

The Russian Language in the babel system

Version 1.3f

Igor A. Kotelnikov*

Released 2014/10/21

Contents

1	The Russian 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	Modern and Ancient 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	9
7.1	Initial setup	9
7.2	Output encoding	10
7.3	Input encoding	13
7.4	Shorthands	14
7.4.1	Quotes	14
7.4.2	Emdash, endash and hyphenation sign	15
7.5	Switching to and from Russian	16
7.5.1	Caption names	16
7.5.2	Date in Russian	21
7.5.3	Hyphenation patterns	22
7.5.4	Extra definitions	22
7.6	Alphabetic counters	23
7.7	Cyrillic math	25
7.8	Final settings	26

*E-mail: [kia999 at mail dot ru](mailto:kia999@mail.ru).

1 The Russian Language Definition File

The file `russianb.ldf`¹ is the source file for the Russian Language Definition file `russianb.ldf` to be loaded by the `babel` package with the option `russian`. It was derived by Igor A. Kotelnikov from the original version of `russianb.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.

Starting the version 1.2, `russianb.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`.

The version 1.3 of `russianb.ldf` has been adapted to new features introduced in the version 3.9 of the `babel` package. In particular, the language attribute `ancient` has been introduced to support typesetting ancient and Church Slavonic books.

2 Usage

Typesetting Russian 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 their 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 Russian), 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 1.3f and was last revised on 2014/10/21.

2.1 L^AT_EX

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,russian]{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 `cp866` (MS DOS), `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 excluded from Russian support stating version 1.2 of `russianb.ldf` since `LWN` is 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[lutf8]{luainputenc}
\usepackage[english,russian]{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,russian]{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,russian]{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=russian,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,russian]{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,russian]{babel}
```

2.4 Modern and Ancient spelling

By default, a modern spelling is enabled. For Church Slavonic and other old books ancient spelling can be enabled by setting the attribute to `ancient`. To set an attribute, put the `\languageattribute` macro within a document preamble after `babel`, for example,

```
\usepackage[english,russian]{babel}
\languageattribute{russian}{ancient}
```

Setting the `ancient` attribute changes the built-in strings (caption names) and a date format. For example, the bibliography will be entitled as ‘Литература’ by default and as ‘Библиография’ if the Russian language attribute is set to `ancient`. Same result can be achieved using a modifier as follows:

```
\usepackage[english,russian.ancient]{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).

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{russian}` switches to the Russian language, with the following effects:

1. Russian hyphenation patterns are made active;
2. `\today` prints the date in Russian;
3. the caption names are translated into Russian (L^AT_EX only);
4. emdash typed by the ligature `---` might be 20% shorter when Russian is the current language; the result depends on the current encoding; `---` always produce long emdash in Lua_TE_X and Xe_TE_X since these engines use same encodings for all languages;
5. emdash typed by the ligature `”---` in Russian 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, a modern spelling is used for built-in strings (caption names) and the date. The spelling can be reverted to ancient by setting the language attribute to `ancient` in the document preamble as discussed in Sec. 2.4.

Since Russian has its own numbering system, `russianb.ldf` adds macros `\asbuk{<counter>}` and `\Asbuk{<counter>}` for formatting numbers appropriately the alphabetic sequence in the Russian 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_TE_X.
2. German quotation marks can be entered using the commands `\glqq` and `\grqq` which work in L^AT_EX 2_ε and Plain_TE_X.

The macro `\Russian` is now defined as an alias for `\selectlanguage{russian}`, and its “opponent” `\English`, existed in `russianb.ldf` prior to version 1.2 has been removed since the Russian 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 Russian; it is essentially an alias for `\foreignlanguage{russian}{<text>}`.

3.1 Active character

Table 1 shows macros and active string which can be used to typeset various dashes and quotes. In the Russian 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 `russianb.ldf`

<code>\glqq</code>	”“	German left double quotes (looks like „).
<code>\grqq</code>	””	German right double quotes (looks like “).
<code>\guillemotleft</code>	”<	French left double quotes (looks like <<).
<code>\guillemotright</code>	”>	French right double quotes (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 does not print hyphen; equivalent to <code>\babelhyphen{empty}</code> in <code>babel 3.9</code> .
<code>\babelhyphen*{nobreak}</code>	”~	A 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 in 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 Russian 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 `\shorthandon{’}` and the switched on again using `\shorthandon{”}`. The aliases `\mdqoff` and `\mdqon` for these two macros has been removed from `russianb.ldf` starting from version 1.3 in favour of the macros `\shorthandon` and `\shorthandoff` provided in the `babel` core.

3.2 Math commands

`russianb.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 Russian 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`, `\NOD`, `\nod`, `\NOK`, `\nok`, `\Proj` print some rare Russian 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, Russian letter ‘a’ matches macro `\cyra`, and capital Russian 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 `russianb.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 `\Russian`, `\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 vice versa delete all `.aux`, `.toc`, `.lot`, `.lof` files as they might have stored incompatible internal encodings.

T2* font encodings do not have old Slavonic letter ‘yat’ (Ѣ, ѣ), which is hard-coded in ancient caption names. Be sure to use an Unicode engine or borrow `\cyryat` and `\CYRYAT` commands from X2 font encoding when setting the language attribute to “ancient”, for example:

```
\usepackage[X2,T2A]{fontenc}
\usepackage[utf8]{inputenc}
```

```
\DeclareUnicodeCharacter{0462}{\CYRYAT}
\DeclareTextSymbolDefault{\CYRYAT}{X2}
\DeclareUnicodeCharacter{0463}{\cyryat}
\DeclareTextSymbolDefault{\cyryat}{X2}
\usepackage[english,russian.ancient]{babel}
```

None of Cyrillic font encoding have ‘iotated E’ (ӀӇ, ӓӇ). When running legacy engines you are advised to substitute it with `\CYRIE`, `\cyrie` (ӀӇ, ӓӇ):

```
\DeclareUnicodeCharacter{0464}{\CYRIE}
\DeclareUnicodeCharacter{0465}{\cyrie}
```

6 History

Changes in version 1.3f

- .

Changes in version 1.3e

- All stuff is now generated from single dtx file.

Changes in version 1.3d

- Bug fix in `\Proj`.

Changes in version 1.3c

- Bug fix in `\daterussian`.

Changes in version 1.3b

- Added support for the packages `listing`, `nomencl`, and `nomentbl`.

Changes in version 1.3a

- Added the Russian language attribute `ancient` for typesetting ancient slavonic books.

Changes in version 1.3

- Updated for babel 3.9.
- The `\alph` and `\Alph` commands are not redefined any more by the `rus-sianb.lfd`.

Changes in version 1.2a

- Indentation of the first paragraph after sectioning command has been removed. Use `indentfirst` to make first paragraph indented.

Changes in version 1.2

- Font encoding LWN is not supported any more.
- Macros `\Rus`, `\English`, `\Eng` have been removed. `\Russian` is now alias of `\selectlanguage{russian}`. Former definition of `\Russian` is preserved in the macros `\cyrillictext` and `\cyr` which should not be used at user level.
- `LuaLATEX` and `XeLATEX` are now supported; `\cyr...` commands are bypassed if `LuaLATEX` or `XeLATEX` run.
- The shorthands `”<` and `”>` are redefined; now they match `\guillemotleft` and `\guillemotright`, respectively, rather than `\flqq` and `\frqq`.
- Definitions of `\latinencoding` and `\latintext` are removed since these macros are defined in `babel.def`.
- `\cyrmathbf`, `\cyrmathsf`, `\cyrmathit`, and `\cyrmathtt` commands are removed; they do work in math mode; instead of them it is advised to use corresponding `\text...` command.
- The macro `\No` is removed since the Cyrillic number sign is available on keyboard and can also be typed using the `\textnumero` macro.
- Added translation for Glossary.

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{russian}{captionsrussian}
```

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 `LATEX 2ε` 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.

```
2 \ifdefined\if@uni@ode
3   \PackageError{babel}{if@uni@ode already defined.\MessageBreak
4     Please contact author of russianb.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 Russian language have been loaded in `language.dat`. Namely, we check for the existence of `\l@russian`. If it is not defined, we declare Russian as dialect for the default language number 0 which almost for sure is English.

```
10 \ifx\l@russian\@undefined
11 \@nopatterns{Russian}
12 \adddialect\l@russian0
13 \fi
```

Now `\l@russian` 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 $\text{Lua}\LaTeX$. It matters whether these packages are loaded before or after `babel`. In the latter case or if these packages are not loaded at all, `russianb.ldf` ignores their effect and tries to provide some reasonable settings. In particular, `T2A` will be selected for Russian language if \LaTeX is running but `EU1` in case of $\text{Xe}\LaTeX$ and `EU2` in case of $\text{Lua}\LaTeX$.

`\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 `russianb.ldf` since it is overruled in `babel.def`. For example, after

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

as well as after

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

`\latinencoding` will be set to `T1`. After

```
\usepackage[english,russian]{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 $\text{Xe}\LaTeX$ and `EU2` for $\text{Lua}\LaTeX$, 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 $\text{Xe}\LaTeX$ or `EU2` for $\text{Lua}\LaTeX$ 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 `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 Russian 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 Russian phrase or vice versa.

We parse the `\cddp@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.

```

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

```

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{russian}`; the latter rises an error if `\cyrillicencoding` would not be defined by that time.

```

56 \AtBeginDocument{\@setcyrillicencoding}
57 \fi

```

`\Russian` For the sake of backward compatibility we keep the macro `\Russian` but redefine its meaning; now `\Russian` is simply an alias for `\selectlanguage{russian}`.
`\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 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 `\Russian` or the command `\foreignlanguage` defined in the `babel` core; their functionality is similar to `\selectlanguage{russian}` but they did not switch caption names, dates and shorthands.

```

58 \DeclareRobustCommand{\Russian}{\selectlanguage{\russian}}
59 \DeclareRobustCommand{\cyrillictext}{%
60   \fontencoding\cyrillicencoding\selectfont
61   \let\encodingdefault\cyrillicencoding
62   \expandafter\set@hyphenmins\russianhyphenmins
63   \language\l@russian}%
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 `russianb.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\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 of `\foreignlanguage{russian}`.

```
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 Russian 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 Russian language}
97 \fi\fi\fi
98 %\fi
99 }

```

7.4 Shorthands

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

```
100 \initiate@active@char{’}
```

`\mdqon` Obsolete: Active double quote can be both activated and deactivated at any time
`\mdqoff` 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{russian}{'‘}{\quotedblbase}
109 \declare@shorthand{russian}{'’}{\textquotedblleft}
110 \declare@shorthand{russian}{'<}{\guillemotleft}
111 \declare@shorthand{russian}{'>}{\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{russian}{'”}{\hskip\z@skip}
113 \declare@shorthand{russian}{'”~}{\textormath{\leavevmode\hbox{-}}{-}}
114 \declare@shorthand{russian}{'”=}{\nobreak-\hskip\z@skip}
115 \declare@shorthand{russian}{'’|}{%
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. Therefore \TeX looks for the next token after the first ‘-’, writes its meaning to `\russian@sh@next` and finally call for `\russian@sh@tmp`.

```
117 \declare@shorthand{russian}{'-}{%
118 \def\russian@sh@tmp{%
119 \if\russian@sh@next-\expandafter\russian@sh@emdash
120 \else\expandafter\russian@sh@hyphen\fi}%
121 \futurelet\russian@sh@next\russian@sh@tmp}
```

Two macros `\russian@sh@hyphen` and `\russian@sh@emdash` called by `\russian@sh@tmp` are defined below. The second of them has two parameters since it must gobble next two hyphen signs.

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

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

```
136 \ifx\cyrdash\undefined
137 \def\cyrdash{\hbox to.8em{--\hss--}}
138 \PackageInfo{babel}{\string\cyrdash\space is defined}
139 \fi
```

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.

```
140 \declare@shorthand{russian}{’},{\nobreak\hskip.2em\ignorespaces}
```

7.5 Switching to and from Russian

Now we define additional macros used to reset current language to Russian 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 Russian equivalents for Russian caption names.

\captionssrussian The macro `\captionssrussian` 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 Russian captions are called only if current language is Russian. However, the macros such as `\Russian` 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.


```

141 \if@uni@ode
142   %\captionsrussian@modern
143   \addto\captionsrussian{%
144     \def\prefacename{Предисловие}% [babel]
145     \def\refname{Список литературы}% [only article]
146     \def\abstractname{Аннотация}% [only article, report]
147     \def\bibName{Литература}% [only book, report]
148     \def\chaptername{Глава}% [only book, report]
149     \def\appendixname{Приложение}%

```

Note that two names for the Table of Contents can be used in Russian publications. For books (and reports) the second variant is appropriate, but for proceedings the first variant is preferred:

```

150   \@ifundefined{thechapter}
151     {\def\contentsname{Содержание}}%
152     {\def\contentsname{Оглавление}}%
153   \let\tocname=\contentsname
154   \def\listfigurename{Список иллюстраций}%
155   \def\listtablename{Список таблиц}%
156   \def\indexname{Предметный указатель}%
157   \def\authorname{Именной указатель}%
158   \def\figurename{Рис.}%
159   \def\tablename{Таблица}%
160   \def\partname{Часть}%
161   \def\enclname{вкл.}%
162   \def\ssname{исх.}%
163   \def\headtoname{вх.}%
164   \def\pagename{с.}% [letter]
165   \def\seename{см.}%
166   \def\alsoname{см. \ также}%
167   \def\proofname{Доказательство}% [amsthm]
168   \def\glossaryname{Словарь терминов}%
169   \def\acronymname{Аббревиатуры}% [glossaries] {Acronyms}
170   \def\lstlistingname{Листинг}% [listings] (the environment) {List-
ing}
171   \def\lstlistlistingname{Листинги}% [listings] (the "List of") {List-
ings}
172   \def\notesname{Замечки}% [endnotes] {Notes}
173   }%

```

Additional definitions for the package `nomenc1`:

```

174   %% =====
175   %% nomenc1
176   \ifdefined\nomname
177     \addto\captionsrussian{%
178       \def\nomname{Обозначения}%
179       \def\eqdeclaration#1{, см.\nobreakspace(#1)}%
180       \def\pagedeclaration#1{, стр.\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 code>}%
188 \@ifclassloaded{revtex4-1}
189 {%
190 \addto\captionsrussian{%
191 \def\lofname{\listfigurename}
192 \def\lotname{\listtablename}
193 \def\figuresname{Рисунки}%{Figures}%
194 \def\tablesname{Таблицы}%{Tables}%
195 \def\appendixesname{Приложения}%{Appendixes}%
196 \def\acknowledgmentsname{Благодарности}%{Acknowledgments}
197 \def\andname{и}%{and}
198 \def\@pacs@name{PACS коды: }%{PACS numbers: }%
199 \def\@keys@name{Ключевые слова: }%{Keywords: }%
200 \def\Dated@name{Дата: }%{Dated: }%
201 \def\Received@name{Получено }%{Received }%
202 \def\Revised@name{Исправленная версия }%{Revised }%
203 \def\Accepted@name{Принято }%{Accepted }%
204 \def\Published@name{Опубликовано }%{Published }%
205 }%
206 }-}%

```

Now we proceed to the ancient version in unicode encoding.

```

207 \def\captionsrussian@ancient{%
208 \def\prefacename{Предисловие}%
209 \def\refname{Примъчания}%
210 \def\abstractname{Аннотація}%
211 \def\bibName{Библиографія}%
212 \def\chaptername{Глава}%
213 \def\appendixname{Приложение}%
214 \@ifundefined{thechapter}
215   {\def\contentsname{Содержание}}%
216   {\def\contentsname{Оглавление}}%
217 \let\tocname=\contentsname
218 \def\listfigurename{Списокъ иллюстрацій}%
219 \def\listtablename{Списокъ таблицъ}%
220 \def\indexname{Предмътный указатель}%
221 \def\authorname{Именной указатель}%
222 \def\figurename{Рис.}%
223 \def\tablename{Таблица}%
224 \def\partname{Часть}%
225 \def\enclname{вкл.}%
226 \def\ssname{исх.}%
227 \def\headtoname{вх.}%
228 \def\pagename{с.}%
229 \def\seename{см.}%
230 \def\alsoname{см. ~также}%
231 \def\proofname{Доказательство}%
232 \def\glossaryname{Словарь терминов}%
233 \def\acronymname{Аббревиатуры}%
234 \def\lstlistingname{Листинг}%
235 \def\lstlistlistingname{Листинги}%
236 %\ifdefined\nomname
237 % see http://old_russian.academic.ru/8433/%D0%BE%D0%B1%D1%8A%CB%AB%D0%B0%D0%B2%D0%BB%D0%BB
238 \def\nomname{Обозначения}%{Объавлениж}% <-- #0465

```

```

239 %\fi
240 \def\notesname{Заметки}%
241 }
242 % =====
243 % RevTeX4 & RevTeX4-1
244 \@ifclassloaded{revtex4-1}
245 {%
246 \addto\captionsrussian@ancient{%
247 \def\lofname{\listfigurename}
248 \def\lotname{\listtablename}
249 \def\figuresname{Рисунки}{Figures}%
250 \def\tablesname{Таблицы}{Tables}%
251 \def\appendixesname{Приложения}{Appendixes}%
252 \def\acknowledgmentsname{Благодарности}{Acknowledgments}
253 \def\andname{}{and}
254 \def\@pacs@name{PACS коды: }{PACS numbers: }%
255 \def\@keys@name{Ключевые слова: }{Keywords: }%
256 \def\Dated@name{Дата: }{Dated: }%
257 \def\Received@name{Получено }{Received }%
258 \def\Revised@name{Исправленная версия }{Revised }%
259 \def\Accepted@name{Принято }{Accepted }%
260 \def\Published@name{Опубликовано }{Published }%
261 }%
262 }{%
263 % =====
264 \else
265 %\def\captionsrussian@modern{%
266 \addto\captionsrussian{%
267 \def\prefacename{%
268 {\cyr\CYRP\cyrr\cyre\cyrd\cyri\cyrs\cyrl\cyro\cyrv\cyri\cyre}}%
269 \def\refname{%
270 {\cyr\CYRS\cyrp\cyri\cyrs\cyro\cyrk\
271 \cyrl\cyri\cyrt\cyre\cyrr\cyra\cyrt\cyru\cyrr\cyrery}}%
272 \def\abstractname{%
273 {\cyr\CYRA\cyrn\cyrn\cyro\cyrt\cyra\cyrc\cyri\cyrya}}%
274 \def\bibname{%
275 {\cyr\CYRL\cyri\cyrt\cyre\cyrr\cyra\cyrt\cyru\cyrr\cyra}}%
276 \def\chaptername{\cyr\CYRG\cyrl\cyra\cyrv\cyra}}%
277 \def\appendixname{%
278 {\cyr\CYRP\cyrr\cyri\cyrl\cyro\cyrz\cyre\cyrn\cyri\cyre}}%
279 \@ifundefined{thechapter}%
280 {\def\contentsname{%
281 {\cyr\CYRS\cyro\cyrd\cyre\cyrr\cyrz\cyra\cyrn\cyri\cyre}}}%
282 {\def\contentsname{%
283 {\cyr\CYRO\cyrg\cyrl\cyra\cyrv\cyrl\cyre\cyrn\cyri\cyre}}}%
284 \let\tocname=\contentsname
285 \def\listfigurename{%
286 {\cyr\CYRS\cyrp\cyri\cyrs\cyro\cyrk
287 \ \cyri\cyrl\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyri\cyrishrt}}%
288 \def\listtablename{%
289 {\cyr\CYRS\cyrp\cyri\cyrs\cyro\cyrk
290 \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc}}%
291 \def\indexname{%
292 {\cyr\CYRP\cyrr\cyre\cyrd\cyrm\cyre\cyrt\cyrn\cyrery\cyrishrt}

```

```

293     \ \cyru\cyrk\cyra\cyrz\cyra\cyrt\cyre\cyrl\cyrsfts}}%
294 \def\authorname{%
295     {\cyr\CYRI\cyrm\cyre\cyrn\cyrn\cyro\cyrishrt
296     \ \cyru\cyrk\cyra\cyrz\cyra\cyrt\cyre\cyrl\cyrsfts}}%
297 \def\figurename{{\cyr\CYRR\cyri\cyrs.}}%
298 \def\tablename{{\cyr\CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyra}}%
299 \def\partname{{\cyr\CYRCH\cyra\cyrs\cyrt\cyrsfts}}%
300 \def\enclname{{\cyr\cyrv\cyrk\cyrl.}}%
301 \def\ccname{{\cyr\cyri\cyrs\cyrh.}}%
302 \def\headtoname{{\cyr\cyrv\cyrh.}}%
303 \def\pagename{{\cyr\cyrs.}}%
304 \def\seename{{\cyr\cyrs\cyrm.}}%
305 \def\alsoname{{\cyr\cyrs\cyrm.\ \cyrt\cyra\cyrk\cyrz\cyre}}%
306 \def\proofname{{\cyr\CYRD\cyro\cyrk\cyra\cyrz\cyra\cyrt
307     \cyre\cyrl\cyrsfts\cyrs\cyrt\cyrv\cyro}}%
308 \def\glossaryname{{\cyr\CYRS\cyrl\cyro\cyrv\cyra\cyrr\cyrsfts\
309     \cyrt\cyre\cyrr\cyrm\cyri\cyrn\cyro\cyrv}}%
310 \def\acronymname{\CYRA\cyrb\cyrb\cyrr\cyre\cyrv\cyri\cyra\cyrt\cyru\cyrr\cyrery}%
311 \def\lstlistingname{\CYRL\cyri\cyrs\cyrt\cyri\cyrn\cyrg}%
312 \def\lstlistlistingname{\CYRL\cyri\cyrs\cyrt\cyri\cyrn\cyrg\cyri}%
313 \def\nomname{\CYRO\cyrb\cyro\cyrz\cyrn\cyra\cyrch\cyre\cyrn\cyri\cyrya}%
314 \def\notesname{\CYRZ\cyra\cyrm\cyre\cyrt\cyrk\cyri}%
315 }%
316 \def\captionsrussian@ancient{%
317     \def\prefacename{%
318         {\cyr\CYRP\cyrr\cyre\cyrd\cyri\cyrs\cyrl\cyro\cyrv\cyrii\cyre}}%
319     \def\refname{%
320         {\cyr\CYRP\cyrr\cyri\cyrm\cyryat\cyrch\cyra\cyrn\cyrii\cyrya}}%
321     \def\abstractname{%
322         {\cyr\CYRA\cyrn\cyrn\cyro\cyrt\cyra\cyrc\cyrii\cyrya}}%
323     \def\bibname{%
324         {\cyr\CYRB\cyri\cyrb\cyrl\cyrii\cyro\cyrg\cyrr\cyra\cyrf\cyrii\cyrya}}%
325     \def\chaptername{{\cyr\CYRG\cyrl\cyra\cyrv\cyra}}%
326     \def\appendixname{%
327         {\cyr\CYRP\cyrr\cyri\cyrl\cyro\cyrz\cyre\cyrn\cyrii\cyre}}%
328     \@ifundefined{thechapter}%
329     {\def\contentsname{%
330         {\cyr\CYRS\cyro\cyrd\cyre\cyrr\cyrz\cyra\cyrn\cyrii\cyre}}}%
331     {\def\contentsname{%
332         {\cyr\CYRO\cyrg\cyrl\cyra\cyrv\cyrl\cyre\cyrn\cyrii\cyre}}}%
333     \let\tocname=\contentsname
334     % Список иллюстраций
335     \def\listfigurename{%
336         {\cyr\CYRS\cyrp\cyri\cyrs\cyro\cyrk\cyrhdsn\
337         \cyri\cyrl\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}
338     % Список таблиц
339     \def\listtablename{%
340         {\cyr\CYRS\cyrp\cyri\cyrs\cyro\cyrk\cyrhdsn\
341         \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrhdsn}}%
342     % Предметный указатель, Ъ (\cyryat) in X2 encoding only
343     \def\indexname{%
344         {\cyr\CYRP\cyrr\cyre\cyrd\cyrm\cyryat\cyrt\cyrn\cyrery\cyrishrt\space

```

```

345     \cyru\cyrk\cyra\cyrz\cyra\cyrt\cyre\cyrl\cyrsftsn}}%
346 \def\authorname{%
347     {\cyr\CYRI\cyrm\cyre\cyrn\cyrn\cyro\cyrisht\
348     \cyru\cyrk\cyra\cyrz\cyra\cyrt\cyre\cyrl\cyrsftsn}}%
349 \def\figurename{{\cyr\CYRR\cyri\cyrs.}}%
350 \def\tablename{{\cyr\CYRT\cyra\cyrb\cyrl\cyri\cyrc\cyra}}%
351 \def\partname{{\cyr\CYRCH\cyra\cyrs\cyrt\cyrsftsn}}%
352 \def\enclname{{\cyr\cyrv\cyrk\cyrl.}}%
353 \def\ccname{{\cyr\cyri\cyrs\cyrh.}}%
354 \def\headtoname{{\cyr\cyrv\cyrh.}}%
355 \def\pagename{{\cyr\cyrs.}}%
356 \def\seename{{\cyr\cyrs\cyrm.}}%
357 \def\alsoname{{\cyr\cyrs\cyrm.\ \cyrt\cyra\cyrk\cyrz\cyre}}%
358 \def\proofname{{\cyr\CYRD\cyro\cyrk\cyra\cyrz\cyra\cyrt
359     \cyre\cyrl\cyrsftsn\cyrs\cyrt\cyrv\cyro}}%
360 \def\glossaryname{{\cyr\CYRS\cyrl\cyro\cyrv\cyra\cyrr\cyrsftsn\
361     \cyrt\cyre\cyrr\cyrm\cyri\cyrn\cyro\cyrv}}% <-- Needs translation
362 \def\acronymname{\CYRA\cyrb\cyrb\cyrr\cyre\cyrv\cyri\cyra\cyrt\cyru\cyrr\cyrery}%
363 \def\lstlistingname{\CYRL\cyri\cyrs\cyrt\cyri\cyrn\cyrg}%
364 \def\lstlistlistingname{\CYRL\cyri\cyrs\cyrt\cyri\cyrn\cyrg\cyri}%
365 \def\nomname{\CYRO\cyrb\cyro\cyrz\cyrn\cyra\cyrch\cyre\cyrn\cyrii\cyrya}%
366 \def\notesname{\CYRZ\cyra\cyrm\cyre\cyrt\cyrk\cyri}%
367 }%
368 \fi

```

7.5.2 Date in Russian

`\daterussian` The macro `\daterussian` is used to reset the macro `\today` in Russian.

```

369 \if@uni@ode
370   \addto\daterussian{%
371     \def\today{\number\day~\ifcase\month\or
372     января\or
373     февраля\or
374     марта\or
375     апреля\or
376     мая\or
377     июня\or
378     июля\or
379     августа\or
380     сентября\or
381     октября\or
382     ноября\or
383     декабря\fi
384     \space \number\year~г.}}
385 \def\daterussian@ancient{%
386 \def\today{\number\day~\ifcase\month\or%
387 января\or
388 февраля\or
389 марта\or
390 апреля\or
391 мая\or
392 июня\or
393 июля\or

```

```

394 августа\or
395 сентября\or
396 октября\or
397 ноября\or
398 декабря\fi%
399 \space \number\year~r.}}
400 \else
401 \def\daterussian{%
402 \def\today{\number\day~\ifcase\month\or
403 \cyrya\cyrn\cyrv\cyra\cyrr\cyrya\or
404 \cyrf\cyre\cyrv\cyrr\cyra\cyrl\cyrya\or
405 \cyrm\cyra\cyrr\cyrt\cyra\or
406 \cyra\cyrp\cyrr\cyre\cyrl\cyrya\or
407 \cyrm\cyra\cyrya\or
408 \cyri\cyryu\cyrn\cyrya\or
409 \cyri\cyryu\cyrl\cyrya\or
410 \cyra\cyrv\cyrg\cyru\cyrs\cyrt\cyra\or
411 \cyrs\cyre\cyrn\cyrt\cyrya\cyrb\cyrr\cyrya\or
412 \cyro\cyrk\cyrt\cyrya\cyrb\cyrr\cyrya\or
413 \cyrn\cyro\cyrya\cyrb\cyrr\cyrya\or
414 \cyrd\cyre\cyrk\cyra\cyrb\cyrr\cyrya\fi
415 \space \number\year~\cyrg.}}
416 \def\daterussian@ancient{%
417 \def\today{\number\day~\ifcase\month\or
418 \cyrya\cyrn\cyrv\cyra\cyrr\cyrya\or
419 \cyrf\cyre\cyrv\cyrr\cyra\cyrl\cyrya\or
420 \cyrm\cyra\cyrr\cyrt\cyra\or
421 \cyra\cyrp\cyrr\cyre\cyrl\cyrya\or
422 \cyrm\cyra\cyrya\or
423 \cyrii\cyryu\cyrn\cyrya\or
424 \cyrii\cyryu\cyrl\cyrya\or
425 \cyra\cyrv\cyrg\cyru\cyrs\cyrt\cyra\or
426 \cyrs\cyre\cyrn\cyrt\cyrya\cyrb\cyrr\cyrya\or
427 \cyro\cyrk\cyrt\cyrya\cyrb\cyrr\cyrya\or
428 \cyrn\cyro\cyrya\cyrb\cyrr\cyrya\or
429 \cyrd\cyre\cyrk\cyra\cyrb\cyrr\cyrya\fi
430 \space \number\year~\cyrg.}}
431 \fi

```

7.5.3 Hyphenation patterns

Russian hyphenation patterns are automatically activated every time Russian 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 `russianb.lda`.

```

432 \providehyphenmins{\CurrentOption}{\tw@\tw@}
433 \providehyphenmins{russian}{\tw@\tw@}

```

7.5.4 Extra definitions

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

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

```
434 \addto\extrasrussian{\cyrillictext}
435 \addto\noextrasrussian{\latintext}
```

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

```
436 \addto\extrasrussian{\languageshorthands{russian}}
437 \addto\extrasrussian{\bbl@activate{''}}
438 \addto\noextrasrussian{\bbl@deactivate{''}}
```

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

```
439 \addto\extrasrussian{\bbl@frenchspacing}
440 \addto\noextrasrussian{\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 Russian 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`.

```
441 %\def\@alph#1{{\fontencoding{\latinencoding}\selectfont
442 % \ifcase#1\or
443 %   a\or b\or c\or d\or e\or f\or g\or h\or
444 %   i\or j\or k\or l\or m\or n\or o\or p\or
445 %   q\or r\or s\or t\or u\or v\or w\or x\or
446 %   y\or z\else\@ctrerr\fi}}%
447 %\def\@Alph#1{{\fontencoding{\latinencoding}\selectfont
```

```

448 % \ifcase#1\or
449 %   A\or B\or C\or D\or E\or F\or G\or H\or
450 %   I\or J\or K\or L\or M\or N\or O\or P\or
451 %   Q\or R\or S\or T\or U\or V\or W\or X\or
452 %   Y\or Z\else\@ctrerr\fi}%

```

We add new enumeration style for Russian 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, HRDSN, ERY, and SFTSN are skipped, as usual for such enumeration.

```

453 \def\Asbuk#1{\expandafter\russian@Alph\csname c@#1\endcsname}
454 \if@uni@ode
455   \def\russian@Alph#1{\ifcase#1\or
456     A\or Б\or В\or Г\or Д\or Е\or Ж\or
457     З\or И\or К\or Л\or М\or Н\or О\or
458     П\or Р\or С\or Т\or У\or Ф\or Х\or
459     Ц\or Ч\or Ш\or Щ\or Э\or Ю\or Я\else\@ctrerr\fi}
460 \else
461   \def\russian@Alph#1{\ifcase#1\or
462     \CYRA\or\CYRB\or\CYRV\or\CYRG\or\CYRD\or\CYRE\or\CYRZH\or
463     \CYRZ\or\CYRI\or\CYRK\or\CYRL\or\CYRM\or\CYRN\or\CYRO\or
464     \CYRP\or\CYRR\or\CYRS\or\CYRT\or\CYRU\or\CYRF\or\CYRH\or
465     \CYRC\or\CYRCH\or\CYRSH\or\CYRSHCH\or\CYREREV\or\CYRYU\or
466     \CYRYA\else\@ctrerr\fi}
467 \fi

```

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

```

468 \def\asbuk#1{\expandafter\russian@alph\csname c@#1\endcsname}
469 \if@uni@ode
470   \def\russian@alph#1{\ifcase#1\or
471     a\or б\or в\or г\or д\or е\or ж\or
472     з\or и\or к\or л\or м\or н\or о\or
473     п\or р\or с\or т\or у\or ф\or х\or
474     ц\or ч\or ш\or щ\or э\or ю\or я\else\@ctrerr\fi}
475 \else
476   \def\russian@alph#1{\ifcase#1\or
477     \cyra\or\cyrb\or\cyrv\or\cyrg\or\cyrd\or\cyre\or\cyrzh\or
478     \cyrz\or\cyri\or\cyrk\or\cyrl\or\cyrm\or\cyrn\or\cyro\or
479     \cyrp\or\cyrr\or\cyrs\or\cyrt\or\cyru\or\cyrf\or\cyrh\or
480     \cyrc\or\cyrch\or\cyrsh\or\cyrshch\or\cyrerev\or\cyryu\or
481     \cyrya\else\@ctrerr\fi}
482 \fi

```

Babel 3.9 has introduced a notion of a language attribute. An **ancient** attribute changes default behavior, which uses modern Russian spelling, an activates an alternative set of captions and date macros suitable for typesetting ancient Slavonic and Church books.

```

483 \bbl@declare@ttribute{russian}{ancient}{%
484 \PackageInfo{babel}{Russian attribute set to ancient}%
485 \let\captionrussian=\captionrussian@ancient
486 \let\daterussian=\daterussian@ancient
487 }

```


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

```
488 \@onlypreamble\captionssussian@ancient
489 \@onlypreamble\daterussian@ancient
```

7.7 Cyrillic math

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

```
490 %\DeclareRobustCommand{\No}{%
491 % \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.

```
492 %\RequirePackage{textmath}
493 % \@ifundefined{sym\cyrillicencoding letters}{}{%
494 % \SetSymbolFont{\cyrillicencoding letters}{bold}\cyrillicencoding
495 % \rmdefault\bfdefault\updefault
496 % \DeclareSymbolFontAlphabet\cyrmathrm{\cyrillicencoding letters}
```

And we need few commands to switch to different variants.

```
497 %\DeclareMathAlphabet\cyrmathbf\cyrillicencoding
498 % \rmdefault\bfdefault\updefault
499 %\DeclareMathAlphabet\cyrmathsf\cyrillicencoding
500 % \sfdefault\mddefault\updefault
501 %\DeclareMathAlphabet\cyrmathit\cyrillicencoding
502 % \rmdefault\mddefault\itdefault
503 %\DeclareMathAlphabet\cyrmathtt\cyrillicencoding
504 % \ttdefault\mddefault\updefault
505 %
506 %\SetMathAlphabet\cyrmathsf{bold}\cyrillicencoding
507 % \sfdefault\bfdefault\updefault
508 %\SetMathAlphabet\cyrmathit{bold}\cyrillicencoding
509 % \rmdefault\bfdefault\itdefault
510 %}
```

```
\sh We also define few math operator names according to Russian typesetting tradi-
\ch tions. Some math functions in Russian math books have names different from
\tg English writings. For example, sinh in Russian is called sh. Special consideration
\ctg needs the macro \th that conflicts with the text symbol \th defined in Latin 1
\arctg encoding:
\arcctg
\th
\cth
\cosec
511 \def\sh{\mathop{\operator@font sh}\nolimits}
512 \def\ch{\mathop{\operator@font ch}\nolimits}
513 \def\tg{\mathop{\operator@font tg}\nolimits}
514 \def\ctg{\mathop{\operator@font ctg}\nolimits}
515 \def\arctg{\mathop{\operator@font arctg}\nolimits}
516 \def\arcctg{\mathop{\operator@font arcctg}\nolimits}
517 \addto\extrarussian{%
518 \babel@save{\th}%
519 \let\ltx@th\th
```

```

520 \def\th{\textormath{\ltx@th}}%
521           {\mathop{\operator@font th}\nolimits}}%
522   }
523 \def\cth{\mathop{\operator@font cth}\nolimits}
524 \def\cosec{\mathop{\operator@font cosec}\nolimits}

```

```

\Prob Finally, we define some rare Russian mathematical symbols:
\Variance 525 \def\Prob{\mathop{\kern\z@\mathsf{P}}\nolimits}
\nod 526 \def\Variance{\mathop{\kern\z@\mathsf{D}}\nolimits}
\nok 527 \if@uni@ode
\NOD 528 \def\nod{\mathop{\mathrm{н.о.д.}}\nolimits}
\NOK 529 \def\nok{\mathop{\mathrm{н.о.к.}}\nolimits}
\Proj 530 \def\NOD{\mathop{\mathrm{НОД}}\nolimits}
531 \def\NOK{\mathop{\mathrm{НОК}}\nolimits}
532 \def\Proj{\mathop{\mathrm{Пр}}\nolimits}
533 \else
534 \def\nod{\mathop{\cyrmathrm{\cyrn.\cyro.\cyrd.}}\nolimits}
535 \def\nok{\mathop{\cyrmathrm{\cyrn.\cyro.\cyrk.}}\nolimits}
536 \def\NOD{\mathop{\cyrmathrm{\CYRN\CYRO\CYRD}}\nolimits}
537 \def\NOK{\mathop{\cyrmathrm{\CYRN\CYRO\CYRK}}\nolimits}
538 \def\Proj{\mathop{\cyrmathrm{\CYRP\cyrr}}\nolimits}
539 \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.

```
540 \ldf@finish{russian}
```

Change History

1.1a	General: use <code>\russianhyphenmins</code> to store the correct values 22	1.1e	General: Added closing brace to second argument of <code>\LdfInit</code> 9
	Use the new mechanism for dealing with active characters 14	1.1f	General: Add macro for thin space between initials 16
1.1b	General: Added switch to LWN encoding 23	1.1k	Added definitions of Cyrillic em-dash stuff and <code>thinspace</code> 9
1.1c	General: Replaced <code>\undefined</code> with <code>\@undefined</code> and <code>\empty</code> with <code>\@empty</code> for consistency with \TeX 9	1.1l	Added switch for doublequote shorthands 14
1.1d	General: Moved the definition of <code>\atcatcode</code> right to the beginning. 9	1.1m	General: replaced all <code>\penalty\@M</code> with <code>\nobreak</code> 9
	Now use <code>\ldf@finish</code> to wrap up 26	1.1n	General: Made not using <code>inputenc</code> a warning instead of an error . . . 14

1.1m	General: Now use <code>\providehyphenmins</code> to provide a default value	22	Warning is added if <code>\cyrdash</code> defined	16
1.1o	General: <code>\latintext</code> is already defined by the core of <code>babel</code>	13	<code>\latinencoding</code> : Removed <code>\latinencoding</code>	10
	<code>\textlatin</code> already defined by the core of <code>babel</code>	13	1.2a	
			General: Indentation of 1st paragraph removed	9
1.2	General: <code>\englishhyphenmins</code> is removed	22	1.2b	
	<code>\lat</code> removed	13	General: Renamed to <code>russianu</code> to work with <code>babel-beta 3.9</code>	9
	Added EU1 and EU2 encodings	14	1.3	
	Change definition of <code>\th</code> only for this language	25	General: Removed <code>\@alph</code> and <code>\@Alph</code>	23
	Check for LuaTeX	9	Removed switch for doublequote shorthands	14
	EU1 and EU2 encodings added	11	1.3b	
	Removed <code>\English</code> macro	13	General: Renamed to <code>russianb</code> to work with <code>babel 3.9</code>	9
	Removed LWN encoding	14	1.3c	
	Removed test for present of encoding files	11	General: Fix bug in <code>\daterussian</code>	9
	Unicode code-points added for LuaLaTeX	16, 21	1.3d	
			General: Fix bug in <code>\Proj</code>	9
			1.3e	
			General: Update documentation	9

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined> refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols		
<code>\''</code>	103	<code>\setcyrillicencoding</code>
<code>\-</code>	122	14, 34, 35, 56
<code>\@Acdash</code>	126, 130	<code>\@undefined</code>
<code>\@Alph</code>	447	10
<code>\@Bcdash</code>	127, 132	<code>\@uni@odetrue</code>
<code>\@Ccdash</code>	128, 134	8, 9
<code>\@SS</code>	105	
<code>\@alph</code>	441	<code>\sqcup</code>
<code>\@ctrerr</code>	446, 452, 459, 466, 474, 481	166, 270, 287, 290, 293, 296, 305, 308, 336, 340, 347, 357, 360
<code>\@ifclassloaded</code>	185, 188, 244	A
<code>\@ifpackageloaded</code>	82	<code>\AA</code>
<code>\@ifundefined</code>	150, 214, 279, 328, 493	72, 75
<code>\@keys@name</code>	199, 255	<code>\aa</code>
<code>\@nopatterns</code>	11	73, 74
<code>\@onlypreamble</code>	35, 36, 37, 38, 488, 489	<code>\abstractname</code>
<code>\@pacs@name</code>	198, 254	146, 210, 272, 321
		<code>\Accepted@name</code>
		203, 259
		<code>\acknowledgmentsname</code>
		196, 252
		<code>\acronymname</code>
		169, 233, 310, 362
		<code>\adddialect</code>
		12
		<code>\addto</code>
		143, 177, 190, 246, 266, 370, 434, 435, 436, 437, 438, 439, 440, 517
		<code>\allowhyphens</code>
		116
		<code>\alsoname</code>
		166, 230, 305, 357
		<code>\andname</code>
		197, 253
		<code>\appendixesname</code>
		195, 251
		<code>\appendixname</code>
		149, 213, 277, 326
		<code>\arcctg</code>
		516
		<code>\arctg</code>
		515
		<code>\Asbuk</code>
		453
		<code>\asbuk</code>
		468
		<code>\AtBeginDocument</code>
		56
		<code>\authorname</code>
		157, 221, 294, 346
		B
		<code>\babel@save</code>
		518

<code>\bbl@activate</code> .. 101, 437	<code>\CYRA</code> 273,	<code>\cyrery</code> 271,
<code>\bbl@allowhyphens</code> . 122	310, 322, 362, 462	292, 310, 344, 362
<code>\bbl@deactivate</code> 102, 438	<code>\cyra</code> 271, 273,	<code>\CYRF</code> 464
<code>\bbl@declare@ttribute</code>	275, 276, 281,	<code>\cyrf</code> .. 324, 404, 419, 479
..... 483	283, 287, 290,	<code>\CYRG</code> 276, 325, 462
<code>\bbl@frenchspacing</code> 439	293, 296, 298,	<code>\cyrg</code> 283, 311,
<code>\bbl@nonfrenchspacing</code>	299, 305, 306,	312, 324, 332,
..... 440	308, 310, 313,	363, 364, 410,
<code>\begingroup</code> 103	314, 320, 322,	415, 425, 430, 477
<code>\bfdefault</code>	324, 325, 330,	<code>\CYRH</code> 464
. 495, 498, 507, 509	332, 337, 341,	<code>\cyrh</code> 301,
<code>\bibname</code> 147, 211, 274, 323	345, 348, 350,	302, 353, 354, 479
	351, 357, 358,	<code>\cyrhdsn</code> . 336, 340, 341
	360, 362, 365,	<code>\CYRI</code> 295, 347, 463
	366, 403, 404,	<code>\cyri</code> .. 268, 270, 271,
	405, 406, 407,	273, 275, 278,
	410, 414, 418,	281, 283, 286,
	419, 420, 421,	287, 289, 290,
	422, 425, 429, 477	297, 298, 301,
	<code>\CYRB</code> 324, 462	309, 310, 311,
	<code>\cyrb</code> 290, 298,	312, 313, 314,
	310, 313, 324,	318, 320, 324,
	341, 350, 362,	327, 336, 337,
	365, 411, 412,	340, 341, 349,
	413, 414, 426,	350, 353, 361,
	427, 428, 429, 477	362, 363, 364,
	<code>\CYRC</code> 465	366, 408, 409, 478
	<code>\cyrc</code> 273, 287,	<code>\cyrii</code> 318,
	290, 298, 322,	320, 322, 324,
	337, 341, 350, 480	327, 330, 332,
	<code>\CYRCH</code> 299, 351, 465	337, 365, 423, 424
	<code>\cyrch</code> . 313, 320, 365, 480	<code>\cyrillicencoding</code> 19,
	<code>\CYRD</code> .. 306, 358, 462, 536	39, 43, 45, 48, 53,
	<code>\cyrd</code> .. 268, 281, 292,	55, 60, 61, 88, 90,
	318, 330, 344,	92, 94, 493, 494,
	414, 429, 477, 534	496, 497, 499,
	<code>\cyrdash</code> .. 131, 133,	501, 503, 506, 508
	135, 136, 137, 138	<code>\cyrillictext</code>
	<code>\CYRE</code> 462 59, 64, 81, 434
	<code>\cyre</code> 268, 271,	<code>\cyrishrt</code> . 287, 292,
	275, 278, 281,	295, 337, 344, 347
	283, 292, 293,	<code>\CYRK</code> 463, 537
	295, 296, 305,	<code>\cyrk</code> 270,
	307, 309, 310,	286, 289, 293,
	313, 314, 318,	296, 300, 305,
	327, 330, 332,	306, 314, 336,
	344, 345, 347,	340, 345, 348,
	348, 357, 359,	352, 357, 358,
	361, 362, 365,	366, 412, 414,
	366, 404, 406,	427, 429, 478, 535
	411, 414, 419,	<code>\CYRL</code> 275, 311,
	421, 426, 429, 477	312, 363, 364, 463
	<code>\CYREREV</code> 465	<code>\cyr1</code> 268, 271,
	<code>\cyrerev</code> 480	276, 278, 283,

287, 290, 293,	\cyrr	268,	362, 410, 425, 479
296, 298, 300,		271, 275, 278,	\CYRV 462
307, 308, 318,		281, 287, 292,	\cyrv . . 268, 276, 283,
324, 325, 327,		308, 309, 310,	300, 302, 307,
332, 337, 341,		318, 320, 324,	308, 309, 310,
345, 348, 350,		327, 330, 337,	318, 325, 332,
352, 359, 360,		344, 360, 361,	352, 354, 359,
404, 406, 409,		362, 403, 404,	360, 361, 362,
419, 421, 424, 478		405, 406, 411,	403, 404, 410,
\CYRM 463		412, 413, 414,	418, 419, 425, 477
\cyrm 292, 295,		418, 419, 420,	\CYRYA 466
304, 305, 309,		421, 426, 427,	\cyrya 273, 313,
314, 320, 344,		428, 429, 479, 538	320, 322, 324,
347, 356, 357,	\CYRS . . . 270, 281, 286,	289, 308, 330,	365, 403, 404,
361, 366, 405,		336, 340, 360, 464	406, 407, 408,
407, 420, 422, 478			409, 411, 412,
\cyrmathbf 497	\cyrS 268,		413, 414, 418,
\cyrmathit 501, 508	270, 286, 287,		419, 421, 422,
\cyrmathrm 496, 534,	289, 297, 299,		423, 424, 426,
535, 536, 537, 538	301, 303, 304,		427, 428, 429, 481
\cyrmathsf 499, 506	305, 307, 311,		\cyryat . . . 320, 342, 344
\cyrmathtt 503	312, 318, 336,		\CYRYU 465
\CYRN 463, 536, 537	337, 340, 349,		\cyryu . 287, 337, 408,
\cyrn 273,	351, 353, 355,		409, 423, 424, 480
278, 281, 283,	356, 357, 359,		\CYRZ 314, 366, 463
292, 295, 309,	363, 364, 410,		\cyrz 293, 296,
311, 312, 313,	411, 425, 426, 479		306, 313, 345,
320, 322, 327,	\cyrSftsn		348, 358, 365, 478
330, 332, 344,	. 293, 296, 299,		\CYRZH 462
347, 361, 363,	307, 308, 345,		\cyrzh . 278, 281, 305,
364, 365, 403,	348, 351, 359, 360		327, 330, 357, 477
408, 411, 413,	\CYRSH 465		
418, 423, 426,	\cyrsh 480		D
428, 478, 534, 535	\CYRSHCH 465		\Dated@name . . . 200, 256
\CYRO . . 283, 313, 332,	\cyrshch 480		\daterussian 370, 401, 486
365, 463, 536, 537	\CYRT 298, 350, 464		\daterussian@ancient
\cyro 268, 270,	\cyrT 271,		. 385, 416, 486, 489
273, 278, 281,	273, 275, 287,		\day . . 371, 386, 402, 417
286, 289, 295,	290, 292, 293,		\declare@shorthand
306, 307, 308,	296, 299, 305,		. 108, 109, 110,
309, 313, 318,	306, 307, 309,		111, 112, 113,
322, 324, 327,	310, 311, 312,		114, 115, 117, 140
330, 336, 340,	314, 322, 337,		\DeclareMathAlphabet
347, 358, 359,	341, 344, 345,		. 497, 499, 501, 503
360, 361, 365,	348, 351, 357,		\DeclareRobustCommand
412, 413, 427,	358, 359, 361,		. 58, 59, 65, 77, 490
428, 478, 534, 535	362, 363, 364,		\DeclareSymbolFontAlphabet
\CYRP 268, 278,	366, 405, 410,	 496
292, 318, 320,	411, 412, 420,		\DeclareTextCommand
327, 344, 464, 538	425, 426, 427, 479	 74, 75
\cyrp 270,	\CYRU 464		\DeclareTextFontCommand
286, 289, 336,	\cyru 271,	 81
340, 406, 421, 479	275, 293, 296,		\DeclareTextSymbolDefault
\CYRR 297, 349, 464	310, 345, 348,	 72, 73

<code>\def</code>	14, 15, 21, 79, 87, 89, 91, 93, 101, 102, 104, 105, 106, 118, 122, 123, 124, 125, 130, 132, 134, 137, 144, 145, 146, 147, 148, 149, 151, 152, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 178, 179, 180, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 207, 208, 209, 210, 211, 212, 213, 215, 216, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 238, 240, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 265, 267, 269, 272, 274, 276, 277, 280, 282, 285, 288, 291, 294, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 308, 310, 311, 312, 313, 314, 316, 317, 319, 321, 323, 325, 326, 329, 331, 335, 339, 343, 346, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 360, 362,	363, 364, 365, 366, 371, 385, 386, 401, 402, 416, 417, 441, 447, 453, 455, 461, 468, 470, 476, 511, 512, 513, 514, 515, 516, 520, 523, 524, 525, 526, 528, 529, 530, 531, 532, 534, 535, 536, 537, 538	<code>\figurename</code>	158, 222, 297, 349	
<code>\discretionary</code>	116	<code>\dq</code>	106	<code>\figuresname</code> ..	193, 249
<code>\edef</code> ..	16, 17, 43, 45, 48	<code>\else</code>	8, 40, 44, 47, 71, 88, 90, 92, 94, 120, 126, 127, 128, 264, 400, 446, 452, 459, 460, 466, 474, 475, 481, 491, 533	<code>\fontencoding</code>	60, 66, 78, 441, 447
<code>\enclname</code>	161, 225, 300, 352	<code>\encodingdefault</code> ..	61, 67, 79	<code>\futurelet</code>	121
<code>\endcsname</code> ..	71, 453, 468	<code>\endgroup</code>	104	G	
<code>\Eng</code>	70	<code>\English</code>	65, 70	<code>\glossaryname</code>	168, 232, 308, 360
<code>\englishhyphenmins</code> ..	68	<code>\eqdeclaration</code>	179	<code>\guillemotleft</code>	110
<code>\errmessage</code>	129	<code>\errmessage</code>	129	<code>\guillemotright</code> ...	111
<code>\exhyphenpenalty</code> ..	133	<code>\expandafter</code>	55, 62, 68, 71, 119, 120, 453, 468	H	
<code>\extrasrussian</code> ..	434, 436, 437, 439, 517	<code>\fi</code>	6, 9, 13, 20, 31, 46, 49, 57, 76, 85, 97, 98, 120, 129, 130, 132, 139, 182, 239, 368, 383, 398, 414, 429, 431, 446, 452, 459, 466, 467, 474, 481, 482, 491, 539	<code>\hbox</code>	113, 137
F		<code>\if</code>	119	<code>\headtoname</code>	163, 227, 302, 354
<code>\fi</code>	6, 9, 13, 20, 31, 46, 49, 57, 76, 85, 97, 98, 120, 129, 130, 132, 139, 182, 239, 368, 383, 398, 414, 429, 431, 446, 452, 459, 466, 467, 474, 481, 482, 491, 539	<code>\if@uni@ode</code>	2, 7, 28, 41, 83, 141, 369, 454, 469, 527	<code>\hskip</code> ..	112, 114, 130, 131, 133, 135, 140
<code>\ifcase</code> ..	2, 8, 9, 39, 42, 44, 176, 236	<code>\ifdefined</code> ..	2, 8, 9, 39, 42, 44, 176, 236	<code>\hss</code>	137
<code>\ifdim</code>	130, 132	<code>\ifmmode</code>	491	I	
<code>\ifmmode</code>	491	<code>\ifx</code>	10, 18, 71, 88, 90, 92, 94, 126, 127, 128, 136	<code>\if</code>	119
<code>\ifx</code>	10, 18, 71, 88, 90, 92, 94, 126, 127, 128, 136	<code>\ignorespaces</code>	131, 133, 135, 140	<code>\if@uni@ode</code>	2, 7, 28, 41, 83, 141, 369, 454, 469, 527
<code>\ignorespaces</code>	131, 133, 135, 140	<code>\indexname</code>	156, 220, 291, 343	<code>\ifcase</code> ..	371, 386, 402, 417, 442, 448, 455, 461, 470, 476
<code>\indexname</code>	156, 220, 291, 343	<code>\initiate@active@char</code>	100	<code>\ifdefined</code> ..	2, 8, 9, 39, 42, 44, 176, 236
<code>\initiate@active@char</code>	100	<code>\input</code>	55	<code>\ifdim</code>	130, 132
<code>\input</code>	55	<code>\itdefault</code>	502, 509	<code>\ifmmode</code>	491
<code>\itdefault</code>	502, 509	K		<code>\ifx</code>	10, 18, 71, 88, 90, 92, 94, 126, 127, 128, 136
<code>\kern</code>	116, 525, 526	<code>\l@english</code>	69	<code>\l@russian</code> ...	10, 12, 63
L		<code>\language</code>	63, 69	<code>\languageshorthands</code>	436
<code>\l@english</code>	69	<code>\lastskip</code>	130, 132	<code>\lat</code>	80
<code>\l@russian</code> ...	10, 12, 63				
<code>\language</code>	63, 69				
<code>\languageshorthands</code>	436				
<code>\lastskip</code>	130, 132				
<code>\lat</code>	80				

<code>\set@hyphenmins</code> . 62, 68	<code>\tempc@</code> 125, 128	<code>\unskip</code> 130, 132
<code>\SetMathAlphabet</code> 506, 508	<code>\tempx@</code> 124, 126, 127, 128	<code>\updefault</code> 495,
<code>\SetSymbolFont</code> 494	<code>\textcyrillic</code> 81	498, 500, 504, 507
<code>\sfdefault</code> 500, 507	<code>\textnumero</code> 491	
<code>\sh</code> 511	<code>\textormath</code> 113, 116, 520	V
<code>\space</code> 138, 344,	<code>\textquotedblleft</code> . 109	<code>\Variance</code> 526
384, 399, 415, 430	<code>\tg</code> 513	
<code>\string</code> 138	<code>\th</code> 518, 519, 520	X
	<code>\tocname</code> 153, 217, 284, 333	<code>\XeTeXrevision</code> . . . 9, 42
T	<code>\today</code> . 371, 386, 402, 417	
<code>\tablename</code>	<code>\ttdefault</code> 504	Y
. 159, 223, 298, 350	<code>\tw@</code> 432, 433	<code>\year</code> . . 384, 399, 415, 430
<code>\tablesname</code> . . . 194, 250		
<code>\tempa@</code> 125, 126	U	Z
<code>\tempb@</code> 125, 127	<code>\undefined</code> 136	<code>\z@</code> . . . 130, 132, 525, 526
		<code>\z@skip</code> . . . 112, 114, 133