The Belarusian Language in the babel system

Version 1.3h

Andrew Shadura[*]

Released 2016/04/07

Contents

[*] E-mail: andrew@shadura.me.

1 The Belarusian Language Definition File

The file `belarusianb.ldf`[1] is the source file for the Belarusian Language Definition file `belarusianb.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 `belarusianb.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`, `belarusianb.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 LaTeX or PDFLaTeX, 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, `belarusianb.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. LuaLaTeX and XeLaTeX, being the Unicode-based succeeders of LaTeX, 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, LaTeX community are moving towards eliminating all existing encodings in favor of Unicode, but nowadays one should take care when switching from LaTeX to LuaLaTeX or XeLaTeX since different packages should be loaded for those compilers.

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

[1] The file described in this document has the version number 1.3h and was last revised on 2016/04/07.

2.1 LaTeX

When user's document is compiled with `latex.exe` or `pdflatex.exe`, recommended set of packages includes the `inputenc` and `fontenc` packages. They should be loaded before `babel`, for example,

```
    \usepackage[T1,T2A]{fontenc}
    \usepackage[utf8]{inputenc}
    \usepackage[english,belarusian]{babel}
```

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 `babel` package should go last, and `fontenc` must be the first.

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

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

2.2 LuaLaTeX

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

```
\usepackage[T1,T2A]{fontenc}
\usepackage[lutf8]{luainputenc}
\usepackage[english,belarusian]{babel}
```

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

To invoke Unicode mode, one needs to load the `fontspec` package instead of `luainputenc` and `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 LaTeX distributions:

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

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 up on per font bases; for example

```
\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}
```

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

```
\usepackage{fontspec}
\usepackage[english,main=belarusian,german]{babel}
```

2.3 XeLaTeX

In 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 XeLaTeX works like a 8-bit LaTeX engine:

```
\XeTeXinputencoding ”bytes”
\usepackage[utf8]{inputenc}
\usepackage[T2A]{fontenc}
\usepackage[english,belarusian]{babel}
```

XeTeX 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 tex.stackexchange.com/questions/36188/do-xetex-and-luatex-always-use-unicode.

Unicode mode is set up same way as for LuaLaTeX, however the option Renderer=Basic can be dropped:

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

2.4 Official and ‘classic’ spelling

By default, the official spelling is enabled. The ‘classic’ (Taraškievič) ancient 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,

```
\usepackage[english,belarusian]{babel}
\languageattribute{belarusian}{classic}
```

Setting the classic attribute changes the built-in strings (caption names) and a date format. For example, the references list will be entitled as ‘Спіс літаратуры’ by default and as ‘Сьпіс літаратуры’ if the Belarusian language attribute is set to classic. The same result can be achieved using a modifier as follows:

```
\usepackage[english,belarusian.classic]{babel}
```

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 classic and 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 `\selectlanguage{belarusian}` switches to the Belarusian language, with the following effects:

1. Belarusian hyphenation patterns are made active;
2. `\today` prints the date in Belarusian;
3. the caption names are translated into Belarusian (LaTeX only);
4. emdash typed by the ligature `---` might be 20% shorter when Belarusian is the current language; the result depends on the current encoding; `---` always produce long emdash in LuaTeX and XeTeX 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 languges; a shorter emdash (i.e. `\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 classic in the document preamble as discussed in Sec. 2.4.

Since Belarusian has its own numbering system, `belarusianb.ldf` adds macros `\asbuk{`⟨*counter*⟩`}` and `\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 `\guillemotleft` and `\guillemotright` which work in LaTeX 2ε and PlainTeX.
2. German quotation marks can be entered using the commands `\glqq` and `\grqq` which work in LaTeX 2ε and PlainTeX.

The macro `\Belarusian` is now defined as an alias for `\selectlanguage{belarusian}`, and its “opponent” `\English`, existed in `belarusianb.ldf` 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 `\textcyrillic{`⟨*text*⟩`}` is intended to typeset small chunks of text in Belarusian; it is essentially an alias for `\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 `belarusianb.ldf`

`\glqq`	’’‘	German opening double quote (looks like „).
`\grqq`	’’’	German closing double quote (looks like “).
`\guillemotleft`	’’<	French opening double quote (looks like ≪).
`\guillemotright`	’’>	French closing double quote (looks like ≫).
`\dq`		Original quotes character (’’).
`\babelhyphen{soft}`	’’-	Optional (soft) hyphen sign, similar to \- but allows hyphenation in the rest of the word; equivalent to `\babelhyphen{soft}` in `babel` 3.9.
`\babelhyphen{empty}`	’”’	Similar to ’’- but prints no hyphen sign (used for compound words with hyphen, e.g. `x-’”’y`); equivalent to `\babelhyphen{empty}` in `babel` 3.9.
`\babelhyphen*{nobreak}`	’’~	Compound word mark without a breakpoint, prints hyphen prohibiting hyphenation at the point; equivalent to `\babelhyphen*{nobreak}` in `babel` 3.9.
`\babelhyphen{hard}`	’’=	A compound word mark with a breakpoint, prints hyphen allowing hyphenation in the composing words. equivalent to `\babelhyphen{hard}` in `babel` 3.9.
`\babelhyphen{nobreak}`	’’|	Disables ligature at this position; equivalent to `\babelhyphen{nobreak}` (??) in `babel` 3.9.
`\cyrdash`		Row Cyrillic emdash (does not care spaces around).
`\cdash---`	’’---	Cyrillic emdash in plain text.
`\cdash--~`	’’--~	Cyrillic emdash in compound names (as in `Mendeleev’’--~Klapeiron`).
`\cdash--*`	’’--*	Cyrillic emdash for denoting direct speech.
	’’,	Thin space (allows further hyphenation as in `D.’’,Mendeleev`).

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 \mdqoff and \mdqon for these two macros has been removed from belarusianb.ldf starting from version 1.3 in favour of the macros \shorthandon and \shorthandoff provided in the babel core.

3.2 Math commands

belarusianb.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, \arcctg, \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 ‘a’ matches macro \cyra, and capital Belarusian letter ‘A’ 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 LuaLATEX or XeLATEX are detected.

Some inconsistences of prior versions of belarusianb.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, \Englsih and their aliases \Rus, \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 LuaLaTeX or XeLaTeX is running. If so, we set boolean key `\if@uni@ode` to true. It will be used to eliminate `\cyr...` commands, which were introduced in LaTeX2e to handle various Cyrillic input encoding. With the advent of Unicode LaTeX is moving to universal input encoding, so we consider these `\cyr...` commands as obsolete. They are preserved though for backward compatibility in case if LaTeX or PDFLaTeX 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
  \PackageError{babel}{if@uni@ode already defined.\MessageBreak
    Please contact author of belarusianb.ldf}
  \relax
\fi
\newif\if@uni@ode
\ifdefined\luatexversion \@uni@odetrue \else
\ifdefined\XeTeXrevision \@uni@odetrue \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 LaTeX core, is `OT1`. This can be changed by the `fontenc` package in case of LaTeX and by `fontspec` package in case of LuaLaTeX. It matters weather these packages are loaded before of after `babel`. In the latter case or if these packages are not loaded at all, `belarusianb.ldf` ignores their effect and tries to provide some reasonable settings. In particular, `T2A` will be selected for Belarusian language if LaTeX is running but `EU1` in case of XeLaTeX and `EU2` in case of LuaLaTeX.

\latinencoding 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 belarusianb.ldf since it is overruled in babel.def. For example, after

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

as well as after

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

\latinencoding will be set to T1. After

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

\latinencoding will be 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 XeLaTeX and EU2 for LuaLaTeX, 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 XeLaTeX or EU2 for LuaLaTeX 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 is 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 XeLaTeX), EU2 (default for LuaLaTeX).

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 (LuaLaTeX and XeLaTeX) since they can run in compatibility (8-bit) mode.

```
\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}%
  \def\cdp@elt##1##2##3##4{%
    \sce@a{##1}{OT2}%
    \sce@a{##1}{LCY}%
    \sce@a{##1}{X2}%
    \sce@a{##1}{T2C}%
    \sce@a{##1}{T2B}%
    \sce@a{##1}{T2A}%
    \if@uni@ode
      \sce@a{##1}{EU1}%
      \sce@a{##1}{EU2}%
    \fi}%
  \cdp@list
}
\@setcyrillicencoding
\@onlypreamble\@setcyrillicencoding
\@onlypreamble\sce@a
\@onlypreamble\sce@b
\@onlypreamble\sce@c
```

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.

```
\ifdefined\cyrillicencoding
\else
  \if@uni@ode
    \ifdefined\XeTeXrevision
      \edef\cyrillicencoding{EU1}
    \else\ifdefined\luatexversion
      \edef\cyrillicencoding{EU2}
    \fi\fi
  \else
    \edef\cyrillicencoding{T2A}
  \fi
  \PackageWarning{babel}%
    {No Cyrillic font encoding has been loaded so far.\MessageBreak
     A font encoding should be declared before babel.\MessageBreak
     Default `\cyrillicencoding' encoding will be loaded
    }%
  \lowercase\expandafter{\expandafter\input\cyrillicencoding enc.def\relax}%
```

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.

```
  \AtBeginDocument{\@setcyrillicencoding}
\fi
```

`\Belarusian`
`\cyr`
`\cyrillictext`

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 `\cyrillictext` and its alias `\cyr` but remove another alias `\Rus`; these macros are intended for use within `babel` macros and do not perform complete switch of the language. In particular, they to do no switch captions and the name of current language stored in the macro `\languagename`. This inconsistency might break some assumptions embedded into `babel`'s. For example, the `\iflanguage` macro will fail.

Second, `\cyrillictext` does not activate shorthands, so that ”<, ”>, ”‘, ”’, ”---, e.t.c. will not work.

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

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

```
\DeclareRobustCommand{\Belarusian}{\selectlanguage{\belarusian}}
\DeclareRobustCommand{\cyrillictext}{%
  \fontencoding\cyrillicencoding\selectfont
  \let\encodingdefault\cyrillicencoding
  \expandafter\set@hyphenmins\belarusianhyphenmins
  \language\l@belarusian}%
\let\cyr\cyrillictext
```

Starting from version v.1.2 we remove the `\English` macro and its aliases. We believe that reasonable place for defining these macros would be `englishb.ldf`. Note also that the macro `\English` and its alias `\Eng` are absent in `belarusianb.ldf`'s counterpart in the package `polyglossia`, analog of `babel` for XeLaTeX.

```
%%\DeclareRobustCommand{\English}{%
%%  \fontencoding\latinencoding\selectfont
%%  \let\encodingdefault\latinencoding
%%  \expandafter\set@hyphenmins\englishhyphenmins
%%  \language\l@english}%
%%\let\Eng\English
```

Next part of code should be moved to `X2enc.def`, `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 `\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`).

```
\expandafter\ifx\csname T@X2\endcsname\relax\else
  \DeclareTextSymbolDefault{\AA}{OT1}
  \DeclareTextSymbolDefault{\aa}{OT1}
  \DeclareTextCommand{\aa}{OT1}{\r a}
  \DeclareTextCommand{\AA}{OT1}{\r A}
\fi
```

The macro `\cyrillictext` switches current (e.g., Latin) font encoding to a Cyrillic font encoding stored in `\cyrillicencoding`. The macro `\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.

```
%\DeclareRobustCommand{\latintext}{%
%   \fontencoding{\latinencoding}\selectfont
%   \def\encodingdefault{\latinencoding}}
%\let\lat\latintext
```

`\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 `\foreignlangauge{belarusian}`.

```
\DeclareTextFontCommand{\textcyrillic}{\cyrillictext}
```

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`.

Should we wrap this chunk into At Begin document?

```
\@ifpackageloaded{inputenc}{%
 \if@uni@ode
   \PackageWarning{babel}{inputenc should not be used with LuaTeX or XeTeX}
 \fi
}{%
  %\def\reserved@a{LWN}%
  %\ifx\reserved@a\cyrillicencoding\else
    \def\reserved@a{OT2}%
    \ifx\reserved@a\cyrillicencoding\else
       \def\reserved@a{EU1}%
       \ifx\reserved@a\cyrillicencoding\else
          \def\reserved@a{EU2}%
          \ifx\reserved@a\cyrillicencoding\else
             \PackageWarning{babel}%
                {No input encoding specified for Belarusian language}
    \fi\fi\fi
  %\fi
}
```

7.4 Shorthands

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

```
\initiate@active@char{”}
```

\mdqon
\mdqoff

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

```
%\def\mdqon{\bbl@activate{”}}
%\def\mdqoff{\bbl@deactivate{”}}
```

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

\dq

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 ‘””’.

```
\begingroup \catcode‘\”12
\def\reserved@a{\endgroup
  \def\@SS{\mathchar”7019 }
  \def\dq{”}}
\reserved@a
```

7.4.1 Quotes

We set ”‘ and ”’ as shorthands for \quotedblbase 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 \quotedblbase 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 \guillemotrigh. 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 \guillemotrigh.

```
\declare@shorthand{belarusian}{”‘}{\quotedblbase}
\declare@shorthand{belarusian}{”’}{\textquotedblleft}
\declare@shorthand{belarusian}{”<}{\guillemotleft}
\declare@shorthand{belarusian}{”>}{\guillemotright}
```

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

```
\declare@shorthand{belarusian}{’”’}{\hskip\z@skip}
\declare@shorthand{belarusian}{”~}{\textormath{\leavevmode\hbox{-}}{-}}
\declare@shorthand{belarusian}{”=}{\nobreak-\hskip\z@skip}
\declare@shorthand{belarusian}{”|}{%
  \textormath{\nobreak\discretionary{-}{}{\kern.03em}\allowhyphens}{}}
```

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 emdash, otherwise a hyphen sign. Therefor TEX looks for the next token after the first '-', writes its meaning to `\belarusian@sh@next` and finally call for `\belarusian@sh@tmp`.

```
\declare@shorthand{belarusian}{"-}{%
  \def\belarusian@sh@tmp{%
    \if\belarusian@sh@next-\expandafter\belarusian@sh@emdash
    \else\expandafter\belarusian@sh@hyphen\fi}%
  \futurelet\belarusian@sh@next\belarusian@sh@tmp}
```

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.

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

`\cdash` In its turn, `\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`.

```
\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}
```

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

`\@Acdash` is invoked by `"---`. It types Cyrillic emdash to be used inside a text and puts an unbreakable thin space before the dash if a *space* is placed before `"---` in the source file; can be used after display maths formulae, formatted lists, enumerations, etc.

```
\def\@Acdash{\ifdim\lastskip>\z@\unskip\nobreak\hskip.2em\fi
  \cyrdash\hskip.2em\ignorespaces}%
```

`\@Bcdash` is invoked by `"--~`. It types Cyrillic emdash in compound names (like Mendeleev–Klapeiron); requires no space characters around and adds extra space after the dash.

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

`\@Ccdash` is invoked by `"--*`. It denotes direct speech and adds small space after the dash.

```
\def\@Ccdash{\leavevmode
 \nobreak\cyrdash\nobreak\hskip.35em\ignorespaces}%
```

`\cyrdash` 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.

\textthreequartersemdash=!–!

```
%\ifx\cyrdash\undefined
%  \def\cyrdash{\hbox to.8em{--\hss--}}
%  \PackageInfo{babel}{\string\cyrdash\space is defined}
%\fi
\ProvideTextCommandDefault{\cyrdash}{\iflanguage{belarusian}%
   {\hbox to.8em{--\hss--}}{\textemdash}}
```

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\hskip.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 LaTeX. 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 LaTeX is running and `utf8` input encoding is not declared.

```
\if@uni@ode
  %\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\enclname{укл.}%
  \def\ccname{зых.}%
  \def\headtoname{вх.}%
  \def\pagename{с.}%              [letter]
  \def\seename{гл.}%
  \def\alsoname{гл.\ таксама}%
  \def\proofname{Доказ}%  [amsthm]
  \def\glossaryname{Слоўнік тэрмінаў}%
  \def\acronymname{Абрэвіятуры}%  [glossaries] {Acronyms}
  \def\lstlistingname{Лістынг}%    [listings]  (the environment) {Listing}
  \def\lstlistlistingname{Лістынгі}% [listings]  (the ’List of’) {Listings}
  \def\notesname{Нататкі}%         [endnotes] {Notes}
}%
```

Additional definitions for the package `nomencl`:

```
%% ========================================
%% nomencl
\ifdefined\nomname
  \addto\captionsbelarusian{%
      \def\nomname{Азначэнні}%
      \def\eqdeclaration#1{, гл.\nobreakspace(#1)}%
      \def\pagedeclaration#1{, стар.\nobreakspace#1}%
  }%
\fi
```

Additional captions for the `revtex` class.

```
%% ========================================
%% RevTeX4 & RevTeX4-1
%%\@ifclassloaded{revtex4-1}
%%{<true code>}
%%{<false coode>}%
%\@ifclassloaded{revtex4-1}
%{%
\ifnum\@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
  \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.

```
\addto\captionsbelarusian@tarask{%
  \def\prefacename{Прадмова}%   [babel]
  \def\refname{Сьпіс 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{Зьмест}%
  \let\tocname=\contentsname
  \def\listfigurename{Сьпіс ілюстрацый}%
  \def\listtablename{Сьпіс табліц}%
  \def\indexname{Прадметны паказальнік}%
  \def\authorname{Паказальнік імён}%
  \def\figurename{Рыс.}%
  \def\tablename{Табліца}%
  \def\partname{Частка}%
  \def\enclname{укл.}%
  \def\ccname{зых.}%
  \def\headtoname{вх.}%
  \def\pagename{с.}%               [letter]
  \def\seename{гл.}%
  \def\alsoname{гл.\ таксама}%
  \def\proofname{Доказ}%  [amsthm]
  \def\glossaryname{Слоўнік тэрмінаў}%
  \def\acronymname{Абрэвіятуры}%  [glossaries] {Acronyms}
  \def\lstlistingname{Лістынг}%    [listings]  (the environment) {Listing}
  \def\lstlistlistingname{Лістынгі}% [listings]  (the ''List of'') {Listings}
  \def\nomname{Азначэньні}%
  \def\notesname{Нататкі}%         [endnotes] {Notes}
}
%% ========================================
%% RevTeX4 & RevTeX4-1
%\@ifclassloaded{revtex4-1}
%{%
\ifnum\@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
  \addto\captionsbelarusian@tarask{%
  \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 вэрсыя }%{Revised }%
  \def\Accepted@name{Прынята }%{Accepted }%
  \def\Published@name{Апублікавана }%{Published }%
  }%
%}{}%
```

```
  \fi
  %% =========================================
\else
  %\def\captionsbelarusian@official{%
  \addto\captionsbelarusian{%
    \def\prefacename{\CYRP\cyrr\cyra\cyrd\cyrm\cyro\cyrv\cyra}%   [babel]
    \def\refname{\CYRS\cyrp\cyrii\cyrs \cyrl\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrr\cyrery}%
ticle]
    \def\abstractname{\CYRA\cyrn\cyra\cyrt\cyra\cyrc\cyrery\cyrya}%    [only ar-
ticle, report]
    \def\bibname{\CYRL\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrr\cyra}%        [only book, re-
port]
    \def\chaptername{\CYRG\cyrl\cyra\cyrv\cyra}%         [only book, report]
    \def\appendixname{\CYRD\cyra\cyrd\cyra\cyrt\cyra\cyrk}%
    \def\contentsname{\CYRZ\cyrm\cyre\cyrs\cyrt}%
    \let\tocname=\contentsname
    \def\listfigurename{\CYRS\cyrp\cyrii\cyrs \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrer
    \def\listtablename{\CYRS\cyrp\cyrii\cyrs \cyrt\cyra\cyrb\cyrl\cyrii\cyrc}%
    \def\indexname{\CYRP\cyrr\cyra\cyrd\cyrm\cyre\cyrt\cyrn\cyrery \cyrp\cyra\cyrk\cyra\cyrz\c
    \def\authorname{\CYRP\cyra\cyrk\cyra\cyrz\cyra\cyrl\cyrsftsn\cyrn\cyrii\cyrk \cyrii\cyrm\c
    \def\figurename{\CYRR\cyrery\cyrs.}%
    \def\tablename{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyra}%
    \def\partname{\CYRCH\cyra\cyrs\cyrt\cyrk\cyra}%
    \def\enclname{\cyru\cyrk\cyrl.}%
    \def\ccname{\cyrz\cyrery\cyrh.}%
    \def\headtoname{\cyrv\cyrh.}%
    \def\pagename{\cyrs.}%               [letter]
    \def\seename{\cyrg\cyrl.}%
    \def\alsoname{\cyrg\cyrl.\ \cyrt\cyra\cyrk\cyrs\cyra\cyrm\cyra}%
    \def\proofname{\CYRD\cyro\cyrk\cyra\cyrz}%  [amsthm]
    \def\glossaryname{\CYRS\cyrl\cyro\cyrushrt\cyrn\cyrii\cyrk \cyrt\cyrerev\cyrr\cyrm\cyrii\c
    \def\acronymname{\CYRA\cyrb\cyrr\cyrerev\cyrv\cyrii\cyrya\cyrt\cyru\cyrr\cyrery}%  [glos-
saries] {Acronyms}
    \def\lstlistingname{\CYRL\cyrii\cyrs\cyrt\cyrery\cyrn\cyrg}%    [list-
ings]  (the environment) {Listing}
    \def\lstlistlistingname{\CYRL\cyrii\cyrs\cyrt\cyrery\cyrn\cyrg\cyrii}% [list-
ings]  (the ''List of'') {Listings}
    \def\nomname{\CYRA\cyrz\cyrn\cyra\cyrch\cyrerev\cyrn\cyrn\cyrii}%
    \def\notesname{\CYRN\cyra\cyrt\cyra\cyrt\cyrk\cyrii}%         [endnotes] {Notes}
  }%
  %% =========================================
  %% RevTeX4 & RevTeX4-1
  \ifnum \@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
    \addto\captionsbelarusian{%
    \def\lofname{\listfigurename}
    \def\lotname{\listtablename}
    \def\figuresname{\CYRR\cyrery\cyrs\cyru\cyrn\cyrk\cyrii}%{Figures}%
    \def\tablesname{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyrery}%{Tables}%
    \def\appendixesname{\CYRD\cyra\cyrd\cyra\cyrt\cyrk\cyrii}%{Appendixes}%
    \def\acknowledgmentsname{\CYRP\cyra\cyrd\cyrz\cyrya\cyrk\cyrii}%{Acknowledgments}
    \def\andname{\cyri}%{and}
    \def\@pacs@name{PACS \cyrk\cyro\cyrd\cyrery: }%{PACS numbers: }%
    \def\@keys@name{\CYRK\cyrl\cyryu\cyrch\cyra\cyrv\cyrery\cyrya \cyrs\cyrl\cyro\cyrv\cyrery:
words: }%
```

```
    \def\Dated@name{\CYRD\cyra\cyrt\cyra: }%{Dated: }%
    \def\Received@name{\CYRA\cyrt\cyrr\cyrery\cyrm\cyra\cyrn\cyra }%{Received }%
    \def\Revised@name{\CYRV\cyrery\cyrp\cyrr\cyra\cyrushrt\cyrl\cyre\cyrn\cyra\cyrya \cyrv\cyr
vised }%
    \def\Accepted@name{\CYRP\cyrr\cyrery\cyrn\cyrya\cyrt\cyra }%{Accepted }%
    \def\Published@name{\CYRA\cyrp\cyru\cyrb\cyrl\cyrii\cyrk\cyra\cyrv\cyra\cyrn\cyra }%{Published }%
    }%
  \fi
  %% ========================================
  \addto\captionsbelarusian@tarask{%
    \def\prefacename{\CYRP\cyrr\cyra\cyrd\cyrm\cyro\cyrv\cyra}%   [babel]
    \def\refname{\CYRS\cyrsftsn\cyrp\cyrii\cyrs \cyrl\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrr
ticle]
    \def\abstractname{\CYRA\cyrn\cyra\cyrt\cyra\cyrc\cyrery\cyrya}%    [only article, report]
    \def\bibname{\CYRL\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrr\cyra}%        [only book, report]
    \def\chaptername{\CYRG\cyrl\cyra\cyrv\cyra}%         [only book, report]
    \def\appendixname{\CYRD\cyra\cyrd\cyra\cyrt\cyra\cyrk}%
    \def\contentsname{\CYRZ\cyrsftsn\cyrm\cyre\cyrs\cyrt}%
    \let\tocname=\contentsname
    \def\listfigurename{\CYRS\cyrsftsn\cyrp\cyrii\cyrs \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\c
    \def\listtablename{\CYRS\cyrsftsn\cyrp\cyrii\cyrs \cyrt\cyra\cyrb\cyrl\cyrii\cyrc}%
    \def\indexname{\CYRP\cyrr\cyra\cyrd\cyrm\cyre\cyrt\cyrn\cyrery \cyrp\cyra\cyrk\cyra\cyrz\c
    \def\authorname{\CYRP\cyra\cyrk\cyra\cyrz\cyra\cyrl\cyrsftsn\cyrn\cyrii\cyrk \cyrii\cyrm\c
    \def\figurename{\CYRR\cyrery\cyrs.}%
    \def\tablename{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyra}%
    \def\partname{\CYRCH\cyra\cyrs\cyrt\cyrk\cyra}%
    \def\enclname{\cyru\cyrk\cyrl.}%
    \def\ccname{\cyrz\cyrery\cyrh.}%
    \def\headtoname{\cyrv\cyrh.}%
    \def\pagename{\cyrs.}%               [letter]
    \def\seename{\cyrg\cyrl.}%
    \def\alsoname{\cyrg\cyrl.\ \cyrt\cyra\cyrk\cyrs\cyra\cyrm\cyra}%
    \def\proofname{\CYRD\cyro\cyrk\cyra\cyrz}%  [amsthm]
    \def\glossaryname{\CYRS\cyrl\cyro\cyrushrt\cyrn\cyrii\cyrk \cyrt\cyrerev\cyrr\cyrm\cyrii\c
    \def\acronymname{\CYRA\cyrb\cyrr\cyrerev\cyrv\cyrii\cyrya\cyrt\cyru\cyrr\cyrery}%  [glossaries] {Acronyms}
    \def\lstlistingname{\CYRL\cyrii\cyrs\cyrt\cyrery\cyrn\cyrg}%    [listings]  (the environment) {Listing}
    \def\lstlistlistingname{\CYRL\cyrii\cyrs\cyrt\cyrery\cyrn\cyrg\cyrii}% [listings]  (the ”List of”) {Listings}
    \def\nomname{\CYRA\cyrz\cyrn\cyra\cyrch\cyrerev\cyrn\cyrsftsn\cyrn\cyrii}%
    \def\notesname{\CYRN\cyra\cyrt\cyra\cyrt\cyrk\cyrii}%         [endnotes] {Notes}
  }%
  %% ========================================
  %% RevTeX4 & RevTeX4-1
  \ifnum \@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
    \addto\captionsbelarusian@tarask{%
    \def\lofname{\listfigurename}
    \def\lotname{\listtablename}
    \def\figuresname{\CYRR\cyrery\cyrs\cyru\cyrn\cyrk\cyrii}%{Figures}%
    \def\tablesname{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyrery}%{Tables}%
    \def\appendixesname{\CYRD\cyra\cyrd\cyra\cyrt\cyrk\cyrii}%{Appendixes}%
    \def\acknowledgmentsname{\CYRP\cyra\cyrd\cyrz\cyrya\cyrk\cyrii}%{Acknowledgments}
    \def\andname{\cyri}%{and}
    \def\@pacs@name{PACS \cyrk\cyro\cyrd\cyrery: }%{PACS numbers: }%
    \def\@keys@name{\CYRK\cyrl\cyryu\cyrch\cyra\cyrv\cyrery\cyrya \cyrs\cyrl\cyro\cyrv\cyrery
words: }%
    \def\Dated@name{\CYRD\cyra\cyrt\cyra: }%{Dated: }%
    \def\Received@name{\CYRA\cyrt\cyrr\cyrery\cyrm\cyra\cyrn\cyra }%{Received }%
    \def\Revised@name{\CYRV\cyrery\cyrp\cyrr\cyra\cyrushrt\cyrl\cyre\cyrn\cyra\cyrya \cyrv\cyr
vised }%
    \def\Accepted@name{\CYRP\cyrr\cyrery\cyrn\cyrya\cyrt\cyra }%{Accepted }%
    \def\Published@name{\CYRA\cyrp\cyru\cyrb\cyrl\cyrii\cyrk\cyra\cyrv\cyra\cyrn\cyra }%{Published }%
    }%
  \fi
  %% ========================================
\fi
```

7.5.2 Date in Belarusian

\datebelarusian

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

```
\if@uni@ode
  \addto\datebelarusian{%
   \def\today{\number\day~\ifcase\month\or
    студзеня\or
    лютага\or
    сакавіка\or
    красавіка\or
    мая\or
    чэрвеня\or
    ліпеня\or
    жніўня\or
    верасня\or
    кастрычніка\or
    лістапада\or
    снежня\fi
    \space \number\year~г.}}
  \def\datebelarusian@tarask{%
   \def\today{\number\day~\ifcase\month\or%
    студзеня\or
    лютага\or
    сакавіка\or
    красавіка\or
    траўня\or
    чэрвеня\or
    ліпеня\or
    жніўня\or
    верасьня\or
    кастрычніка\or
    лістапада\or
    сьнежня\fi%
    \space \number\year~г.}}
```

```
\else
  \def\datebelarusian{%
   \def\today{\number\day~\ifcase\month\or
    \cyrs\cyrt\cyru\cyrd\cyrz\cyre\cyrn\cyrya\or
    \cyrl\cyryu\cyrt\cyra\cyrg\cyra\or
    \cyrs\cyra\cyrk\cyra\cyrv\cyrii\cyrk\cyra\or
    \cyrk\cyrr\cyra\cyrs\cyra\cyrv\cyrii\cyrk\cyra\or
    \cyrm\cyra\cyrya\or
    \cyrch\cyrerev\cyrr\cyrv\cyre\cyrn\cyrya\or
    \cyrl\cyrii\cyrp\cyre\cyrn\cyrya\or
    \cyrzh\cyrn\cyrii\cyrushrt\cyrn\cyrya\or
    \cyrv\cyre\cyrr\cyra\cyrs\cyrn\cyrya\or
    \cyrk\cyra\cyrs\cyrt\cyrr\cyrery\cyrch\cyrn\cyrii\cyrk\cyra\or
    \cyrl\cyrii\cyrs\cyrt\cyra\cyrp\cyra\cyrd\cyra\or
    \cyrs\cyrn\cyre\cyrzh\cyrn\cyrya\fi
    \space \number\year~\cyrg.}}
  \def\datebelarusian@tarask{%
   \def\today{\number\day~\ifcase\month\or
    \cyrs\cyrt\cyru\cyrd\cyrz\cyre\cyrn\cyrya\or
    \cyrl\cyryu\cyrt\cyra\cyrg\cyra\or
    \cyrs\cyra\cyrk\cyra\cyrv\cyrii\cyrk\cyra\or
    \cyrk\cyrr\cyra\cyrs\cyra\cyrv\cyrii\cyrk\cyra\or
    \cyrt\cyrr\cyra\cyrushrt\cyrn\cyrya\or
    \cyrch\cyrerev\cyrr\cyrv\cyre\cyrn\cyrya\or
    \cyrl\cyrii\cyrp\cyre\cyrn\cyrya\or
    \cyrzh\cyrn\cyrii\cyrushrt\cyrn\cyrya\or
    \cyrv\cyre\cyrr\cyra\cyrs\cyrsftsn\cyrn\cyrya\or
    \cyrk\cyra\cyrs\cyrt\cyrr\cyrery\cyrch\cyrn\cyrii\cyrk\cyra\or
    \cyrl\cyrii\cyrs\cyrt\cyra\cyrp\cyra\cyrd\cyra\or
    \cyrs\cyrsftsn\cyrn\cyre\cyrzh\cyrn\cyrya\fi%
    \space \number\year~\cyrg.}}
\fi
```

7.5.3 Hyphenation patterns

Belarusian hyphenation patterns are automatically activated every time Belarusian language is selected via \selectlanguage, \foreignlanguage or equivalent command. But we need to declare values of \lefthyphenmin and \righthyphenmin; both are set to 2.

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

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

7.5.4 Extra definitions

\extrasbelarusian
\noextrasbelarusian

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 `\latinencoding` 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 (e.g., footnotes 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`.

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

We add new enumeration style for Belarusian manuscripts with Cyrillic letters.

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

```
\def\Asbuk#1{\expandafter\belarusian@Alph\csname c@#1\endcsname}
\if@uni@ode
  \def\belarusian@Alph#1{\ifcase#1\or
    А\or Б\or В\or Г\or Д\or Е\or Ж\or
    З\or І\or К\or Л\or М\or Н\or О\or
    П\or Р\or С\or Т\or У\or Ф\or Х\or
    Ц\or Ч\or Ш\or Э\or Ю\or Я\else\@ctrerr\fi}
\else
  \def\belarusian@Alph#1{\ifcase#1\or
    \CYRA\or\CYRB\or\CYRV\or\CYRG\or\CYRD\or\CYRE\or\CYRZH\or
    \CYRZ\or\CYRII\or\CYRK\or\CYRL\or\CYRM\or\CYRN\or\CYRO\or
    \CYRP\or\CYRR\or\CYRS\or\CYRT\or\CYRU\or\CYRF\or\CYRH\or
    \CYRC\or\CYRCH\or\CYRSH\or\CYREREV\or\CYRYU\or
    \CYRYA\else\@ctrerr\fi}
\fi
```

\asbuk The macro \asbuk is similar to \alph; it produces lowercase Belarusian letters.

```
\def\asbuk#1{\expandafter\belarusian@alph\csname c@#1\endcsname}
\if@uni@ode
  \def\belarusian@alph#1{\ifcase#1\or
    а\or б\or в\or г\or д\or е\or ж\or
    з\or і\or к\or л\or м\or н\or о\or
    п\or р\or с\or т\or у\or ф\or х\or
    ц\or ч\or ш\or э\or ю\or я\else\@ctrerr\fi}
\else
  \def\belarusian@alph#1{\ifcase#1\or
    \cyra\or\cyrb\or\cyrv\or\cyrg\or\cyrd\or\cyre\or\cyrzh\or
    \cyrz\or\cyrii\or\cyrk\or\cyrl\or\cyrm\or\cyrn\or\cyro\or
    \cyrp\or\cyrr\or\cyrs\or\cyrt\or\cyru\or\cyrf\or\cyrh\or
    \cyrc\or\cyrch\or\cyrsh\or\cyrerev\or\cyryu\or
    \cyrya\else\@ctrerr\fi}
\fi
```

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.

```
\bbl@declare@ttribute{belarusian}{classic}{%
 \PackageInfo{babel}{Belarusian attribute set to classic}%
 \let\captionsbelarusian=\captionsbelarusian@tarask
 \let\datebelarusian=\datebelarusian@tarask
}
\bbl@declare@ttribute{belarusian}{tarask}{%
 \PackageInfo{babel}{Belarusian attribute set to tarask}%
 \let\captionsbelarusian=\captionsbelarusian@tarask
 \let\datebelarusian=\datebelarusian@tarask
}
```

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

```
\@onlypreamble\captionsbelarusian@tarask
```

```
\@onlypreamble\datebelarusian@tarask
```

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.

```
%\DeclareRobustCommand{\No}{%
%   \ifmmode{\nfss@text{\textnumero}}\else\textnumero\fi}
```

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.

```
%\RequirePackage{textmath}
%  \@ifundefined{sym\cyrillicencoding letters}{}{%
%  \SetSymbolFont{\cyrillicencoding letters}{bold}\cyrillicencoding
%    \rmdefault\bfdefault\updefault
%  \DeclareSymbolFontAlphabet\cyrmathrm{\cyrillicencoding letters}
```

And we need few commands to switch to different variants.

```
%\DeclareMathAlphabet\cyrmathbf\cyrillicencoding
%  \rmdefault\bfdefault\updefault
%\DeclareMathAlphabet\cyrmathsf\cyrillicencoding
%  \sfdefault\mddefault\updefault
%\DeclareMathAlphabet\cyrmathit\cyrillicencoding
%  \rmdefault\mddefault\itdefault
%\DeclareMathAlphabet\cyrmathtt\cyrillicencoding
%  \ttdefault\mddefault\updefault
%
%\SetMathAlphabet\cyrmathsf{bold}\cyrillicencoding
%  \sfdefault\bfdefault\updefault
%\SetMathAlphabet\cyrmathit{bold}\cyrillicencoding
%  \rmdefault\bfdefault\itdefault
%}
```

\sh
\ch
\tg
\ctg
\arctg
\arcctg
\th
\cth
\cosec

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 sh. Special consideration needs the macro \th that conflicts with the text symbol \th defined in Latin 1 encoding:

```
\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\arctg{\mathop{\operator@font arctg}\nolimits}
\def\arcctg{\mathop{\operator@font arcctg}\nolimits}
\addto\extrasbelarusian{%
  \babel@save{\th}%
  \let\ltx@th\th
  \def\th{\textormath{\ltx@th}%
                     {\mathop{\operator@font th}\nolimits}}%
  }
```

```
\def\cth{\mathop{\operator@font cth}\nolimits}
\def\cosec{\mathop{\operator@font cosec}\nolimits}
```

\Prob \Variance \nod \nok \NOD \NOK \Proj

Finally, we define some rare Belarusian mathematical symbols:

```
\def\Prob{\mathop{\kern\z@\mathsf{P}}\nolimits}
\def\Variance{\mathop{\kern\z@\mathsf{D}}\nolimits}
\if@uni@ode
  \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{\cyrmathrm{\cyrn.\cyra.\cyrd.}}\nolimits}
  \def\nok{\mathop{\cyrmathrm{\cyrn.\cyra.\cyrk.}}\nolimits}
  \def\NOD{\mathop{\cyrmathrm{\CYRN\CYRA\CYRD}}\nolimits}
  \def\NOK{\mathop{\cyrmathrm{\CYRN\CYRA\CYRK}}\nolimits}
  \def\nad{\mathop{\cyrmathrm{\cyrn.\cyra.\cyrd.}}\nolimits}
  \def\nak{\mathop{\cyrmathrm{\cyrn.\cyra.\cyrk.}}\nolimits}
  \def\NAD{\mathop{\cyrmathrm{\CYRN\CYRA\CYRD}}\nolimits}
  \def\NAK{\mathop{\cyrmathrm{\CYRN\CYRA\CYRK}}\nolimits}
  \def\Proj{\mathop{\cyrmathrm{\CYRP\cyrr}}\nolimits}
\fi
```

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