The Russian Language
in the babel system

Version ?

Igor A. Kotelnikov*

Released ?

Contents

1 The Russian Language Definition File 2

2 Usage 2
   2.1 \texttt{BfX} 3
   2.2 \texttt{LuaBfX} 3
   2.3 \texttt{XeBfX} 4
   2.4 Modern and Ancient spelling 4

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

4 \TeX{}nical details 7

5 Known problems 7

6 Implementation 8
   6.1 Initial setup 8
   6.2 Output encoding 9
   6.3 Input encoding 13
   6.4 Shorthands 13
      6.4.1 Quotes 14
      6.4.2 Emdash, endash and hyphenation sign 14
   6.5 Switching to/from Russian 15
      6.5.1 Caption names 16
      6.5.2 Date in Russian 21
      6.5.3 Hyphenation patterns 22
      6.5.4 Extra definitions 23
   6.6 Alphabetic counters 23
   6.7 Cyrillic math 25
   6.8 Final settings 26

7 Change History 26

*E-mail: kia999 at mail dot ru.
1 The Russian Language Definition File

The file `russianb.ldf` 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 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 have 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.

---

\footnote{The file described in this document has the version ? and was last revised on ?.}
2.1 \LaTeX

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

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

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

Input encoding should be declared as option to the \texttt{inputenc} package. Known Cyrillic encodings include \texttt{cp866} (MS DOS), \texttt{cp1251} (Windows), \texttt{koi8-u} (UNIX) and their variants. Nowadays, this list is appended with \texttt{utf8} input encoding.

Output encodings (also known as font encodings) are declared as options to the \texttt{fontenc} package. Known Cyrillic encodings are \texttt{T2A}, \texttt{T2B}, \texttt{T2C}, \texttt{LCY}, and \texttt{X2}; \texttt{LWN} is excluded from Russian support starting version 1.2 of \texttt{russianb.ldf} since \texttt{LWN} is excluded from the \texttt{cyrillic} bundle of related files.

2.2 \texttt{LuaLaTeX}

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

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

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

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

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

```latex
\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 available 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

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

\section*{2.3 \LaTeXX}

In \LaTeXX, 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 \LaTeXX to work like a 8-bit \LaTeX engine:

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

\LaTeXX can use a different input encoding but it always uses the Unicode internally, so that `\XeTeXinputencoding` performs a conversion of the input stream into Unicode; see tex.stackexchange.com/questions/36188/do-xetex-and-latex-always-use-unicode.

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

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

\section*{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,

```latex
\usepackage[english, russian]{babel}
\languageattribute{russian}{ancient}
```
Setting the \texttt{ancient} attribute changes the built-in strings (caption names) and a date format. For example, the bibliography will be entitled as ‘Ирпеппаръяп’ by default and as ‘Баfжioraртiп’ if the Russian language attribute is set to \texttt{ancient}. Same result can be achieved using a modifier as follows:

\texttt{\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).

\section{User’s commands}

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

Regarding local typography, the macro \texttt{selectlanguage(russian)} switches to the Russian language, with the following effects:

1. Russian hyphenation patterns are made active;
2. \texttt{\today} prints the date in Russian;
3. the caption names are translated into Russian;
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 \texttt{Lua\TeX} and \texttt{Xe\TeX} 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. \texttt{\cyrdash}) can be typeset in any language using special macros enlisted in table 1.

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

Since Russian has its own numbering system, \texttt{russianb.1df} adds macros \texttt{\asbuk{⟨counter⟩}} and \texttt{\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 \texttt{\guillemotleft} and \texttt{\guillemotright} which work in \texttt{\LaTeX\ 2ε} and \texttt{Plain\TeX}.
2. German quotation marks can be entered using the commands \texttt{\glqq} and \texttt{\grqq} which work in \texttt{\LaTeX\ 2ε} and \texttt{Plain\TeX}.

The macro \texttt{\Russian} is now defined as an alias for \texttt{selectlanguage(russian)}, and its “opponent” \texttt{\English}, existed in \texttt{russianb.1df} 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 \texttt{\textcyrillic{⟨text⟩}} is intended to typeset small chunks of text in Russian; it is essentially an alias for \texttt{\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 \texttt{russianb.ldf}

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

Note that the standard soft hyphen \- is equivalent to \texttt{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 encoding (TU) as ‘<’ and ‘>’ characters in 7-bit Cyrillic font encodings (OT2 and LWN).
In Unicode encoding, cyrdashes and quotes can be typed as single character if your text editor allows inserting 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 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, \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 TEXnical details

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

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

Some inconsistencies of prior versions of russianb.ldf was also overcame in the version 1.2. Those users who used \selectlanguage macro, defined in the core babel system, to switch between different languages should not worry. However, the macros \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 vise versa delete all .aux, .toc, .lot, .lot 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:
None of Cyrillic font encoding has ‘iotated E’ (Ѥ, ѥ). When running legacy engines you are advised to substitute it with \CYRIE, \cyrie (Є, є):

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

The dash might change its length after switching the current language with \selectlanguage. Legacy engines (latex.exe, pdflatex.exe) take the dash symbol (—) from same code point 22 of a font file but from different fonts for Cyrillic and Latin languages. Cyrillic fonts take care that Cyrillic dash is 20% shorter than Latin one. Unicode engines (xelatex.exe, lualatex.exe) take the dash from the code point x2022, but may substitute the font dash with fake symbol which is shorter. See discussion at https://tex.stackexchange.com/questions/294178/what-about-cyrdash-in-eu1-and-eu2-encodings.

6 Implementation

6.1 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.

```
\ProvidesLanguage{russian}
\LdfInit{russian}{captionsrussian}
```

First, we check if LuaL\TeX or XeL\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\TeXe to handle various Cyrillic input encoding. With the advent of Unicode L\TeX is moving to universal input encoding, so we consider these \cyr... commands as obsolete. They are preserved though for backward compatibility in case if L\TeX or PDFL\TeX are running.

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

\begin{verbatim}
12 \ifx\l@russian\@undefined
13 \@@nopatterns{Russian}
14 \addialect\l@russian0
15 \fi
\end{verbatim}

Now \l@russian is always defined.

\section{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 \texttt{fontenc} package in case of \LaTeX{} and by \texttt{fontspec} package in case of \texttt{LuaLaTeX}. It matters whether these packages are loaded before or after \texttt{babel}. In the latter case or if these packages are not loaded at all, \texttt{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 TU in case of \texttt{XeLaTeX} or \texttt{LuaLaTeX}.

\texttt{\latinencoding} The macro \texttt{\latinencoding} keeps the name of Latin encoding. It is defined in \texttt{babel.def} and is wrapped into \texttt{\AtBeginDocument} to allow for late loading \texttt{fontenc}. Therefore it does not matter whether \texttt{babel} is loaded before or after the \texttt{fontenc}. As of version 1.2, definition of \texttt{\latinencoding} was removed from \texttt{russianb.ldf} since it is overruled in \texttt{babel.def}. For example, after

\begin{verbatim}
\usepackage[T1,T2A]{fontenc}
\usepackage[english,russian]{babel}
\end{verbatim}

as well as after

\begin{verbatim}
\usepackage[english,russian]{babel}
\usepackage[T1,T2A]{fontenc}
\end{verbatim}

\texttt{\latinencoding} will be set to T1. After

\begin{verbatim}
\usepackage[english,russian]{babel}
\end{verbatim}

\texttt{\latinencoding} will be OT1.

In Unicode mode, the package \texttt{fontspec} should be loaded instead of \texttt{fontenc} to make font preparation; \texttt{fontspec} sets current encoding (kept in \texttt{\cf@encoding}) to TU, and the \texttt{babel} package sets the macro \texttt{\latinencoding} to \texttt{\cf@encoding}. Since \texttt{babel} scan for value \texttt{\cf@encoding} within \texttt{\AtBeginDocument}, \texttt{\latinencoding} will be set to TU for \texttt{XeLaTeX} or \texttt{LuaLaTeX} no matter which of the packages, \texttt{babel} or \texttt{fontspec} is loaded first.
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 \texttt{LWN}. Correct 8-bit Cyrillic encodings are \texttt{T2A} (default for 8-bit compilers), \texttt{T2B}, \texttt{T2C}, \texttt{LCY} and \texttt{X2}. Correct utf8 encodings are \texttt{TU} (default for \LaTeX{}X and \texttt{Lua\LaTeX{}X}), \texttt{EU1} (obsolete, formerly used for \texttt{Xe\LaTeX{}X}), \texttt{EU2} (obsolete, formerly used for \texttt{Lua\LaTeX{}X}).

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

Remember that for the Russian language, the \texttt{T2A} encoding is better than \texttt{X2}, because \texttt{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 \texttt{\cdp@list} containing encodings known to \LaTeX{}X in the order they have been loaded by the time \texttt{babel} is called. We set the \texttt{\cyrillicencoding} to the last loaded encoding in the list of supported Cyrillic encodings: \texttt{OT2}, \texttt{LCY}, \texttt{X2}, \texttt{T2C}, \texttt{T2B}, \texttt{T2A}. In Unicode mode, \texttt{\cyrillicencoding} is set to \texttt{TU} by \texttt{fontspec}. Nevertheless here we provide similar definitions; 8-bit encodings are kept for Unicode compilers (\texttt{Lua\LaTeX{}X} and \texttt{Xe\LaTeX{}X}) since they can run in compatibility (8-bit) mode.

\begin{verbatim}
\def\@setcyrillicencoding{%
  \def\sce@a##1##2{%
    \edef\sce@b{##1}%
    \edef\sce@c{##2}%
    \ifx\sce@b\sce@c
      \let\cyrillicencoding\sce@c
    \fi}%
  \def\cdp@elt##1##2##3##4{%
    \sce@a{##1}{OT2}%
    \sce@a{##1}{LCY}%
    \sce@a{##1}{X2}%
    \sce@a{##1}{T2C}%
    \sce@a{##1}{T2B}%
    \sce@a{##1}{T2A}%
    \if@uni@ode
      \%\sce@a{##1}{EU1}%
      \%\sce@a{##1}{EU2}%
    \sce@a{##1}{TU}%
  \fi}%
  \cdp@list
  \ifx\cyrillicencoding\undefined
    \@setcyrillicencoding
  \fi
  \@onlypreamble\@setcyrillicencoding
  \@onlypreamble\sce@a
  \@onlypreamble\sce@b
  \@onlypreamble\sce@c
}\end{verbatim}

The last lines are to free the memory occupied by the macros \texttt{\@setcyrillicencoding} and \texttt{\sce@x} that are useless in the document. The contents of \texttt{\@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.

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

As a 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.

\AtBeginDocument{\@setcyrillicencoding}

\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}.

\DeclareRobustCommand{\Russian}{\selectlanguage{\russian}}

\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 "<", ">", ",", "", "---", c.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.
Since version v.1.2, the \English macro and its alias \Eng were removed as a reasonable place for defining these macros would be englishb.ldf. Note also that these macros are absent from russianb.ldf’s counterpart of the package polyglossia, analog of babel for Xe\LaTeX.

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 main 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.

The macro \textcyrillic takes an argument which is then typeset using the \cyrillictext declaration.

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).

\textcyrillic \{⟨text⟩\}

\textcyrillic takes an argument which is then typeset using the \cyrillictext declaration.

Next chunk 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 X 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).
6.3 Input encoding

We do not assume any default input encoding of the source file, so the `inputenc` package should be explicitly loaded by `\usepackage{...}(inputenc)` before `babel`. Note that default font encoding `T2A` for legacy 8-bit latex engines fits well enough to Russian version of Windows ANSI encoding which is almost equivalent to `cp1251` input encoding.

Should we wrap this chunk into `AtBeginDocument`? Note also that `inputenc` has its own check whether it is correctly called. Also, it is difficult to imagine that TU encoding is not declared in case of unicode engines. Hence, we remove this check in `babel` starting since version 1.3j.

6.4 Shorthands

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

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

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

\begingroup \catcode`\"12 \endgroup
6.4.1 Quotes

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

\begin{verbatim}
113 \declare@shorthand{russian}{“”}{\quotedblbase}
114 \declare@shorthand{russian}{“”}{\textquotedblleft}
115 \declare@shorthand{russian}{“”}{\guillemotleft}
116 \declare@shorthand{russian}{“”}{\guillemotright}
\end{verbatim}

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.

\begin{verbatim}
117 \declare@shorthand{russian}{“”}{\hskip\z@skip}
118 \declare@shorthand{russian}{“”}{\textormath{\leavevmode\hbox{-}}{-}}
119 \declare@shorthand{russian}{“”}{\nobreak\-\hskip\z@skip}
120 \declare@shorthand{russian}{“”}{%}
121 \textormath{\nobreak\discretionary{-}{}{\kern.03em}\allowhyphens}{%}
\end{verbatim}

6.4.2 Emdash, endash and hyphenation sign

To distinguish between ‘- and ‘--- we must check whether the next token after - 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.

\begin{verbatim}
122 \declare@shorthand{russian}{“”}{%}
123 \def\russian@sh@tmp{%
124 \if\russian@sh@next-\expandafter\russian@sh@emdash
125 \else\expandafter\russian@sh@hyphen\fi%
126 \futurelet\russian@sh@next\russian@sh@tmp}
\end{verbatim}

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.

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

\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.

\begin{verbatim}
129 \def\cdash#1#2#3{\def\tempx@{#3}%
130 \def\tempa@{-}
131 \ifx\tempx@\tempa@\@Acdash
132 \else\ifx\tempx@\tempb@\@Bcdash
133 \else\@Ccdash\fi%}
\end{verbatim}
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.

\@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.

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

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.

\cyrdash The \cyrdash macro is defined in Cyrillic font encodings (LCY, T2*, OT2, and X2) by means of \DeclareTextSymbol. In T2* encodings \cyrdash refers to same code point 22 as \textemdash does so that these two macros are equivalent. However the dash at the code point 22 have different length in different fonts. The dash in Cyrillic fonts LH is 20% shorter as compared to Latin fonts such as CM (Computer Modern). As a result, the dash typed by the ligature --- or its variations mentioned in Table 1 might change its length after \selectlanguage.

The \cyrdash macro is not available in Latin encodings such as T1. Therefore an explicit or implicit call for \cyrdash when current language is English causes an error. For such a case, we provide a fake default. A standard check such as \ifx\cyrdash\undefined ...\fi fails to detect absent definitions for Latin encodings since the \cyrdash macro is in fact defined. Therefore we use the \ProvideTextCommandDefault method:

6.5 Switching to/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.

6.5.1 Caption names

First, we define Russian equivalents for Russian caption names.

The macro \captionsrussian 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 their Unicode code-points, the \hbox{ab\ldots} notation is not used since it causes error at compilation time in case if \LaTeX is running and utf8 input encoding is not declared.

146 \if@uni@ode
147 \captionsrussianmodern
148 \addto\captionsrussian{%
149 \def\prefacename{Предисловие}% [babel]
150 \def\refname{Список литературы}% [only article]
151 \def\abstractname{Аннотация}% [only article, report]
152 \def\bibname{Литература}% [only book, report]
153 \def\chaptername{Глава}% [only book, report]
154 \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:

155 \ifdefined{thechapter}
156 \{\def\contentsname{Содержание}}% [babel]
157 \{\def\contentsname{Оглавление}}% [only article]
158 \let\tocname=\contentsname
159 \{\def\listfigurename{Список иллюстраций}}% [only article, report]
160 \{\def\listtablename{Список таблиц}}% [only book, report]
161 \{\def\indexname{Предметный указатель}}%
162 \{\def\authorname{Именной указатель}}%
163 \{\def\figurename{Рис.}}%
164 \{\def\tablename{Таблица}}%
165 \{\def\partname{Часть}}%
166 \{\def\enclname{вкл.}}%
167 \{\def\ccname{исх.}}%
168 \{\def\headtoname{вх.}}%
169 \{\def\pagename{с.}}% [letter]
170 \{\def\seename{см.}}%
171 \{\def\also{см. \ также}}%
172 \{\def\proofname{Доказательство}}% [amsthm]
173 \{\def\glossaryname{Словарь терминов}}%
174 \{\def\acronymname{Аббревиатуры}}% [glossaries] {Acronyms}
175 \{\def\listlistingname{Листинг}}% [listings] (the environment) {Listings}
176 \{\def\lstlistingname{Листинги}}% [listings] (the "List of") {Listings}
177 \{\def\notesname{Заметки}}% [endnotes] {Notes}
178 }

16
Additional definitions for the package `nomencl`:

```latex
\% Additional captions for the revtex class.
```

Now we proceed to the ancient version in Unicode encoding.
6.5.2 Date in Russian

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

```latex
\daterussian
```

21

6.5.3 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.ldf.

\providehyphenmins{\CurrentOption}{\tw@\tw@}
6.5.4 Extra definitions

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.

\addto\extrasrussian{\cyrillictext}
\addto\noextrasrussian{\latintext}

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

\addto\extrasrussian{\languageshorthands{russian}}
\addto\extrasrussian{\bbl@activate{"}}
\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.

\addto\extrasrussian{\bbl@frenchspacing}
\addto\noextrasrussian{\bbl@nonfrenchspacing}

6.6 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 \texttt{X2} do have Latin letters. When using the \texttt{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 \texttt{microtype} package after the \appendix declaration (which resets \texttt{\thechapter} to \@Alph\c@chapter). Most languages do not reset \@alph and \@Alph macros and only \texttt{ukrainian} and \texttt{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 \texttt{smartref} is loaded and a \texttt{sref} occurs after the \appendix declaration which resets \texttt{\thechapter} to \@Alph\c@chapter.

\def\@alph#1{{\fontencoding{\latinencoding}\selectfont}}
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.

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

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

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

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

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

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.

534 \bbl@declareattribute{russian}{ancient}{%
535 \PackageInfo{babel}{Russian attribute set to ancient}%
536 \let\captionsrussian=\captionsrussian@ancient
537 \let\daterussian=\daterussian@ancient }

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

539 \@onlypreamble\captionsrussian@ancient
540 \@onlypreamble\daterussian@ancient

6.7 Cyrillic math

For compatibility with older Russian packages we could define the \textnumero macro. However the Russian number sign is now superseded with \textnumero. Moreover, it can be found on the keyboard. Therefore we discard \textnumero since v.1.2.

541 %\DeclareRobustCommand{\textnumero}{%
542 % \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.

543 %\RequirePackage{textmath}
544 % \ifundef{sym\cyrillicencoding letters}{%}
545 % \SetSymbolFont{\cyrillicencoding letters}{bold}{\cyrillicencoding}
546 % \rmdefault\bfdefault\updefault
547 % \DeclareSymbolFontAlphabet\cyrmathrm{\cyrillicencoding letters}

And we need few commands to switch to different variants.

548 %\DeclareMathAlphabet\cyrmathbf{\cyrillicencoding}
549 % \rmdefault\bfdefault\updefault
550 %\DeclareMathAlphabet\cyrmaths{\cyrillicencoding}
551 % \sfdefault\mddefault\updefault
552 %\DeclareMathAlphabet\cyrmathit{\cyrillicencoding}
553 % \rmdefault\mddefault\itdefault
554 %\DeclareMathAlphabet\cyrmathtt{\cyrillicencoding}
555 % \ttdefault\mddefault\updefault
556 %
557 %\SetMathAlphabet{\cyrillicencoding}
558 % \sfdefault\bfdefault\updefault
559 %\SetMathAlphabet{\cyrillicencoding}
560 % \rmdefault\bfdefault\itdefault

\sh \ch \ctg \arctg \arccotg \th \cth \cosec

We also define few math operator names according to Russian typesetting traditions. Some math functions in Russian math books have names different from English writings. For example, \text{sinh} in Russian is called \text{sh}. Special consideration needs the macro \text{\th} that conflicts with the text symbol \text{\th} defined in Latin 1 encoding:

562 \def\sinh{\mathop{\operator@font sh}\nolimits}
563 \def\cosec{\mathop{\operator@font ch}\nolimits}
Finally, we define some rare Russian mathematical symbols:

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

6.8 Final settings

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

7 Change History

1.1a  encoding ................. 21

General: use \russianhyphenmins to store the correct values .... 21 1.1c
Use the new mechanism for dealing with active characters 12

1.1b  General: Added switch to LWN
1.1d  General: Moved the definition of `\atcatcode` right to the
beginning. 7
Now use `\ldf@finish` to wrap up 25
Now use `\LdfInit` to perform initial checks 7

1.1e  General: Added closing brace to second argument of `\LdfInit` . 7

1.1f  General: Add macro for thin space between initials . 14
Added definitions of Cyrillic emdash stuff and thinspace . 7
Added switch for doublequote shorthands . 12

1.1k  General: replaced all `\penalty\@M` with `\nobreak` . 7

1.1l  General: Made not using inputenc a warning instead of an error . 11

1.1m  General: Now use `\providehyphenmins` to provide a default value . 21

1.1o  General: `\latintext` is already defined by the core of babel . 11
`\textlatin` already defined by the core of babel . 11

1.2  `\latinencoding`: Removed
`\latinencoding` . 8
General: `\englishhyphenmins` is removed . 21
`\latex` removed . 11
Added EU1 and EU2 encodings . 11
Change definition of `\th` only for this language . 24
Check for LuaTeX . 7
EU1 and EU2 encodings added . 9
Removed `\English` and `\Eng` macros . 10

1.2a  General: Indentation of 1st paragraph removed . 7

1.2b  General: Renamed to russian to work with babel-beta 3.9 . 7

1.2c  General: Renamed to russianb to work with babel 3.9 . 7

1.3  General: `\englishhyphenmins` is removed . 7
Removed switch for doublequote shorthands . 12

1.3a  General: Removed `\alph` and `\Alph` . 22

1.3b  General: Fixed bug in `\daterussian` . 7

1.3c  General: Fixed bug in `\Proj` . 7

1.3d  General: Update documentation . 7

1.3e  General: Update documentation for generating by `pdflatex.exe` . 7

1.3f  General: Added support for revtex4 and revtex4-1 classes . 7

1.3g  General: `\cyrdash` is redefined . 7
Bugs fixed in captions for revtex4 and revtex4-1 classes . 7
Revtex captions fixed . 16

1.3h  General: Default for `\cyrdash` is provided . 14
Removed check if inputenc is loaded . 11
TU encoding added . 9

8 Index

Numbers written in dark blue refer to the page where the corresponding entry is described; numbers in black roman refer to the code lines where the entry is used.

Symbols  
\- ................. 127 \@Alph ............... 498
\" ................. 108 \@Acirclogdash ...... 131, 135 \@Acirclogdash ...... 132, 137

27