH\or
295   \CYRZ\or\CYRI\or\CYRISHRT\or\CYRK\or\CYRL\or\CYRM\or\CYRN\or\CYRO\or
296   \CYROTLD\or\CYRP\or\CYRR\or\CYRS\or\CYRT\or\CYRU\or\CYRY\or\CYRF\or
297   \CYRH\or\CYRC\or\CYRCH\or\CYRSH\or\CYRSHCH\or\CYRHRDSN\or\CYRERY\or
298   \CYRSFTSN\or\CYREREV\or\CYRYU\or\CYRYA\else\@ctrerr\fi}
```

`\useg` The macro `\useg` is similar to `\alph`; it produces lowercase Mongolian letters.

```
299 \def\useg#1{\expandafter\@useg\csname c@#1\endcsname}
300 \def\@useg#1{\ifcase#1\or
301   \cyra\or\cyrb\or\cyrv\or\cyrg\or\cyrd\or\cyre\or\cyryo\or\cyrzh\or
302   \cyrz\or\cyri\or\cyrishrt\or\cyrk\or\cyrl\or\cym\or\cyrn\or\cyro\or
303   \cyrotld\or\cyrp\or\cyrr\or\cyrs\or\cyrt\or\cyru\or\cyry\or\cyrf\or
304   \cyrh\or\cyrc\or\cyrch\or\cyrsh\or\cyrshch\or\cyrhrdsn\or\cyrery\or
305   \cyrsftsn\or\cyrerev\or\cyryu\or\cyrya\else\@ctrerr\fi}
```

Set up default Cyrillic math alphabets. The math groups for Cyrillic letters are defined in the encoding definition files. First, declare a new alphabet for symbols, `\cyrmathrm`, based on the symbol font for Cyrillic letters defined in the encoding definition file. Note, that by default Cyrillic letters are taken from upright font in math mode (unlike Latin letters).

```
306 %\RequirePackage{textmath}
307 \@ifundefined{sym\cyrillicencoding letters}{}{%
308 \SetSymbolFont{\cyrillicencoding letters}{bold}\cyrillicencoding
309   \rmdefault\bfdefault\updefault
310 \DeclareSymbolFontAlphabet\cyrmathrm{\cyrillicencoding letters}
```

And we need a few commands to be able to switch to different variants.

```
311 \DeclareMathAlphabet\cyrmathbf\cyrillicencoding
312   \rmdefault\bfdefault\updefault
313 \DeclareMathAlphabet\cyrmathsf\cyrillicencoding
314   \sfdefault\mddefault\updefault
315 \DeclareMathAlphabet\cyrmathit\cyrillicencoding
316   \rmdefault\mddefault\itdefault
317 \DeclareMathAlphabet\cyrmathtt\cyrillicencoding
318   \ttdefault\mddefault\updefault
319 %
320 \SetMathAlphabet\cyrmathsf{bold}\cyrillicencoding
321   \sfdefault\bfdefault\updefault
322 \SetMathAlphabet\cyrmathit{bold}\cyrillicencoding
323   \rmdefault\bfdefault\itdefault
324 }
```

Some math functions in Mongolian and Russian math books have other names: e.g., `sinh` in Russian is written as `sh` etc. So we define a number of new math operators.

`\sinh`:

```
325 \def\sh{\mathop{\operator@font sh}\nolimits}
```

```

\cosh:
326 \def\ch{\mathop{\operator@font ch}\nolimits}
\tan:
327 \def\tg{\mathop{\operator@font tg}\nolimits}
\arctan:
328 \def\arctg{\mathop{\operator@font arctg}\nolimits}
arcctg:
329 \def\arcctg{\mathop{\operator@font arcctg}\nolimits}
The following macro conflicts with \th defined in Latin 1 encoding:
\tanh:
330 \addto\extrasmongolian{%
331 \babel@save{\th}%
332 \let\ltx@th\th
333 \def\th{\textormath{\ltx@th}%
334 \mathop{\operator@font th}\nolimits}}%
335 }
\cot:
336 \def\ctg{\mathop{\operator@font ctg}\nolimits}
\coth:
337 \def\cth{\mathop{\operator@font cth}\nolimits}
\csc:
338 \def\cosec{\mathop{\operator@font cosec}\nolimits}
And finally some other Mongolian and Russian mathematical symbols:
339 \def\Prob{\mathop{\kern\z@\mathsf{P}}\nolimits}
340 \def\Variance{\mathop{\kern\z@\mathsf{D}}\nolimits}
341 \def\nsd{\mathop{\cyrmathrm{\cyrn.\cyrn.\cyrd.}}\nolimits}
342 \def\nsk{\mathop{\cyrmathrm{\cyrn.\cyrn.\cyrk.}}\nolimits}
343 \def\NSD{\mathop{\cyrmathrm{\CYRN\CYRS\CYRD}}\nolimits}
344 \def\NSK{\mathop{\cyrmathrm{\CYRN\CYRS\CYRK}}\nolimits}
345 \def\nod{\mathop{\cyrmathrm{\cyrn.\cyrn.\cyrd.}}\nolimits}
346 \def\nok{\mathop{\cyrmathrm{\cyrn.\cyrn.\cyrk.}}\nolimits}
347 \def\NOD{\mathop{\cyrmathrm{\CYRN\CYRO\CYRD}}\nolimits}
348 \def\NOK{\mathop{\cyrmathrm{\CYRN\CYRO\CYRK}}\nolimits}
349 \def\Proj{\mathop{\cyrmathrm{\CYRP\cyrp}}\nolimits}
350 \DeclareRobustCommand{\No}{%
351 \ifmode{\nfss@text{\textnumero}}\else\textnumero\fi
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.
352 \ldf@finish{mongolian}
353 \code

```