

The Belarusian Language in the babel system

Version 1.3h

Andrej Shadura*

Released 2016/04/07

Contents

1	The Belarusian Language Definition File	2
2	Usage	2
2.1	L ^A T _E X	3
2.2	LuaL ^A T _E X	3
2.3	XeL ^A T _E X	4
2.4	Official and ‘classic’ spelling	4
3	User’s commands	5
3.1	Active character	6
3.2	Math commands	7
4	T_EXnical 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`¹ 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 `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, `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. `LuaLATEX` and `XeLATEX`, being the Unicode-based successors 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.

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

2.1 L^AT_EX

When user's document is compiled with `latex` or `pdflatex`, 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 LuaL^AT_EX

If Unicode fonts are not available, LuaL^AT_EX can run in compatibility (8-bit) mode to use same font as L^AT_EX does. However the package `inputenc` does not work with LuaL^AT_EX 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 LuaL^AT_EX. The 8-bit mode is invoked by the following sequence of packages:

```
\usepackage[T1,T2A]{fontenc}
\usepackage[lut8]{luainputenc}
\usepackage[english,belarusian]{babel}
```

The order of the packages is crucial for LuaL^AT_EX 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 LuaL^AT_EX 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 roman, sans-serif and monospaced types. The following example shows how to load Computer Modern Unicode (CMU) fonts, which is a part of all modern L^AT_EX 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 roman 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 Xe \LaTeX

In Xe \LaTeX , 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 Xe \LaTeX works like a 8-bit \LaTeX engine:

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

Xe \TeX 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 Lua \LaTeX , 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 ‘Сніс літаратуры’ by default and as ‘Сьніс 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 (L^AT_EX 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 Lua_T_EX and Xe_T_EX 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. `\cyrdash`) can be typeset in any language using special macros enlisted in table 1.

By default, the official spelling is used for built-in 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, `belarusian.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 L^AT_EX 2_ε and Plain_T_EX.
2. German quotation marks can be entered using the commands `\glqq` and `\grqq` which work in L^AT_EX 2_ε and Plain_T_EX.

The macro `\Belarusian` is now defined as an alias for `\selectlanguage{belarusian}`, and its “opponent” `\English`, existed in `belarusian.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 `belarusian.ldf`

<code>\glqq</code>	”“	German opening double quote (looks like „).
<code>\grqq</code>	””	German closing double quote (looks like “).
<code>\guillemotleft</code>	”<	French opening double quote (looks like <<).
<code>\guillemotright</code>	”>	French closing double quote (looks like >>).
<code>\dq</code>		Original quotes character (”).
<code>\babelhyphen{soft}</code>	”-	Optional (soft) hyphen sign, similar to \- but allows hyphenation in the rest of the word; equivalent to <code>\babelhyphen{soft}</code> in <code>babel 3.9</code> .
<code>\babelhyphen{empty}</code>	””	Similar to ”- but prints no hyphen sign (used for compound words with hyphen, e.g. x-”y); equivalent to <code>\babelhyphen{empty}</code> in <code>babel 3.9</code> .
<code>\babelhyphen*{nobreak}</code>	”~	Compound word mark without a breakpoint, prints hyphen prohibiting hyphenation at the point; equivalent to <code>\babelhyphen*{nobreak}</code> in <code>babel 3.9</code> .
<code>\babelhyphen{hard}</code>	”=	A compound word mark with a breakpoint, prints hyphen allowing hyphenation in the composing words. equivalent to <code>\babelhyphen{hard}</code> in <code>babel 3.9</code> .
<code>\babelhyphen{nobreak}</code>	”	Disables ligature at this position; equivalent to <code>\babelhyphen{nobreak}</code> (??) in <code>babel 3.9</code> .
<code>\cyrdash</code>		Row Cyrillic emdash (does not care spaces around).
<code>\cdash---</code>	”---	Cyrillic emdash in plain text.
<code>\cdash--~</code>	”--~	Cyrillic emdash in compound names (as in Mendeleev”--~Klapeiron).
<code>\cdash--*</code>	”--*	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 `belarusian.ldf` starting from version 1.3 in favour of the macros `\shorthandon` and `\shorthandoff` 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`, `\arcttg`, `\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 T_EXnical 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 L^AT_EX, but they are bypassed if LuaL^AT_EX or XeL^AT_EX are detected.

Some inconsistencies of prior versions of `belarusian.ldf` was also overcome 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.

```
1 \LdfInit{belarusian}{captionsbelarusian}
```

First, we check if `Lua \TeX` or `Xe \TeX` 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 `L \TeX 2e` 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 `PDF \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.

```
2 \ifdefined\if@uni@ode
3   \PackageError{babel}{if@uni@ode already defined.\MessageBreak
4     Please contact author of belarusian.ldf}
5   \relax
6 \fi
7 \newif\if@uni@ode
8 \ifdefined\luatexversion \@uni@odetrue \else
9 \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.

```
10 \ifx\l@belarusian\undefined
11   \@nopatterns{Belarusian}
12   \adddialect\l@belarusian0
13 \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 `fontspec` package in case of `Lua \TeX` . It matters whether these packages are loaded before or 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 `Xe \TeX` and `EU2` in case of `Lua \TeX` .

`\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 `belarusian.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 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 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 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.

```
14 \def\@setcyrillicencoding{%
15   \def\sce@a##1##2{%
16     \edef\sce@b{##1}%
17     \edef\sce@c{##2}%
18     \ifx\sce@b\sce@c
19       \let\cyrillicencoding\sce@c
20     \fi}%
```

```

21 \def\cdp@elt##1##2##3##4{%
22   \sca{##1}{OT2}%
23   \sca{##1}{LCY}%
24   \sca{##1}{X2}%
25   \sca{##1}{T2C}%
26   \sca{##1}{T2B}%
27   \sca{##1}{T2A}%
28   \if@uni@ode
29     \sca{##1}{EU1}%
30     \sca{##1}{EU2}%
31   \fi}%
32 \cdp@list
33 }
34 \@setcyrillicencoding
35 \@onlypreamble\@setcyrillicencoding
36 \@onlypreamble\sca
37 \@onlypreamble\sca@b
38 \@onlypreamble\sca@c

```

The last lines are to free the memory occupied by the macros `\@setcyrillicencoding` and `\sca@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.

```

39 \ifdefined\cyrillicencoding
40 \else
41   \if@uni@ode
42     \ifdefined\XeTeXrevision
43       \edef\cyrillicencoding{EU1}
44     \else\ifdefined\luatexversion
45       \edef\cyrillicencoding{EU2}
46     \fi\fi
47   \else
48     \edef\cyrillicencoding{T2A}
49   \fi
50   \PackageWarning{babel}%
51   {No Cyrillic font encoding has been loaded so far.\MessageBreak
52    A font encoding should be declared before babel.\MessageBreak
53    Default ‘\cyrillicencoding’ encoding will be loaded
54   }%
55   \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.

```

56 \AtBeginDocument{\@setcyrillicencoding}
57 \fi

```

`\Belarusian` For the sake of backward compatibility we keep the macro `\Belarusian` but redefine its meaning; now `\Belarusian` is simply an alias for `\selectlanguage{belarusian}`.
`\cyr`
`\cyrillictext`

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 do not switch captions and the name of current language stored in the macro `\language`. 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.

```
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 `\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 `belarusian.ldf`'s counterpart in the package `polyglossia`, analog of `babel` for Xe \LaTeX .

```
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 `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).

```
71 \expandafter\ifx\csname T@X2\endcsname\relax\else
72   \DeclareTextSymbolDefault{\AA}{OT1}
73   \DeclareTextSymbolDefault{\aa}{OT1}
74   \DeclareTextCommand{\aa}{OT1}{\r a}
75   \DeclareTextCommand{\AA}{OT1}{\r A}
76 \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.

```
77 %\DeclareRobustCommand{\latintext}{%
78 % \fontencoding{\latinencoding}\selectfont
79 % \def\encodingdefault{\latinencoding}}
80 %\let\lat\latintext
```

`\textcyrillic` $\{\langle text \rangle\}$

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}`.

```
81 \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?

```
82 \@ifpackageloaded{inputenc}{%
83 \if@uni@ode
84 \PackageWarning{babel}{inputenc should not be used with LuaTeX or XeTeX}
85 \fi
86 }{%
87 %\def\reserved@a{LWN}%
88 %\ifx\reserved@a\cyrillicencoding\else
89 \def\reserved@a{OT2}%
90 \ifx\reserved@a\cyrillicencoding\else
91 \def\reserved@a{EU1}%
92 \ifx\reserved@a\cyrillicencoding\else
93 \def\reserved@a{EU2}%
94 \ifx\reserved@a\cyrillicencoding\else
95 \PackageWarning{babel}%
96 {No input encoding specified for Belarusian language}
97 \fi\fi\fi
98 %\fi
99 }
```

7.4 Shorthands

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

```
100 \initiate@active@char{''}
```

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

```
101 %\def\mdqon{\bbl@activate{''}}
102 %\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 `''`.

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

```
108 \declare@shorthand{belarusian}{''}{\quotedblbase}
109 \declare@shorthand{belarusian}{'''}{\textquotedblleft}
110 \declare@shorthand{belarusian}{''<}{\guillemotleft}
111 \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.

```
112 \declare@shorthand{belarusian}{' '}{\hskip\z@skip}
113 \declare@shorthand{belarusian}{' '}{\textormath{\leavevmode\hbox{-}}{-}}
114 \declare@shorthand{belarusian}{' '=}{\nobreak-\hskip\z@skip}
115 \declare@shorthand{belarusian}{'|}{\%}
116 \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`.

```
117 \declare@shorthand{belarusian}{' -}{\%}
118 \def\belarusian@sh@tmp{\%
```

```

119 \if\belarusian@sh@next-\expandafter\belarusian@sh@emdash
120 \else\expandafter\belarusian@sh@hyphen\fi}%
121 \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.

```

122 \def\belarusian@sh@hyphen{\nobreak\-\bbl@allowhyphens}
123 \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`.

```

124 \def\cdash#1#2#3{\def\tempx@{#3}%
125 \def\tempa@{-}\def\tempb@{~}\def\tempc@{*}%
126 \ifx\tempx@\tempa@\@Acdash\else
127 \ifx\tempx@\tempb@\@Bcdash\else
128 \ifx\tempx@\tempc@\@Ccdash\else
129 \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.

```

130 \def\@Acdash{\ifdim\lastskip>\z@\unskip\nobreak\hskip.2em\fi
131 \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.

```

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

```

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

```

134 \def\@Ccdash{\leavevmode
135 \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=!—!`

```

136 %\ifx\cyrdash\undefined
137 % \def\cyrdash{\hbox to.8em{--\hss--}}
138 % \PackageInfo{babel}{\string\cyrdash\space is defined}
139 %\fi
140 \ProvideTextCommandDefault{\cyrdash}{\iflanguage{belarusian}%
141 {\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.

```
142 \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.

```
143 \if@uni@code
144   %\captionsbelarusian@official
145   \addto\captionsbelarusian{%
146     \def\prefacename{Прадмова}% [babel]
147     \def\refname{Спіс літаратуры}% [only article]
148     \def\abstractname{Анагачыя}% [only article, report]
149     \def\bibname{Літаратура}% [only book, report]
150     \def\chaptername{Глава}% [only book, report]
151     \def\appendixname{Дадатак}%
152     \def\contentsname{Змест}%
153     \let\tocname=\contentsname
154     \def\listfigurename{Спіс ілюстрацый}%
155     \def\listtablename{Спіс табліц}%
156     \def\indexname{Прадметны паказальнік}%
157     \def\authorname{Паказальнік імён}%
158     \def\figurename{Рыс.}%
159     \def\tablename{Табліца}%
160     \def\partname{Частка}%
161     \def\enclname{укл.}%
162     \def\ccname{зых.}%
163     \def\headtoname{вх.}%
164     \def\pagename{с.}% [letter]
165     \def\seename{гл.}%
166     \def\alsosome{гл.\ таксама}%
167     \def\proofname{Доказ}% [amsthm]
168     \def\glossaryname{Слоўнік тэрмінаў}%
169     \def\acronymname{Абрэвіятуры}% [glossaries] {Acronyms}
```

```

170 \def\lstlistingname{Лістынг}% [listings] (the environment) {Listing}
171 \def\lstlistlistingname{Лістынгi}% [listings] (the "List of") {Listings}
172 \def\notesname{Назакi}% [endnotes] {Notes}
173 }%

```

Additional definitions for the package nomencl:

```

174 %% =====
175 %% nomencl
176 \ifdefined\nomname
177 \addto\captionsbelarusian{%
178 \def\nomname{Азначэннi}%
179 \def\eqdeclaration#1{, гл.\nobreakspace(#1)}%
180 \def\pagedeclaration#1{, сар.\nobreakspace#1}%
181 }%
182 \fi

```

Additional captions for the revtex class.

```

183 %% =====
184 %% RevTeX4 & RevTeX4-1
185 %\@ifclassloaded{revtex4-1}
186 %\{<true code>}
187 %\{<false coode>}%
188 %\@ifclassloaded{revtex4-1}
189 %\{
190 \ifnum\@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
191 \addto\captionsbelarusian{%
192 \def\lofname{\listfigurename}
193 \def\lotname{\listtablename}
194 \def\figuresname{Рысункi}%{Figures}%
195 \def\tablesname{Табліцы}%{Tables}%
196 \def\appendixesname{Дадацкi}%{Appendixes}%
197 \def\acknowledgmentsname{Падзякi}%{Acknowledgments}
198 \def\andname{и}%{and}
199 \def\@pacs@name{PACS коды: }%{PACS numbers: }%
200 \def\@keys@name{Ключавыя словы: }%{Keywords: }%
201 \def\Dated@name{Дата: }%{Dated: }%
202 \def\Received@name{Атрымана }%{Received }%
203 \def\Revised@name{Выпраўленая версія }%{Revised }%
204 \def\Accepted@name{Прынята }%{Accepted }%
205 \def\Published@name{Апублікавана }%{Published }%
206 }%
207 %\}%
208 \fi

```

Now we proceed to the classic version in Unicode encoding.

```

209 \addto\captionsbelarusian@tarask{%
210 \def\prefacename{Прадмова}% [babel]
211 \def\refname{Сьпіс літаратуры}% [only article]
212 \def\abstractname{Анааця}% [only article, report]
213 \def\bibname{Літаратура}% [only book, report]
214 \def\chaptername{Глава}% [only book, report]
215 \def\appendixname{Дадааак}%
216 \def\contentsname{Зьмест}%
217 \let\tocname=\contentsname
218 \def\listfigurename{Сьпіс ілюстрацый}%

```



```

219 \def\listtablename{Сьпіс табліц}%
220 \def\indexname{Прадметны паказальнік}%
221 \def\authorname{Паказальнік імён}%
222 \def\figurename{Рыс.}%
223 \def\tablename{Табліца}%
224 \def\partname{Частка}%
225 \def\enclname{укл.}%
226 \def\ssname{зых.}%
227 \def\headtoname{вх.}%
228 \def\pagename{с.}% [letter]
229 \def\seename{гл.}%
230 \def\alsoname{гл.\ таксама}%
231 \def\proofname{Доказ}% [amsthm]
232 \def\glossaryname{Слоўнік тэрмінаў}%
233 \def\acronymname{Абрэвіятуры}% [glossaries] {Acronyms}
234 \def\lstlistingname{Лістынг}% [listings] (the environment) {Listing}
235 \def\lstlistlistingname{Лістынгі}% [listings] (the "List of") {Listings}
236 \def\nomname{Азначэньні}%
237 \def\notesname{Надаткі}% [endnotes] {Notes}
238 }
239 %% =====
240 %% RevTeX4 & RevTeX4-1
241 %\@ifclassloaded{revtex4-1}
242 %{\%
243 \ifnum\@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
244 \addto\captionsbelarusian@tarask{%
245 \def\lofname{\listfigurename}
246 \def\lotname{\listtablename}
247 \def\figuresname{Рысункі}%{Figures}%
248 \def\tablesname{Табліцы}%{Tables}%
249 \def\appendixesname{Дадаткі}%{Appendixes}%
250 \def\acknowledgmentsname{Падзякі}%{Acknowledgments}
251 \def\andname{и}%{and}
252 \def\@pacs@name{PACS коды: }%{PACS numbers: }%
253 \def\@keys@name{Ключавыя словы: }%{Keywords: }%
254 \def\Dated@name{Дата: }%{Dated: }%
255 \def\Received@name{Атрымана }%{Received }%
256 \def\Revised@name{Выпраўленая вэрсія }%{Revised }%
257 \def\Accepted@name{Прынята }%{Accepted }%
258 \def\Published@name{Апублікавана }%{Published }%
259 }%
260 %}{}%
261 \fi
262 %% =====
263 \else
264 %\def\captionsbelarusian@official{%
265 \addto\captionsbelarusian{%
266 \def\prefacename{\CYRP\cyrr\cyra\cyrd\cyrm\cyro\cyrv\cyra}% [babel]
267 \def\refname{\CYRS\cyrp\cyrii\cyrs \cyrl\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrr\cyry}%
268 \def\abstractname{\CYRA\cyrn\cyra\cyrt\cyra\cyrc\cyry\cyrya}% [only article, report]
269 \def\bibname{\CYRL\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrr\cyra}% [only book, report]
270 \def\chaptername{\CYRG\cyrl\cyra\cyrv\cyra}% [only book, report]
271 \def\appendixname{\CYRD\cyra\cyrd\cyra\cyrt\cyra\cyrk}%
272 \def\contentsname{\CYRZ\cyrm\cyre\cyrs\cyrt}%

```

```

273 \let\tocname=\contentsname
274 \def\listfigurename{\CYRS\cyrp\cyrii\cyrs \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyre}
275 \def\listtablename{\CYRS\cyrp\cyrii\cyrs \cyrt\cyra\cyrb\cyrl\cyrii\cyrc}%
276 \def\indexname{\CYRP\cyrr\cyra\cyrd\cyrm\cyre\cyrt\cyrn\cyrery \cyrp\cyra\cyrk\cyra\cyrz\cyra\cyrl\cyrsftsn\cyrn\cyrii\cyrk \cyrii\cyrm\cyra\cyrc}%
277 \def\authorname{\CYRP\cyra\cyrk\cyra\cyrz\cyra\cyrl\cyrsftsn\cyrn\cyrii\cyrk \cyrii\cyrm\cyra\cyrc}%
278 \def\figurename{\CYRR\cyrrery\cyrs.}%
279 \def\tablename{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyra}%
280 \def\partname{\CYRCH\cyra\cyrs\cyrt\cyrk\cyra}%
281 \def\enclname{\cyru\cyrk\cyrl.}%
282 \def\ccname{\cyrz\cyrrery\cyrh.}%
283 \def\headtoname{\cyrv\cyrh.}%
284 \def\pagename{\cyrs.}% [letter]
285 \def\seename{\cyrg\cyrl.}%
286 \def\alsoname{\cyrg\cyrl.\ \cyrt\cyra\cyrk\cyrs\cyra\cyrm\cyra}%
287 \def\proofname{\CYRD\cyro\cyrk\cyra\cyrz}% [amsthm]
288 \def\glossaryname{\CYRS\cyrl\cyro\cyrushrt\cyrn\cyrii\cyrk \cyrt\cyrrerev\cyrr\cyrm\cyrii\cyrc}%
289 \def\acronymname{\CYRA\cyrb\cyrr\cyrrerev\cyrv\cyrii\cyrya\cyrt\cyru\cyrr\cyrrery}% [gloss]
290 \def\lstlistingname{\CYRL\cyrii\cyrs\cyrt\cyrrery\cyrn\cyrg}% [listings] (the environme
291 \def\lstlistlistingname{\CYRL\cyrii\cyrs\cyrt\cyrrery\cyrn\cyrg\cyrii}% [listings] (the 'l
292 \def\nomname{\CYRA\cyrz\cyrn\cyra\cyrch\cyrrerev\cyrn\cyrn\cyrii}%
293 \def\notesname{\CYRN\cyra\cyrt\cyra\cyrt\cyrk\cyrii}% [endnotes] {Notes}
294 }%
295 %% =====
296 %% RevTeX4 & RevTeX4-1
297 \ifnum \@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
298 \addto\captionsbelarusian{%
299 \def\lofname{\listfigurename}
300 \def\lotname{\listtablename}
301 \def\figuresname{\CYRR\cyrrery\cyrs\cyru\cyrn\cyrk\cyrii}{Figures}%
302 \def\tablesname{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyrrery}{Tables}%
303 \def\appendixesname{\CYRD\cyra\cyrd\cyra\cyrt\cyrk\cyrii}{Appendixes}%
304 \def\acknowledgmentsname{\CYRP\cyra\cyrd\cyrz\cyrya\cyrk\cyrii}{Acknowledgments}
305 \def\andname{\cyri}{and}
306 \def\@pacs@name{PACS \cyrk\cyro\cyrd\cyrrery: }{PACS numbers: }%
307 \def\@keys@name{\CYRK\cyrl\cyryu\cyrch\cyra\cyrv\cyrrery\cyrya \cyrs\cyrl\cyro\cyrv\cyrrery}
308 \def\Dated@name{\CYRD\cyra\cyrt\cyra: }{Dated: }%
309 \def\Received@name{\CYRA\cyrt\cyrr\cyrrery\cyrm\cyra\cyrn\cyra }{Received }%
310 \def\Revised@name{\CYRV\cyrrery\cyrp\cyrr\cyra\cyrushrt\cyrl\cyre\cyrn\cyra\cyrya \cyrv\cyra}
311 \def\Accepted@name{\CYRP\cyrr\cyrrery\cyrn\cyrya\cyrt\cyra }{Accepted }%
312 \def\Published@name{\CYRA\cyrp\cyru\cyrb\cyrl\cyrii\cyrk\cyra\cyrv\cyra\cyrn\cyra }{Publ
313 }%
314 \fi
315 %% =====
316 \addto\captionsbelarusian@tarask{%
317 \def\prefacename{\CYRP\cyrr\cyra\cyrd\cyrm\cyro\cyrv\cyra}% [babel]
318 \def\refname{\CYRS\cyrsftsn\cyrp\cyrii\cyrs \cyrl\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrn\cyra}
319 \def\abstractname{\CYRA\cyrn\cyra\cyrt\cyra\cyrc\cyrrery\cyrya}% [only article, report]
320 \def\bibname{\CYRL\cyrii\cyrt\cyra\cyrr\cyra\cyrt\cyru\cyrr\cyra}% [only book, report]
321 \def\chaptername{\CYRG\cyrl\cyra\cyrv\cyra}% [only book, report]
322 \def\appendixname{\CYRD\cyra\cyrd\cyra\cyrt\cyra\cyrk}%
323 \def\contentsname{\CYRZ\cyrsftsn\cyrm\cyre\cyrs\cyrt}%
324 \let\tocname=\contentsname
325 \def\listfigurename{\CYRS\cyrsftsn\cyrp\cyrii\cyrs \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc}
326 \def\listtablename{\CYRS\cyrsftsn\cyrp\cyrii\cyrs \cyrt\cyra\cyrb\cyrl\cyrii\cyrc}%

```

```

327 \def\indexname{\CYRP\cyrr\cyra\cyrd\cym\cyre\cyrt\cyrn\cyrery \cyrp\cyra\cyrk\cyra\cyrz\cyro}
328 \def\authorname{\CYRP\cyra\cyrk\cyra\cyrz\cyra\cyrl\cyrstfn\cyrn\cyrii\cyrk \cyrii\cym\cyro}
329 \def\figurename{\CYRR\cyrery\cyr.}%
330 \def\tablename{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyra}%
331 \def\partname{\CYRCH\cyra\cyr\cyrt\cyrk\cyra}%
332 \def\enclname{\cyru\cyrk\cyrl.}%
333 \def\ccname{\cyrz\cyrery\cyrh.}%
334 \def\headtoname{\cyrv\cyrh.}%
335 \def\pagename{\cyr.}% [letter]
336 \def\seename{\cyrg\cyrl.}%
337 \def\alsoname{\cyrg\cyrl.\ \cyrt\cyra\cyrk\cyr\cyra\cym\cyra}%
338 \def\proofname{\CYRD\cyro\cyrk\cyra\cyrz}% [amsthm]
339 \def\glossaryname{\CYRS\cyrl\cyro\cyrushrt\cyrn\cyrii\cyrk \cyrt\cyrerev\cyrr\cym\cyrii\cyro}
340 \def\acronymname{\CYRA\cyrb\cyrr\cyrerev\cyrv\cyrii\cyra\cyrt\cyru\cyrr\cyrery}% [gloss]
341 \def\lstlistingname{\CYRL\cyrii\cyr\cyrt\cyrery\cyrn\cyrg}% [listings] (the environm
342 \def\lstlistlistingname{\CYRL\cyrii\cyr\cyrt\cyrery\cyrn\cyrg\cyrii}% [listings] (the 'T
343 \def\nomname{\CYRA\cyrz\cyrn\cyra\cyrch\cyrerev\cyrn\cyrstfn\cyrn\cyrii}%
344 \def\notesname{\CYRN\cyra\cyrt\cyra\cyrt\cyrk\cyrii}% [endnotes] {Notes}
345 }%
346 %% =====
347 %% RevTeX4 & RevTeX4-1
348 \ifnum \@ifclassloaded{revtex4}{1}{\@ifclassloaded{revtex4-1}{1}{0}}>0
349 \addto\captionsbelarusian@tarask{%
350 \def\lofname{\listfigurename}
351 \def\lotname{\listtablename}
352 \def\figuresname{\CYRR\cyrery\cyr\cyru\cyrn\cyrk\cyrii}% {Figures}%
353 \def\tablesname{\CYRT\cyra\cyrb\cyrl\cyrii\cyrc\cyrery}% {Tables}%
354 \def\appendixesname{\CYRD\cyra\cyrd\cyra\cyrt\cyrk\cyrii}% {Appendixes}%
355 \def\acknowledgmentsname{\CYRP\cyra\cyrd\cyrz\cyra\cyrk\cyrii}% {Acknowledgments}
356 \def\andname{\cyri}% {and}
357 \def\@pacs@name{PACS \cyrk\cyro\cyrd\cyrery: }% {PACS numbers: }%
358 \def\@keys@name{\CYRK\cyrl\cyru\cyrch\cyra\cyrv\cyrery\cyra \cyr\cyrl\cyro\cyrv\cyrery}
359 \def\Dated@name{\CYRD\cyra\cyrt\cyra: }% {Dated: }%
360 \def\Received@name{\CYRA\cyrt\cyrr\cyrery\cym\cyra\cyrn\cyra }% {Received }%
361 \def\Revised@name{\CYRV\cyrery\cyrp\cyrr\cyra\cyrushrt\cyrl\cyre\cyrn\cyra\cyra \cyrv\cyro}
362 \def\Accepted@name{\CYRP\cyrr\cyrery\cyrn\cyra\cyrt\cyra }% {Accepted }%
363 \def\Published@name{\CYRA\cyrp\cyru\cyrb\cyrl\cyrii\cyrk\cyra\cyrv\cyra\cyrn\cyra }% {Publ
364 }%
365 \fi
366 %% =====
367 \fi

```

7.5.2 Date in Belarusian

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

```

368 \if@uni@ode
369 \addto\datebelarusian{%
370 \def\today{\number\day~\ifcase\month\or
371 студзеня\or
372 лютага\or
373 сакавіка\or
374 красавіка\or

```

```

375    мая\or
376    чэрвеня\or
377    ліпеня\or
378    жніўня\or
379    верасня\or
380    кастрычніка\or
381    лістапада\or
382    снежня\fi
383    \space \number\year~r.}}
384 \def\datebelarusian@tarask{%
385 \def\today{\number\day~\ifcase\month\or%
386 студзеня\or
387 лютага\or
388 сакавіка\or
389 красавіка\or
390 траўня\or
391 чэрвеня\or
392 ліпеня\or
393 жніўня\or
394 верасня\or
395 кастрычніка\or
396 лістапада\or
397 сьнежня\fi%
398 \space \number\year~r.}}
399 \else
400 \def\datebelarusian{%
401 \def\today{\number\day~\ifcase\month\or
402 \cyr\cyrt\cyru\cyrd\cyrz\cyre\cyrn\cyrya\or
403 \cyr\cyryu\cyrt\cyra\cyrg\cyra\or
404 \cyr\cyra\cyrk\cyra\cyrv\cyrii\cyrk\cyra\or
405 \cyrk\cyrr\cyra\cyrs\cyra\cyrv\cyrii\cyrk\cyra\or
406 \cyrm\cyra\cyrya\or
407 \cyrch\cyrerev\cyrr\cyrv\cyre\cyrn\cyrya\or
408 \cyr\cyrii\cyrp\cyre\cyrn\cyrya\or
409 \cyrzh\cyrn\cyrii\cyrushrt\cyrn\cyrya\or
410 \cyrv\cyre\cyrr\cyra\cyrs\cyrn\cyrya\or
411 \cyrk\cyra\cyrs\cyrt\cyrr\cyrery\cyrch\cyrn\cyrii\cyrk\cyra\or
412 \cyr\cyrii\cyrs\cyrt\cyra\cyrp\cyra\cyrd\cyra\or
413 \cyrs\cyrn\cyre\cyrzh\cyrn\cyrya\fi
414 \space \number\year~\cyr.g.}}
415 \def\datebelarusian@tarask{%
416 \def\today{\number\day~\ifcase\month\or
417 \cyrs\cyrt\cyru\cyrd\cyrz\cyre\cyrn\cyrya\or
418 \cyr\cyryu\cyrt\cyra\cyrg\cyra\or
419 \cyrs\cyra\cyrk\cyra\cyrv\cyrii\cyrk\cyra\or
420 \cyrk\cyrr\cyra\cyrs\cyra\cyrv\cyrii\cyrk\cyra\or
421 \cyrt\cyrr\cyra\cyrushrt\cyrn\cyrya\or
422 \cyrch\cyrerev\cyrr\cyrv\cyre\cyrn\cyrya\or
423 \cyr\cyrii\cyrp\cyre\cyrn\cyrya\or
424 \cyrzh\cyrn\cyrii\cyrushrt\cyrn\cyrya\or
425 \cyrv\cyre\cyrr\cyra\cyrs\cyrsftsn\cyrn\cyrya\or
426 \cyrk\cyra\cyrs\cyrt\cyrr\cyrery\cyrch\cyrn\cyrii\cyrk\cyra\or
427 \cyr\cyrii\cyrs\cyrt\cyra\cyrp\cyra\cyrd\cyra\or
428 \cyrs\cyrsftsn\cyrn\cyre\cyrzh\cyrn\cyrya\fi%

```

```

429 \space \number\year~\cyr{.}}
430 \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 `belarusian.ldf`.

```

431 \providehyphenmins{\CurrentOption}{\tw@\tw@}
432 \providehyphenmins{belarusian}{\tw@\tw@}

```

7.5.4 Extra definitions

`\extrasbelarusian` 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`.

```

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

```

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

```

435 \addto\extrasbelarusian{\languageshorthands{belarusian}}
436 \addto\extrasbelarusian{\bbl@activate{'}}
437 \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.

```

438 \addto\extrasbelarusian{\bbl@frenchspacing}
439 \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`.

```
440 %\def\@alph#1{\fontencoding{\latinencoding}\selectfont
441 % \ifcase#1\or
442 %   a\or b\or c\or d\or e\or f\or g\or h\or
443 %   i\or j\or k\or l\or m\or n\or o\or p\or
444 %   q\or r\or s\or t\or u\or v\or w\or x\or
445 %   y\or z\else\@ctrerr\fi}%
446 %\def\@Alph#1{\fontencoding{\latinencoding}\selectfont
447 % \ifcase#1\or
448 %   A\or B\or C\or D\or E\or F\or G\or H\or
449 %   I\or J\or K\or L\or M\or N\or O\or P\or
450 %   Q\or R\or S\or T\or U\or V\or W\or X\or
451 %   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.

```
452 \def\Asbuk#1{\expandafter\belarusian@Alph\csname c@#1\endcsname}
453 \if@uni@ode
454   \def\belarusian@Alph#1{\ifcase#1\or
455     A\or B\or B\or Г\or Д\or E\or Ж\or
456     З\or I\or K\or Л\or M\or H\or O\or
457     П\or P\or C\or T\or Y\or Ф\or X\or
458     Ц\or Ч\or Ш\or Э\or Ё\or Я\else\@ctrerr\fi}
459 \else
460   \def\belarusian@Alph#1{\ifcase#1\or
461     \CYRA\or\CYRB\or\CYRV\or\CYRG\or\CYRD\or\CYRE\or\CYRZH\or
462     \CYRZ\or\CYRII\or\CYRK\or\CYRL\or\CYRM\or\CYRN\or\CYRO\or
463     \CYRP\or\CYRR\or\CYRS\or\CYRT\or\CYRU\or\CYRF\or\CYRH\or
464     \CYRC\or\CYRCH\or\CYRSH\or\CYREREV\or\CYRYU\or
465     \CYRYA\else\@ctrerr\fi}
466 \fi
```

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

```
467 \def\asbuk#1{\expandafter\belarusian@alph\csname c@#1\endcsname}
468 \if@uni@ode
469   \def\belarusian@alph#1{\ifcase#1\or
470     a\or б\or в\or r\or д\or e\or ж\or
471     з\or i\or к\or л\or м\or н\or o\or
```

```

472   п\or p\or c\or т\or y\or ф\or x\or
473   ц\or ч\or ш\or э\or ю\or я\else\@ctrerr\fi}
474 \else
475   \def\belarusian@alph#1{\ifcase#1\or
476     \cyra\or\cyrb\or\cyrv\or\cyrg\or\cyrd\or\cyre\or\cyrzh\or
477     \cyrz\or\cyrii\or\cyrk\or\cyr1\or\cyrn\or\cyro\or
478     \cyrp\or\cyrr\or\cyrs\or\cyrt\or\cyru\or\cyrf\or\cyrh\or
479     \cyrç\or\cyrch\or\cyrsh\or\cyrrev\or\cyrü\or
480     \cyrÿ\else\@ctrerr\fi}
481 \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.

```

482 \bbl@declare@ttribute{belarusian}{classic}{%
483   \PackageInfo{babel}{Belarusian attribute set to classic}%
484   \let\captionbelarusian=\captionbelarusian@tarask
485   \let\datebelarusian=\datebelarusian@tarask
486 }
487 \bbl@declare@ttribute{belarusian}{tarask}{%
488   \PackageInfo{babel}{Belarusian attribute set to tarask}%
489   \let\captionbelarusian=\captionbelarusian@tarask
490   \let\datebelarusian=\datebelarusian@tarask
491 }

```

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

```

492 \@onlypreamble\captionbelarusian@tarask
493 \@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.

```

494 %\DeclareRobustCommand{\No}{%
495 %   \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.

```

496 %\RequirePackage{textmath}
497 %   \@ifundefined{sym\cyrillicencoding letters}{\}%
498 %   \SetSymbolFont{\cyrillicencoding letters}{bold}\cyrillicencoding
499 %   \rmdefault\bfdefault\updefault
500 %   \DeclareSymbolFontAlphabet\cyrmathrm{\cyrillicencoding letters}

```

And we need few commands to switch to different variants.

```

501 %\DeclareMathAlphabet\cyrmathbf\cyrillicencoding
502 %   \rmdefault\bfdefault\updefault
503 %\DeclareMathAlphabet\cyrmathsf\cyrillicencoding
504 %   \sfdefault\mddefault\updefault
505 %\DeclareMathAlphabet\cyrmathit\cyrillicencoding

```

```

506 % \rmdefault\mddefault\itdefault
507 %\DeclareMathAlphabet\cyrmath\cyrillicencoding
508 % \ttdefault\mddefault\updefault
509 %
510 %\SetMathAlphabet\cyrmathsf\bold\cyrillicencoding
511 % \sfdefault\bfdefault\updefault
512 %\SetMathAlphabet\cyrmathit\bold\cyrillicencoding
513 % \rmdefault\bfdefault\itdefault
514 %}

```

\sh 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:

```

\arctg
\arcctg
\th
\cth
\cosec
515 \def\sh{\mathop{\operator@font sh}\nolimits}
516 \def\ch{\mathop{\operator@font ch}\nolimits}
517 \def\tg{\mathop{\operator@font tg}\nolimits}
518 \def\ctg{\mathop{\operator@font ctg}\nolimits}
519 \def\arctg{\mathop{\operator@font arctg}\nolimits}
520 \def\arcctg{\mathop{\operator@font arcctg}\nolimits}
521 \addto\extrasbelarusian{%
522   \babel@save{\th}%
523   \let\ltx@th\th
524   \def\th{\textormath{\ltx@th}}%
525   {\mathop{\operator@font th}\nolimits}}%
526 }
527 \def\cth{\mathop{\operator@font cth}\nolimits}
528 \def\cosec{\mathop{\operator@font cosec}\nolimits}

```

\Prob Finally, we define some rare Belarusian mathematical symbols:

```

\Variance
\nod
\nok
\NOD
\NOK
\Proj
529 \def\Prob{\mathop{\kern\z@\mathsf{P}}\nolimits}
530 \def\Variance{\mathop{\kern\z@\mathsf{D}}\nolimits}
531 \if@uni@ode
532 \def\nod{\mathop{\mathrm{н.а.д.}}\nolimits}
533 \def\nok{\mathop{\mathrm{н.а.к.}}\nolimits}
534 \def\NOD{\mathop{\mathrm{HAД}}\nolimits}
535 \def\NOK{\mathop{\mathrm{HAK}}\nolimits}
536 \def\nad{\mathop{\mathrm{н.а.д.}}\nolimits}
537 \def\nak{\mathop{\mathrm{н.а.к.}}\nolimits}
538 \def\NAD{\mathop{\mathrm{HAД}}\nolimits}
539 \def\NAK{\mathop{\mathrm{HAK}}\nolimits}
540 \def\Proj{\mathop{\mathrm{\Pi p}}\nolimits}
541 \else
542 \def\nod{\mathop{\cyrmathrm{\cyrn.\cyra.\cyrd.}}\nolimits}
543 \def\nok{\mathop{\cyrmathrm{\cyrn.\cyra.\cyrk.}}\nolimits}
544 \def\NOD{\mathop{\cyrmathrm{\CYRN\CYRA\CYRD}}\nolimits}
545 \def\NOK{\mathop{\cyrmathrm{\CYRN\CYRA\CYRK}}\nolimits}
546 \def\nad{\mathop{\cyrmathrm{\cyrn.\cyra.\cyrd.}}\nolimits}
547 \def\nak{\mathop{\cyrmathrm{\cyrn.\cyra.\cyrk.}}\nolimits}
548 \def\NAD{\mathop{\cyrmathrm{\CYRN\CYRA\CYRD}}\nolimits}
549 \def\NAK{\mathop{\cyrmathrm{\CYRN\CYRA\CYRK}}\nolimits}

```



```

550 \def\Proj{\mathop{\cyrmathrm{\CYRP\cyrr}}\nolimits}
551 \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.

```

552 \ldf@finish{belarusian}

```

20pt by default