The Belarusian Language
in the babel system
Version 1.5
Andrei Shadura*
Released 2018/10/09

Contents

1 The Belarusian Language Definition File 2

2 Usage 2
  2.1 \TeX 3
  2.2 \LaTeX 3
  2.3 Xe\TeX 4
  2.4 Official and ‘classic’ spelling 4

3 User’s commands 5
  3.1 Active character 6
  3.2 Math commands 7

4 \TeX nical details 7

5 Known problems 7

6 History 8

7 Implementation 8
  7.1 Initial setup 8
  7.2 Output encoding 8
  7.3 Input encoding 12
  7.4 Shorthands 12
    7.4.1 Quotes 13
    7.4.2 Emdash, endash and hyphenation sign 13
  7.5 Switching to and from Belarusian 15
    7.5.1 Caption names 15
    7.5.2 Date in Belarusian 19
    7.5.3 Hyphenation patterns 21
    7.5.4 Extra definitions 21
  7.6 Alphabetic counters 21
  7.7 Cyrillic math 23
  7.8 Final settings 25

---

*E-mail: andrew@shadura.me.
1 The Belarusian Language Definition File

The file belarusian.ldf is the source file for the Belarusian Language Definition file belarusian.ldf to be loaded by the babel package with the option belarusian. It’s based on the Russian language definition file derived by Igor A. Kotelnikov from the original version of belarusian.ldf, first released by Olga Lapko and Johannes Braams and then adapted to the T2* and X2 Cyrillic encodings by Vladimir Volovich and Werner Lemberg, and the Belarusian language definition file written by Aleksey Novodvorsky based on even older version of the Russian language definition file.

Just like russianb.ldf, belarusian.ldf is designed to work both with legacy non-unicode (8-bit) and new Unicode encodings of the source document files (input encodings) and of the font files (font encodings). This is achieved by excluding (bypassing) the \cyr... macros, which map every letter in a source file with given input encoding to a corresponding code point in a font file with a given font encoding when running modern engines, such as LuaLaTeX or XeLaTeX, in native Unicode mode instead of legacy engines, such as BtEPX or PDFbEPX, or Unicode engines in a compatibility (8-bit) mode. A few obsolete and controversial macros has been eliminated in first public release of version 1.2 of russianb.ldf.

Similarly to russianb.ldf’s support for typesetting ancient and Church Slavonic books, belarusian.ldf supports two versions of Belarusian orthography, the official one (official) and the ‘classic’ (Taraškievič) orthography.

2 Usage

Typesetting Belarusian texts implies that a special input and output encodings should be used. Input encodings are those which are used in source (.tex) file. Output encoding is also known as the font encoding. It is implemented within the font files.

Generally, the user may choose between different available Cyrillic encodings. The current support for Cyrillic uses LH family of MetaFont fonts and theirs Postscript versions such as CM-super. LuaBtEPX and XeBtEPX, being the Unicode-based successors of BtEPX, allow also for any Open Type (OTF) and True Type (TTF) fonts which has Cyrillic script, e.g. Computer Modern Unicode, Linux Libertine, and many other system fonts that came with Linux, Mac and Windows operating systems.

With the advent of Unicode, BtEPX community are moving towards eliminating all existing encodings in favor of Unicode, but nowadays one should take care when switching from BtEPX to LuaBtEPX or XeBtEPX since different packages should be loaded for those compilers.

Since earlier versions babel did not support XeBtEPX (at least for some languages including Belarusian), the polyglossia package was generally recommended in the past for use with XeBtEPX as a replacement for babel. Nowadays, babel can be used with any engines, including BtEPX, PDFbEPX, LuaBtEPX, and XeBtEPX. Nevertheless some troubles may occur with some languages which have no promptly updated .ldf files.

---

1The file described in this document has the version number 1.5 and was last revised on 2018/10/09.
2.1 LaTeX

When user’s document is compiled with \texttt{latex} or \texttt{pdflatex}, recommended set of packages includes the \texttt{inputenc} and \texttt{fontenc} packages. They should be loaded before \texttt{babel}, for example,

\begin{verbatim}
\usepackage[T1,T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english,belarusian]{babel}
\end{verbatim}

Some variations in the order of loading the packages are allowed in this case but it is better to follow one and the same convention at all circumstances: the \texttt{babel} package should go last, and \texttt{fontenc} must be the first.

Input encoding should be declared as option to the \texttt{inputenc} package. Known Cyrillic encodings include \texttt{cp1251} (Windows), \texttt{koi8-u} (UNIX) and their variants. Nowadays, this list is appended with \texttt{utf8} input encoding.

Output encodings (also known as font encodings) are declared as options to the \texttt{fontenc} package. Known Cyrillic encodings are \texttt{T2A}, \texttt{T2B}, \texttt{T2C}, \texttt{LCY}, and \texttt{X2}; \texttt{LWN} is not supported since \texttt{LWN} has been excluded from the \texttt{cyrillic} bundle of related files.

2.2 Lualatex

If Unicode fonts are not available, \texttt{Lualatex} can run in compatibility (8-bit) mode to use same font as \texttt{LaTeX} does. However the package \texttt{inputenc} does not work with \texttt{Lualatex} and should be substituted with \texttt{luainputenc}. Source file is to be converted to \texttt{UTF8} (Unicode-8) encoding; it is the only input encoding accepted by \texttt{Lualatex}. The 8-bit mode is invoked by the following sequence of packages:

\begin{verbatim}
\usepackage[T1,T2A]{fontenc}
\usepackage[utf8]{luainputenc}
\usepackage[english,belarusian]{babel}
\end{verbatim}

The order of the packages is crucial for \texttt{Lualatex} in 8-bit mode. Since both \texttt{luainputenc} and \texttt{babel} should know what is a selected font encoding, the \texttt{fontenc} package should be loaded first. Input encoding management for \texttt{Lualatex} is needed only for compatibility with old documents. For new documents, using UTF-8 encoding and Unicode fonts is strongly recommended. \textit{You’ve been warned!} See \url{tex.stackexchange.com/questions/31709/can-one-instruct-lualatex-to-use-t2a-encoded-fonts}.

To invoke Unicode mode, one needs to load the \texttt{fontspec} package instead of \texttt{luainputenc} and \texttt{fontspec} and explicitly indicate which True Type or Open Type fonts should be used for romanic, sans-serif and monospaced types. The following example shows how to load Computer Modern Unicode (CMU) fonts, which is a part of all modern \texttt{Lualatex} distributions:

\begin{verbatim}
\usepackage[fontspec]
  \defaultfontfeatures{Renderer=Basic,Ligatures=TeX}
  \setmainfont{CMU Serif}
  \setsansfont{CMU Sans Serif}
  \setmonofont{CMU Typewriter Text}
\end{verbatim}
The `\defaultfontfeatures` declares default font features for subsequent `\setmainfont` (which sets romanic fonts), `\setsansfont` (sans-serif) and `\setmonofont` (monospaced font). Font features can be set upon per font bases; for example:

\begin{verbatim}
\usepackage{fontspec}
\setmainfont[Renderer=Basic,Ligatures={TeX}]{CMU Serif}
\setsansfont[Renderer=Basic,Ligatures={TeX, Historic}]{CMU Sans Serif}
\setmonofont{CMU Typewriter Text}
\usepackage[english, belarusian]{babel}
\end{verbatim}

Here `Renderer=Basic, Ligatures={TeX}` activates ligatures which are existed in \LaTeX. Recall that the language enlisted last in the list of options of the `babel` package is assumed to be the main language of the document, which is also active language right after `\begin{document}`. As of version 3.9, the main language can be set as a value of the `main` option as follows:

\begin{verbatim}
\usepackage{fontspec}
\usepackage[english, main=belarusian, german]{babel}
\end{verbatim}

### 2.3 \textit{XeLaTeX}

In \textit{XeLaTeX}, there is also a special mode for 8-bit compatibility. One can use `\XeTeXinputencoding` to change the input encoding temporarily, and the ”bytes” encoding makes \textit{XeLaTeX} works like a 8-bit \LaTeX engine:

\begin{verbatim}
\XeTeXinputencoding "bytes"
\usepackage[utf8]{inputenc}
\usepackage[T2A]{fontenc}
\usepackage[english, belarusian]{babel}
\end{verbatim}

\textit{XeLaTeX} can use a different input encoding but it always uses the Unicode internally, so that `\XeTeXinputencoding` performs a conversion of the input into Unicode; see \url{tex.stackexchange.com/questions/36188/do-xetex-and-luatex-always-use-unicode}. Unicode mode is set up same way as for \textit{LuaLaTeX}, however the option `Renderer=Basic` can be dropped:

\begin{verbatim}
\usepackage{fontspec}
\defaultfontfeatures{Ligatures={TeX}}
\setmainfont{CMU Serif}
\setsansfont{CMU Sans Serif}
\setmonofont{CMU Typewriter Text}
\usepackage[english, belarusian]{babel}
\end{verbatim}

### 2.4 Official and ‘classic’ spelling

By default, the official spelling is enabled. The ‘classic’ (Taraškievič) spelling can be enabled by setting the attribute to `classic` or `tarask`. To set an attribute, put the `\languageattribute` macro within a document preamble after `babel`, for example:

\begin{verbatim}
\usepackage[english, belarusian]{babel}
\languageattribute{belarusian}{classic}
\end{verbatim}
Setting the \texttt{classic} attribute changes the built-in strings (caption names) and a date format. For example, the references list will be entitled as ‘Список литературы’ by default and as ‘Список литературы’ if the Belarusian language attribute is set to \texttt{classic}. The same result can be achieved using a modifier as follows:

\begin{verbatim}
\usepackage[english, belarusian.classic]{babel}
\end{verbatim}

Using a modifier in a package option is often better. A modifier is set after the language name, and is prefixed with a dot (only when the language is set as package option, neither global options nor the main key accept them).

Both attributes \texttt{classic} and \texttt{tarask} may be used interchangeably.

3 User’s commands

In a multilingual document, some typographic rules are language dependent and should apply to the whole document.

Regarding local typography, the macro \texttt{\selectlanguage{belarusian}} switches to the Belarusian language, with the following effects:

1. Belarusian hyphenation patterns are made active;
2. \texttt{\today} prints the date in Belarusian;
3. the caption names are translated into Belarusian (\LaTeX\ only);
4. emdash typed by the ligature \texttt{---} might be 20\% shorter when Belarusian is the current language; the result depends on the current encoding; \texttt{---} always produce long emdash in Lua\TeX\ and Xe\TeX\ since these engines use same encodings for all languages;
5. emdash typed by the ligature ‘’’---’’’ in Belarusian is 20\% shorter, however the ligature ‘’’---’’’ might not be defined in other languages; a shorter emdash (i.e. \texttt{\cyrdash}) can be typeset in any language using special macros enlisted in table 1.

By default, the official spelling is used for built-it strings (caption names) and the date. The spelling can be reverted to the classic one by setting the language attribute to \texttt{classic} in the document preamble as discussed in Sec. 2.4.

Since Belarusian has its own numbering system, \texttt{belarusian.1df} adds macros \texttt{\asbuk\{counter\}} and \texttt{\Asbuk\{counter\}} for formatting numbers appropriately the alphabetic sequence in the Belarusian alphabet. Additional commands are provided to typeset quotes:

1. French quotation marks can be entered using the commands \texttt{\guillemotleft} and \texttt{\guillemotright} which work in \LaTeX\ 2\epsilon and Plain\TeX.\XeTeX.
2. German quotation marks can be entered using the commands \texttt{\glqq} and \texttt{\grqq} which work in \LaTeX\ 2\epsilon and Plain\TeX.\XeTeX.

The macro \texttt{\Belarusian} is now defined as an alias for \texttt{\selectlanguage{belarusian}}, and its “opponent” \texttt{\English}, existed in \texttt{belarusian.1df} prior to version 1.2 has been removed since the Belarusian language definition file is wrong place for definition of macros which switch to a distinct other language.

The macro \texttt{\textcyrillic\{text\}} is intended to typeset small chunks of text in Belarusian; it is essentially an alias for \texttt{\foreignlanguage{belarusian}\{\text\}}.
3.1 Active character

Table 1 shows macros and active string which can be used to typeset various dashes and quotes. In the Belarusian language, the character "" is made active. It can be considered as second escape character in addition to \. Some dashes and all quotes can be typed using both active character "" and ordinary macros as indicated in the table. However, some shorthanded hyphenations have no macro counterpart.

Table 1: Extra definitions made by belarusan.ldf

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\glqq</td>
<td>German opening double quote (looks like &quot;&quot;).</td>
</tr>
<tr>
<td>\grqq</td>
<td>German closing double quote (looks like &quot;).</td>
</tr>
<tr>
<td>\guillemotleft</td>
<td>French opening double quote (looks like &lt;&lt;).</td>
</tr>
<tr>
<td>\guillemotright</td>
<td>French closing double quote (looks like &gt;&gt;).</td>
</tr>
<tr>
<td>dq</td>
<td>Original quotes character (&quot;&quot;).</td>
</tr>
<tr>
<td>\babelhyphen{soft}</td>
<td>Optional (soft) hyphen sign, similar to - but allows hyphenation in the rest of the word; equivalent to \babelhyphen{soft} in babel 3.9.</td>
</tr>
<tr>
<td>\babelhyphen{empty}</td>
<td>Similar to &quot;&quot;- but prints no hyphen sign (used for compound words with hyphen, e.g. x-&quot;&quot;y); equivalent to \babelhyphen{empty} in babel 3.9.</td>
</tr>
<tr>
<td>\babelhyphen{nobreak}</td>
<td>Compound word mark without a breakpoint, prints hyphen prohibiting hyphenation at the point; equivalent to \babelhyphen{nobreak} in babel 3.9.</td>
</tr>
<tr>
<td>\babelhyphen{hard}</td>
<td>A compound word mark with a breakpoint, prints hyphen allowing hyphenation in the composing words. equivalent to \babelhyphen{hard} in babel 3.9.</td>
</tr>
<tr>
<td>\babelhyphen</td>
<td>nobreak}</td>
</tr>
<tr>
<td>\cyrdash</td>
<td>Cyrillic emdash (does not care spaces around).</td>
</tr>
<tr>
<td>\cdash---</td>
<td>Cyrillic emdash in plain text.</td>
</tr>
<tr>
<td>\cdash--</td>
<td>Cyrillic emdash in compound names (as in Mendeleev'&quot;--&quot;Klapeiron).</td>
</tr>
<tr>
<td>\cdash--*</td>
<td>Cyrillic emdash for denoting direct speech.</td>
</tr>
<tr>
<td>,</td>
<td>Thin space (allows further hyphenation as in D.&quot;&quot;,Mendeleev).</td>
</tr>
</tbody>
</table>

Note that the standard soft hyphen \- is equivalent to \babelhyphen{soft}.

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

The French quotes are also available as ligatures ‘<<’ and ‘>>’ in 8-bit Cyrillic
font encodings (LCY, X2, T2*) and in Unicode encodings (EU1 and EU2) and as ‘<’ and ‘>’ characters in 7-bit Cyrillic font encodings (OT2 and LWN).

In Unicode encodings EU1 and EU2 cyrdashes and quotes can be typed as single character if text editor makes it possible to insert characters which absent of standard keyboard. This method works as well for 8-bit fonts encoded according to T2A if source file is encoded with cp1251 or utf8.

By default, active double quote is switched on. It can be switched off any time using \shorthandoff{"} and the switched on again using \shorthandon{"}. The aliases \dqoff and \dqon for these two macros has been removed from belarusian.ldf starting from version 1.3 in favour of the macros \shorthandoff and \shorthandon provided in the babel core.

### 3.2 Math commands

belarusian.ldf defines few macros than can be used independently of current language. These are 9 macros to be used in math mode to type the names of trigonometric functions common for Belarusian documents: \sh, \ch, \tg, \ctg, \arctg, \arccctg, \th, \cth, and \cosec. Cyrillic letters in math mode can be typed with the aid of text commands such as \textbf, \textsf, \textit, \texttt, e.t.c.

The macros \Prob, \Variance, \NAD, \nad, \NAK, \nak, \Proj print some rare Belarusian mathematical symbols.

### 4 TeXnical details

The packages inputenc and luainputenc make Cyrillic letters active so that a compiler converts them into corresponding \cyr... macro at compilation time. For example, Belarusian letter ‘а’ matches macro \cyra, and capital Belarusian letter ‘А’ matches \CYRA. The package fontenc then matches every macro \cyr... to corresponding glyph in a font file depending on a declared font encoding.

Nowadays, Unicode makes \cyr... macros outdated since both source file and font file are encoded consistently. These macros should therefore be removed because mixing them with Unicode characters breaks sorting mechanism of such utilities as bibtex and makeindex. For the sake of backward compatibility, \cyr... are still kept for \LaTeX, but they are bypassed if Lua\LaTeX or Xe\LaTeX are detected.

Some inconsistencies of prior versions of belarusian.ldf was also overcame in the version 1.2. Those users who used \selectlanguage macro, defined in the core babel system, to switch between different languages should not worry. However, the macros \Belarusian, \English and their aliases \rua, \cyr, \Eng are modified or removed as they did not conform the mechanism of language switching encoded into the core of babel and therefore can mess it.

### 5 Known problems

Before switching from a legacy 8-bit engine (tex, pdftex) to an Unicode engine (xetex, luatex) and vise versa delete all .aux, .toc, .lot, .lof files as they might have stored incompatible internal encodings.

\cyrdash might produce dashes of different width in same document after switching to/from Belarusian language if the document is compiled with legacy engines.
6 History

7 Implementation

7.1 Initial setup

The macro \LdfInit performs a couple of standard checks that must be made at
the beginning of a language definition file, such as checking the category code of
the @-sign, preventing the .ldf file from being processed twice, etc.

\LdfInit{belarusian}{captionsbelarusian}

First, we check if LuaL\TeX or XeL\TeX is running. If so, we set boolean key
\if@uni@ode@be to true. It will be used to eliminate \cyr commands, which
were introduced in L\TeX2e to handle various Cyrillic input encoding. With the
advent of Unicode L\TeX is moving to universal input encoding, so we consider
these \cyr commands as obsolete. They are preserved though for backward
compatibility in case if L\TeX or PDFL\TeX are running.

We don’t load the ifluatex or ifxetex package because \RequirePackage
is not allowed at the stage of processing options (note that babel loads this file right
when it processes its own options) but we borrow code from these packages.

\ifdefined\if@uni@ode@be
  \PackageError{babel}{if@uni@ode already defined.}{MessageBreak
  Please contact author of belarusian.ldf}
  \relax
\fi
\newif\if@uni@ode@be
\ifdefinedluatexversion \@uni@ode@be true \else
\ifdefined\XeTeXrevision \@uni@ode@be true \fi\fi

Check if hyphenation patterns for the Belarusian language have been loaded in
language.dat. Namely, we check for the existence of \l@belarusian. If it is not
defined, we declare Belarusian as dialect for the default language number 0 which
almost for sure is English.

\ifx\l@belarusian\undefined
  \nopatterns{Belarusian}
  \adddialect\l@belarusian0
\fi

Now \l@belarusian is always defined.

7.2 Output encoding

We need to know font encoding that is supposed to be active at the end of the babel
package. Default font encoding, set by L\TeX core, is OT1. This can be changed
by the fontenc package in case of L\TeX and by fontsip package in case of
LuaL\TeX. It matters weather these packages are loaded before of after babel. In
the latter case or if these packages are not loaded at all, belarusian.ldf ignores
their effect and tries to provide some reasonable settings. In particular, T2A
will be selected for Belarusian language if L\TeX is running but EU1 in case of XeL\TeX
and EU2 in case of LuaL\TeX.
The macro \latinencoding keeps the name of Latin encoding. It is defined in babel.def and is wrapped into \AtBeginDocument to allow for late loading fontenc. Therefore it does not matter whether babel is loaded before or after the fontenc. As of version 1.2, definition of \latinencoding was removed from belarusian.ldf since it is overruled in babel.def. For example, after

```latex
\usepackage[T1,T2A]{fontenc}
\usepackage[english,belarusian]{babel}
```

as well as after

```latex
\usepackage[english,belarusian]{babel}
\usepackage[T1,T2A]{fontenc}
```

\latinencoding will be set to T1. After

```latex
\usepackage[english,belarusian]{babel}
```

\latinencoding will be set to OT1.

In Unicode mode, the package fontspec should be loaded instead of fontenc to make font preparation; fontspec loads the package xunicode which sets current encoding (kept in \cf@encoding) to EU1 for Xe\LaTeX{} and EU2 for Lua\LaTeX{}, and the babel package sets the macro \latinencoding to \cf@encoding. Since babel scan for value \cf@encoding within \AtBeginDocument, \latinencoding will be set to either EU1 for Xe\LaTeX{} or EU2 for Lua\LaTeX{} no matter which of the packages, babel or fontspec is loaded first.

\cyrillicencoding There is a limited list of encodings appropriate for Cyrillic text. We will look which of them is declared and keep its name in the macro \cyrillicencoding. Correct (but obsolete and now deleted) 7-bit Cyrillic encoding mass LWN. Correct 8-bit Cyrillic encodings are T2A (default for 8-bit compilers), T2B, T2C, LCY and X2. Correct utf8 encodings are EU1 (default for Xe\LaTeX{}), EU2 (default for Lua\LaTeX{}).

In 8-bit (\LaTeX{}) mode, user may choose between different non-unicode Cyrillic encodings—e.g., X2 or LCY. If user wants to use another font encoding rather than default (T2A), he has to load the corresponding file before babel.sty.

Remember that for the Belarusian 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 Belarusian phrase or vice versa.

We parse the \cdp@list containing encodings known to \LaTeX{} in the order they were loaded by the time when babel is loaded. We set the \cyrillicencoding to the last loaded encoding in the list of supported Cyrillic encodings: OT2, LCY, X2, T2C, T2B, T2A. In Unicode mode we also try the Unicode encodings EU1 and EU2; 8-bit encodings are kept for Unicode compilers (Lua\LaTeX{} and Xe\LaTeX{}) since they can run in compatibility (8-bit) mode.

\begin{verbatim}
\def\@setcyrillicencoding{%
  \def\sce@a##1##2{%\edef\sce@b{##1}%\edef\sce@c{##2}%\ifx\sce@b\sce@c\let\cyrillicencoding\sce@c\fi}%
\end{verbatim}
The last lines are to free the memory occupied by the macros \@setcyrillicencoding and \sce@x that are useless in the document. The contents of \@begindocumenthook is cleared automatically.

If \cyrillicencoding is still undefined, we issue warning and provide reasonable default value for \cyrillicencoding. We then load default encoding definitions; we use the lowercase names (i.e., \lcyenc .def instead of \LCYenc .def) when we do that.

As final wisdom, we repeat \@setcyrillicencoding at \begin{document} time. We could not avoid previous call to \@setcyrillicencoding since compiler scan .aux file before it executes delayed code, and .aux may contain \set@langauge{belarusian}; the latter rises an error if \cyrillicencoding would not be defined by that time.

For the sake of backward compatibility we keep the macro \Belarusian but redefine its meaning; now \Belarusian is simply an alias for \selectlanguage{belarusian}.
We define \texttt{\cyrillictext} and its alias \texttt{\cyr} but remove another alias \texttt{\rus}; these macros are intended for use within \texttt{babel} macros and do not perform complete switch of the language. In particular, they do no switch captions and the name of current language stored in the macro \texttt{\languagename}. This inconsistency might break some assumptions embedded into \texttt{babel}'s. For example, the \texttt{\iflanguage} macro will fail.

Second, \texttt{\cyrillictext} does not activate shorthands, so that ‘<’, ‘>’, ‘“’, ‘”’, ‘---’ will not work.

And third, \texttt{\cyrillictext} does not write its trace to \texttt{.aux} file, which might result in wrong typesetting of table of contents, list of table and list of figures in multilingual documents.

Due to any of these reasons the use of the declaration \texttt{\cyrillictext} and its aliases in ordinary text is strongly discouraged. Instead of the declaration \texttt{\cyrillictext} it is recommended to use \texttt{\Belarusian} or the command \texttt{\foreignlanguage} defined in the \texttt{babel} core; their functionality is similar to \texttt{\selectlanguage{belarusian}} but they did not switch caption names, dates and shorthands.


def

\begin{verbatim}
58 \DeclareRobustCommand{\Belarusian}{\selectlanguage{\belarusian}}
59 \DeclareRobustCommand{\cyrillictext}{%
60 \fontencoding{\cyrillicencoding}\selectfont
61 \let\encodingdefault{\cyrillicencoding}
62 \expandafter{\set@hyphenmins{\belarusianhyphenmins}}
63 \language{\l@belarusian}%
64 \let{\cyr{\cyrillictext}}

Starting from version v.1.2 we remove the \texttt{\English} macro and its aliases. We believe that reasonable place for defining these macros would be \texttt{englishb.1df}. Note also that the macro \texttt{\English} and its alias \texttt{\Eng} are absent in \texttt{belarusian.1df}'s counterpart in the package \texttt{polyglossia}, analog of \texttt{babel} for Xe\LaTeX.

\begin{verbatim}
65 \%\DeclareRobustCommand{\English}{%
66 \% \fontencoding{\latinencoding}\selectfont
67 \% \let\encodingdefault{\latinencoding}
68 \% \expandafter{\set@hyphenmins{\englishhyphenmins}}
69 \% \language{\l@english}%
70 \%\let{\Eng{\English}}

NEXT PART OF CODE SHOULD BE MOVED TO \texttt{X2enc.def}, \texttt{X2enc.dfu}, IF NEEDED. Since the X2 encoding does not contain Latin letters, we should make some redefinitions of \LaTeX macros which implicitly produce Latin letters.

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

\begin{verbatim}
71 \expandafter{\ifx+cname \texttt{\textumlaut{\textasciitilde}}\textasciitilde{\endcname}\relax}else
72 \DeclareTextSymbolDefault{\AA}{OT1}{\r{\AA}}
73 \DeclareTextSymbolDefault{\aa}{OT1}{\r{\a}}
74 \DeclareTextCommand{\aa}{OT1}{\r{\AA}}
75 \DeclareTextCommand{\AA}{OT1}{\r{\a}}
76 \fi

The macro \texttt{\cyrillictext} switches current (e.g., Latin) font encoding to a Cyrillic font encoding stored in \texttt{\cyrillicencoding}. The macro \texttt{\latintext} switches
back. This method assumes that an font encoding is a Latin one. But in fact the latter assumption does not matter if any other language is switched on using same method, i.e. if corresponding .ldf file defines required macros to switch that language on from same standard (Latin) state. Since \latintext is defined by the core of babel we do not repeat its definition here.

\textcyrillic {{text}}

The macros \cyrillictext and \latintext are declarations. For shorter chunks of text the commands \textcyrillic and \textlatin can be used.

The macro \textcyrillic takes an argument which is then typeset using the requested font encoding. It is thus an equivalent or \foreignlanguage{belarusian}.

7.3 Input encoding

User should 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 inputenc package should be explicitly called by \usepackage{inputenc} before babel. Note however that default font encoding T2A fits well enough to Belarusian version of Windows ANSI encoding which is almost equivalent to cp1251.

\textcyrillic}

7.4 Shorthands

The double quote character " is declared to be active in Belarusian language.

\textcyrillic
Obsolete: Active double quote can be both activated and deactivated at any time using the macros \mdqon and \mdqoff.

\begin{verbatim}
103 \%\def\mdqon\bbl@activate{"}
104 \%\def\mdqoff\bbl@deactivate{"}
\end{verbatim}

Initial activation state will set to on later in section 7.5.4.

The active character ‘‘ is used as indicated in table 1. We save the original double quote character in the \dq macro to keep it available. The math accent ‘‘ can now be typed as ‘‘.

\begin{verbatim}
103 \begingroup \catcode'"12
104 \def\reserved@a{\endgroup
105 \def\@SS\mathchar"7019 }
106 \def\dq{"}
107 \reserved@a
\end{verbatim}

7.4.1 Quotes

We set ‘‘ and ‘‘ as shorthands for \textquotedblbase and \textquotedblleft, respectively. Prior to ver.1.2, these shorthands were defined through german quotes \glqq and \grqq, which in their turn are defined in babel.def via \textquotedblbase and \textquotedblleft, respectively. It occurred, that old definition caused errors in Unicode mode if fontspec is loaded.

Prior to version 1.2, the shorthands ‘‘< and ‘‘> were declared to be equivalents for the French quotes \flqq and \frqq, respectively. They are defined in babel.def via \guillemotleft and \guillemotright. However, \flqq and \guillemotleft (and their right counterparts) are typeset differently if current encoding is not T1. Therefore, since v.1.2, we define ‘‘< and ‘‘> directly through \guillemotleft and \guillemotright.

\begin{verbatim}
108 \declare@shorthand{belarusian}{"}{\textquotedblbase}
109 \declare@shorthand{belarusian}{","}{textquotedblright}
110 \declare@shorthand{belarusian}{"<}{\guillemotleft}
111 \declare@shorthand{belarusian}{">}{\guillemotright}
\end{verbatim}

Next set of shorthands is intended for variations of standard macro \textasciitilde which indicates explicitly breakpoint for hyphenation in a word. Meaning of these shorthands is explained in table 1.

\begin{verbatim}
112 \declare@shorthand{belarusian}{"}{\hskip\z@skip}
113 \declare@shorthand{belarusian}{","}{\textasciitilde\leavevmode\hbox{-} {{}}}
114 \declare@shorthand{belarusian}{"}{\nobreak- \hskip\z@skip}
115 \declare@shorthand{belarusian}{"}{%\textasciitilde\nobreak\discretionary{-}{}{-}}
116 \\textasciitilde\nobreak\discretionary{-}{}{-}}\{egin{small}\kern.03em}\allowhyphens\end{small}\}
\end{verbatim}

7.4.2 Emdash, endash and hyphenation sign

To distinguish between ‘‘- and ‘‘--- we must check whether the next after - token is a hyphen character. If it is, we output an endash, otherwise a hyphen sign. Therefore T\textasciitilde X looks for the next token after the first ‘‘-, writes its meaning to \belarusian@sh@next and finally call for \belarusian@sh@tmp.

\begin{verbatim}
117 \declare@shorthand{belarusian}{"-}{%\def\belarusian@sh@tmp{%
118 \\belarusian@sh@tmp{%
\end{verbatim}
Two macros \belarusian@sh@hyphen and \belarusian@sh@emdash called by \belarusian@sh@tmp are defined below. The second of them has two parameters since it must gobble next two hyphen signs.

\begin{verbatim}
\def\belarusian@sh@hyphen{\nobreak\-\bbl@allowhyphens}
\def\belarusian@sh@emdash#1#2{\cdash-#1#2}
\end{verbatim}

Initsturn, \belarusian@sh@emdash simply calls for \cdash which has rich use. It analyses 3rd of 3 characters and calls for one of few predefined macros \@Acdash, \@Bcdash, \@Ccdash.

\begin{verbatim}
\def\cdash#1#2#3{\def\tempx@{#3}\
\def\tempa@{-}\def\tempb@{~}\def\tempc@{*}\
\ifx\tempx@\tempa@\@Acdash\else\ifx\tempx@\tempb@\@Bcdash\else\ifx\tempx@\tempc@\@Ccdash\else\errmessage{Wrong usage of cdash}\fi\fi\fi}
\end{verbatim}

All these 3 internal macro call for \cyrdash, which type Cyrillic emdash, but put different spaces around the dash.

\begin{verbatim}
\def\@Acdash{\ifdim\lastskip>\z@\unskip\nobreak\hskip.2em\fi\cyrdash\hskip.2em\ignorespaces}\
\def\@Bcdash{\leavevmode\ifdim\lastskip>\z@\unskip\fi\nobreak\cyrdash\penalty\exhyphenpenalty\hskip\z@skip\ignorespaces}\
\def\@Ccdash{\leavevmode\nobreak\cyrdash\nobreak\hskip.35em\ignorespaces}
\end{verbatim}

The \cyrdash can be defined in a fontenc file as it is done for the T2* encodings. We provide fake definition of \cyrdash only if it is not defined there.

Cyrillic T2* fonts contains shorter (Cyrillic) emdash. It can be typeset also via the ligature ---. So, if \laticencoding differs from T2* switching of languages also changes the emdash length typed as ligature in the source file.
Finally, we define a shorthand thin space to be placed between initials as in D.", Mendeleev. When used instead of \, as in D. \, Mendeleev it allows hyphenation in the next word.

\declare@shorthand{belarusian}{"}{{\nobreak\hspace{2em}\ignorespaces}}

7.5 Switching to and from Belarusian

Now we define additional macros used to reset current language to Belarusian and back to some original state. The package babel based on the assumption that original state is characterized by a Latin encoding. Previously, for back reset the macro \OriginalTeX was used, but now use \latintext for the same purpose.

7.5.1 Caption names

First, we define Belarusian equivalents for Belarusian caption names.

\captionsbelarusian The macro \captionsbelarusian defines caption names used in the four standard document classes provided with L\TeX. The macro \cyr activates Cyrillic encoding. It could be dropped if we would be sure that Belarusian captions are called only if current language is Belarusian. However, the macros such as \Belarusian do not conform to strict rules of the package babel as explained in the above.

As of version v.1.2 we eliminate \cyr macros from caption names if Unicode engine is running. In the latter case, Cyrillic letters are typed in by their Unicode code-points, the ~~~-abcd notation is not used since it causes error at compilation time in case if L\TeX is running and utf8 input encoding is not declared.

\if@uni@ode@be %\captionsbelarusian@official\addto\captionsbelarusian{%\def\prefacename{Прадмова}\[babel]\def\refname{Спіс літаратуры}\[only article]\def\abstractname{Анатацыя}\[only article, report]\def\bibname{Літаратура}\[only book, report]\def\chaptername{Глава}\[only book, report]\def\appendixname{Дадатак}%\def\contentsname{Змест}%\let\tocname=\contentsname\def\listfigurename{Спіс ілюстрацый}%\def\listtablename{Спіс табліц}%\def\indexname{Прадметны паказальнік}%\def\authorname{Паказальнік імён}%\def\figurename{Рыс.}%\def\tablename{Табліца}%\def\partname{Частка}%\def\encname{укл.}%\def\ccname{зых.}%\def\headtoname{вх.}%\def\pagename{с.}%[letter]\def\alsoname{гл. таксама}%\def\proofname{Доказ}%[amsthm]\def\glossaryname{Слоўнік тэрмінаў}%[glossaries] \{Acronyms\}
Additional definitions for the package \texttt{nocemcl}:

\begin{verbatim}
\addto\captionsbelarusian{%
\def\nomname{Азначэнні}%
\def\eqdeclaration#1{, гл.\nobreakspace(#1)}%
\def\pagedeclaration#1{, стар.\nobreakspace#1}%
}
\fi

Additional captions for the \texttt{revtex} class.

\begin{verbatim}
\addto\captionsbelarusian{%
\def\lofname{\listfigurename}
\def\lotname{\listtablename}
\def\figuresname{Рысункі}%{Figures}%
\def\tablesname{Табліцы}%{Tables}%
\def\appendixesname{Дадаткі}%{Appendixes}%
\def\acknowledgmentsname{Падзякі}%{Acknowledgments}%
\def\andname{и}%{and}%
\def\@pacs@name{PACS коды: }%{PACS numbers: }%
\def\@keys@name{Ключавыя словы: }%{Keywords: }%
\def\Dated@name{Дата: }%{Dated: }%
\def\Received@name{Атрымана }%{Received }%
\def\Revised@name{Выпраўленая версія }%{Revised }%
\def\Accepted@name{Прынята }%{Accepted }%
\def\Published@name{Апублікавана }%{Published }%
}
\fi

Now we proceed to the classic version in Unicode encoding.

\begin{verbatim}
\addto\captionsbelarusian@tarask{%
\def\prefacename{Прадмова}% \[babel\]
\def\refname{Сыпіс літаратурнага }% \[only\ article\]
\def\abstractname{Анатацыя}% \[only\ article,\ report\]
\def\bibname{Літаратура}% \[only\ book,\ report\]
\def\chaptername{Глава}% \[only\ book,\ report\]
\def\appendixname{Дадатак}\
\def\contentsname{Зьмест}\
\let\tocname=\contentsname
\def\listfigurename{Сыпіс ілюстрацый}%
\end{verbatim}

16
7.5.2 Date in Belarusian

\datebelarusian The macro \datebelarusian is used to reset the macro \today in Belarusian.
7.5.3 Hyphenation patterns

Belarusian hyphenation patterns are automatically activated every time Belarusian language is selected via \selectlanguage{belarusian} or equivalent command. But we need to declare values of \leftyhyphenmin and \rightyhyphenmin; both are set to 2.

As of v.1.2 we removed a definition for \englishhyphenmins. It is not deal of belarusian.ldf.

\providehyphenmins{\CurrentOption}{\tw@	w@}
\providehyphenmins{belarusian}{\tw@	w@}

7.5.4 Extra definitions

The macro \extrasbelarusian performs extra definitions in addition to resetting the caption names and date. The macro \noextrasbelarusian is used to cancel the actions of \extrasbelarusian.

First, we instruct babel to switch font encoding using earlier defined macros \cyrillictext and \latintext.

\addto\extrasbelarusian{\cyrillictext}
\addto\noextrasbelarusian{\latintext}

Second, we specify that the Belarusian group of shorthands should be used.

\addto\extrasbelarusian{\languageshorthands{belarusian}}
\addto\extrasbelarusian{\bbl@activate{}}
\addto\noextrasbelarusian{\bbl@deactivate{}}

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

\addto\extrasbelarusian{\bbl@frenchspacing}
\addto\noextrasbelarusian{\bbl@nonfrenchspacing}

7.6 Alphabetic counters

Do we need to reset \alph and \Alph? They are used in the \LaTeX{} core to define the macros \alph and \Alph, respectively, which type a counter with a corresponding letter of Latin alphabet. We just want to make sure that correct \latinencoding is used instead of \latinencodin to typeset the counter. Starting from v.1.2 we do not reset these macros since all Cyrillic encoding but X2 do have Latin letters. When using the X2 encoding user must himself take care about selecting correct encoding when he switches his keyboard. Our decision is motivated as follows. If selected Cyrillic font is visually different from Latin font, the macro \@alph and \@Alph will produce visually different output from surrounding text if they are used with Belarusian text, which is completely legitimate.

Notice for commented code:
We put \latinencoding in braces to avoid problems with \@alph inside minipages
where \@alph is expanded and we get for example ‘\fontencoding OT1’ (\fontencoding is robust).

Note added on 2013/03/22: {\fontencoding{\latinencoding}\selectfont rises an error with recent version of \microtype package after the \appendix declaration (which resets \thechapter to \@Alph\c@chapter)). Most languages do not reset \@alph and \@Alph macros and only ukrainian and bulgarian add \fontencoding to \@alph and \@Alph.

Since v.1.3 we do not reset \@alph and \@Alph here. Resetting \fontencoding in \@alph and \@Alph causes an error if the package \smartref is loaded and a \sref occurs after the \appendix declaration which resets \thechapter to \@Alph\c@chapter.

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

\Asbuk The macro \asbuk is similar to \alph; it produces lowercase Belarusian letters.
Babel 3.9 has introduced a notion of a language attribute. The `classic` attribute (or its alias `tarask`) changes the default behavior which uses the official Belarusian spelling, and activates an alternative set of captions and date macros.

We don’t want for long internal macros to waste memory. So we declare them to be usable within the preamble only.

### 7.7 Cyrillic math

For compatibility with older Russian packages we could define the `\No` macro. However the Belarusian number sign is now superseded with `\textnumero`. Moreover, it can be found of Belarusian keyboard. Therefore we discard `\No` since v.1.2.

As of version 1.2 the macros `\cyrmath..` are not supported any more. They requires package `textmath` which is not available now. Instead of `\cyrmath..` it is advised to use corresponding `\text..` commands; they do work in math mode.
We also define few math operator names according to Belarusian typesetting traditions. Some math functions in Belarusian math books have names different from English writings. For example, \(\sinh\) in Belarusian is called \(\text{sh}\). Special consideration needs the macro \(\text{th}\) that conflicts with the text symbol \(\text{th}\) defined in Latin 1 encoding:

\[
\begin{align*}
&\def\sh{\mathop{\operator@font sh}\nolimits} \\
&\def\ch{\mathop{\operator@font ch}\nolimits} \\
&\def\tg{\mathop{\operator@font tg}\nolimits} \\
&\def\ctg{\mathop{\operator@font ctg}\nolimits} \\
&\def\arcctg{\mathop{\operator@font arcctg}\nolimits} \\
&\def\cosec{\mathop{\operator@font cosec}\nolimits} \\
\end{align*}
\]

Finally, we define some rare Belarusian mathematical symbols:

\[
\begin{align*}
&\def\Prob{\mathop{\kern\z@\mathsf{P}}\nolimits} \\
&\def\Variance{\mathop{\kern\z@\mathsf{D}}\nolimits} \\
&\if@uni@ode@be \\
&\def\nod{\mathop{\mathrm{н.а.д.}}\nolimits} \\
&\def\nok{\mathop{\mathrm{н.а.к.}}\nolimits} \\
&\def\NOD{\mathop{\mathrm{НАД}}\nolimits} \\
&\def\NOK{\mathop{\mathrm{НАК}}\nolimits} \\
&\def\nad{\mathop{\mathrm{н.а.д.}}\nolimits} \\
&\def\nak{\mathop{\mathrm{н.а.к.}}\nolimits} \\
&\def\NAD{\mathop{\mathrm{НАД}}\nolimits} \\
&\def\NAK{\mathop{\mathrm{НАК}}\nolimits} \\
&\def\Proj{\mathop{\mathrm{Пр}}\nolimits} \\
&\else \\
&\def\nod{\mathop{\textnormal{н.а.д.}}\nolimits} \\
&\def\nok{\mathop{\textnormal{н.а.к.}}\nolimits} \\
&\def\NOD{\mathop{\textnormal{НАД}}\nolimits} \\
&\def\NOK{\mathop{\textnormal{НАК}}\nolimits} \\
&\def\nad{\mathop{\textnormal{н.а.д.}}\nolimits} \\
&\def\nak{\mathop{\textnormal{н.а.к.}}\nolimits} \\
&\def\NAD{\mathop{\textnormal{НАД}}\nolimits} \\
&\def\NAK{\mathop{\textnormal{НАК}}\nolimits} \\
&\def\Proj{\mathop{\textnormal{Пр}}\nolimits} \\
&\fi
\end{align*}
\]
7.8 Final settings

The macro `\ldf@finish` does work needed at the end of each `.ldf` file. This includes resetting the category code of the `@` sign, loading a local configuration file, and preparing the language to be activated at `\begin{document}` time.

\ldf@finish{belarusian}

20pt by default