The Ukrainian Language in the babel system

Version 1.4e

Sergiy M. Ponomarenko*

Released 2020/10/14

Contents

1 The Ukrainian Language Definition File

The file `ukraineb.ldf`[1] is the source file for the Ukrainian Language Definition file `ukraineb.ldf` to be loaded by the babel package with the option ukrainian. It's based on the Russian language definition file `russianb.ldf` derived by Igor A. Kotelnikov.

2 Usage

Typesetting Ukrainian 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.

*E-mail: sergiy dot ponomarenko at gmail dot com.

[1]The file described in this document has the version number 1.4e and was last revised on 2020/10/14.

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

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

Since earlier versions babel did not support XeLaTeX (at least for some languages including Ukrainian), 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.

2.1 LaTeX

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

```
\usepackage[T1,T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[english,ukrainian]{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 Ukrainian support of `ukraineb.ldf` since `LWN` is excluded from the `cyrillic` bundle of related files.

2.2 LuaLaTeX

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

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

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

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

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

The \defaultfontfeatures declares default font features for subsequent \setmainfont (which sets romanic fonts), \setsansfont (sans-serif) and \setmonofont (monospaced font). Font features can be set up on per font bases; for example

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

2.3 XeLaTeX

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

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

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

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

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

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{ukrainian} switches to the Ukrainian language, with the following effects:

1. Ukrainian hyphenation patterns are made active;
2. \today prints the date in Ukrainian;
3. the caption names are translated into Ukrainian (LaTeX only);
4. emdash typed by the ligature --- might be 20% shorter when Ukrainian is the current language; the result depends on the current encoding; --- always produce long emdash in LuaTeX and XeTeX since these engines use same encodings for all languages;

5. emdash typed by the ligature ’’--- in Ukrainian is 20% shorter, however the ligature ’’--- might not be defined in other languges; a shorter emdash (i.e. \cyrdash) can be typeset in any language using special macros enlisted in table 1.

Additional commands are provided to typeset quotes:

1. French quotation marks can be entered using the commands \guillemotleft and \guillemotright which work in LaTeX 2ε and PlainTeX.
2. German quotation marks can be entered using the commands \glqq and \grqq which work in LaTeX 2ε and PlainTeX.

The macro \Ukrainian defined as an alias for \selectlanguage{ukrainian}, and its “opponent” \English, existed in ukraineb.ldf has been removed since the Ukrainian 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 Ukrainian; it is essentially an alias for \foreignlanguage{ukrainian}{⟨*text*⟩}.

3.1 Active character

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

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

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

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

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

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

By default, active double quote is switched on. It can be switched off any time using \shorthandoff{''} and the switched on again using \shorthandon{''}.

3.2 Math commands

The ukraineb.ldf defines few macros that can be used independently of current language. These are macros to be used in math mode to type the names of trigonometric functions common for Ukrainian 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.

4 TEXnical details

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

Nowadays, Unicode makes \cyr<letter> 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<letter> are still kept for LaTeX, but they are bypassed if LuaLaTeX or XeLaTeX are detected.

5 Known problems

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

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.

```
\LdfInit{ukrainian}{captionsukrainian}
```

First, we check if LuaLaTeX or XeLaTeX is running. If so, we set boolean key \if@ukr@uni@ode to true. It will be used to eliminate \cyr... commands, which were introduced in LaTeX2e to handle various Cyrillic input encoding. With the advent of Unicode LaTeX is moving to universal input encoding, so we consider these \cyr... commands as obsolete. They are preserved though for backward compatibility in case if LaTeX or PDFLaTeX are running.

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

```
\ifdefined\if@ukr@uni@ode
    \PackageError{babel}{if@ukr@uni@ode already defined.\MessageBreak
        Please contact author of ukraineb.ldf}
    \relax
\fi
```

```
\newif\if@ukr@uni@ode
\ifdefined\luatexversion
    \@ukr@uni@odetrue
\else
    \ifdefined\XeTeXrevision
        \@ukr@uni@odetrue
    \fi
\fi
```

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

```
\ifx\l@ukrainian\@undefined
    \@nopatterns{Ukrainian}
  \adddialect\l@ukrainian0
\fi
```

Now `\l@ukrainian` is always defined.

6.2 Output encoding

We need to know font encoding that is supposed to be active at the end of the babel package. Default font encoding, set by LaTeX core, is `OT1`. This can be changed by the `fontenc` package in case of LaTeX and by `fontspec` package in case of LuaLaTeX. It matters weather these packages are loaded before of after babel. In the latter case or if these packages are not loaded at all, `ukraineb.ldf` ignores their effect and tries to provide some reasonable settings. In particular, `T2A` will be selected for Ukrainian language if LaTeX is running but `EU1` in case of XeLaTeX and `EU2` in case of LuaLaTeX.

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

`\cyrillicencoding`

There is a limited list of encodings appropriate for Cyrillic text. We will look which of them is declared and keep its name in the macro `\cyrillicencoding`. Correct (but obsolete and now deleted) 7-bit Cyrillic encoding is `LWN`. Correct 8-bit Cyrillic encodings are `T2A` (default for 8-bit compilers), `T2B`, `T2C`, `LCY` and `X2`. Correct utf8 encodings are `TU` (default for XeLaTeX and LuaLaTeX), `EU1` (obsolete, formerly used for XeLaTeX), `EU2` (obsolete, formerly used 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 Ukrainian 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 Ukrainian phrase or vice versa.

We parse the `\cdp@list` containing encodings known to LaTeX in the order they have been loaded by the time babel is called. 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, `\cyrillicencoding` is set to `TU` by `fontspec`. Nevertheless here we provide similar definitions; 8-bit encodings are kept for Unicode compilers (LuaLaTeX and XeLaTeX) since they can run in compatibility (8-bit) mode.

```
\def\@setcyrillicencoding{%
    \def\sce@a##1##2{%
        \edef\sce@b{##1}%
        \edef\sce@c{##2}%
        \ifx\sce@b\sce@c
            \let\cyrillicencoding\sce@c
        \fi}%
    \def\cdp@elt##1##2##3##4{%
        \sce@a{##1}{OT2}%
        \sce@a{##1}{LCY}%
        \sce@a{##1}{X2}%
        \sce@a{##1}{T2C}%
```

```
        \sce@a{##1}{T2B}%
        \sce@a{##1}{T2A}%
        \if@ukr@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
```

The last lines are to free the memory occupied by the macros `\@setcyrillicencoding` and `\sce@x` that are useless in the document. The contents of `\@begindocumenthook` is cleared automatically.

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

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

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

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

`\Ukrainian` `\cyr` `\cyrillictext`

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

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

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

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

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

```
\DeclareRobustCommand{\Ukrainian}{\selectlanguage{\ukrainian}}
\DeclareRobustCommand{\cyrillictext}{%
    \fontencoding\cyrillicencoding\selectfont
    \let\encodingdefault\cyrillicencoding
    \expandafter\set@hyphenmins\ukrainianhyphenmins
    \language\l@ukrainian}%
\let\cyr\cyrillictext
```

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

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

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

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

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

`\textcyrillic` `{`⟨*text*⟩`}`

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

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

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

6.3 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 Ukrainian version of Windows ANSI encoding which is almost equivalent to `cp1251`.

6.4 Shorthands

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

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

Initial activation state will set to on later in section 6.5.4.

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

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

6.4.1 Quotes

We set ’’‘ and ’’’ as shorthands for `\quotedblbase` and `\textquotedblleft`, respectively. 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.

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

```
\declare@shorthand{ukrainian}{’’‘}{\quotedblbase}
\declare@shorthand{ukrainian}{’’’}{\textquotedblleft}
\declare@shorthand{ukrainian}{’’<}{\guillemotleft}
\declare@shorthand{ukrainian}{’’>}{\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.

Because of pdfstrings patches for ukrainian shorthands were removed from `hyperref`, the support for them was added to the `ukrainian.ldf` file.

```
\providecommand\texorpdfstring[2]{#1}
\declare@shorthand{ukrainian}{’’”}{\hskip\z@skip}
\declare@shorthand{ukrainian}{’’~}{%
      \texorpdfstring{\textormath{\leavevmode\hbox{-}}{-}}{-}}
\declare@shorthand{ukrainian}{’’=}{\nobreak-\hskip\z@skip}
\declare@shorthand{ukrainian}{’’|}{%
    \texorpdfstring{\textormath{\nobreak\discretionary{-}{}{\kern.03em}\allowhyphens}{}}{}}
```

6.4.2 Emdash, endash and hyphenation sign

To distinguish between ’’- and ’’--- we must check whether the next after - token is a hyphen character. If it is, we output an emdash, otherwise a hyphen sign. Therefor TEX looks for the next token after the first ‘-’, writes its meaning to `\ukrainian@sh@next` and finally call for `\ukrainian@sh@tmp`.

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

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

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

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

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

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

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

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

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

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

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

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

\cyrdash The \cyrdash 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 **??** might change its length after \selectlanguage.

The \cyrdash macro is not available in Latin encodings such as T1. Therefor 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. Therefor we use the \ProvideTextCommandDefault method:

```
\PackageInfo{babel}{Default for \string\cyrdash\space is provided}
%%\ProvideTextCommandDefault{\cyrdash}{\iflanguage{ukrainian}%
%%  {\hbox to.8em{--\hss--}}{\textemdash}}
\ProvideTextCommandDefault{\cyrdash}{\hbox to.8em{--\hss--}}
```

The \cyrdash macro is not defined in the Unicode encoding TU. The fake definition given above cope with this case.

Finally, we define a shorthand thin space to be placed between initials as in D.'',Mendeleev. When used instead of \, as in D.\,Mendeleev it allows hyphenation in the next word.

```
\declare@shorthand{ukrainian}{'',}{\nobreak\hskip.2em\ignorespaces}
```

6.5 Switching to and from Ukrainian

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

\captionsukrainian The macro \captionsukrainian 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 Ukrainian captions are called only if current language is Ukrainian. However, the macros such as \Ukrainian do not conform to strict rules of the package babel as explained in the above.

We now use babel's 3.9 \Set<macro> macro for defining caption names as well as date. If Unicode engine is running, Cyrillic letters are typed in by their Unicode code-points.

```
%
% ------------------------- Caption Names (Unucode case) ------------------------------
%
\if@ukr@uni@ode
    \PackageInfo{ukrainian.ldf}{Executing the 3.9 or latter}
    \StartBabelCommands*{ukrainian}{captions}[unicode, fontenc=EU1 EU2, charset=utf8]
        \SetString\prefacename{Вступ}%                [babel]
        \SetString\refname{Література}%               [only article]
```

```
        \SetString\abstractname{Анотація}%            [only article, report]
        \SetString\bibname{Бібліографія}%             [only book, report]
        \SetString\chaptername{Розділ}%               [only book, report]
        \SetString\appendixname{Додаток}%
        \SetString\contentsname{Зміст}%
        \SetString\tocname{\contentsname}%
        \SetString\listfigurename{Перелік ілюстрацій}%
        \SetString\listtablename{Перелік таблиць}%
        \SetString\indexname{Предметний покажчик}%
        \SetString\authorname{Іменний покажчик}%
        \SetString\figurename{Рис.}%
        \SetString\tablename{Таблиця}%
        \SetString\partname{Частина}%
        \SetString\enclname{вкл.}%
        \SetString\ccname{вих.}%
        \SetString\headtoname{вх.}%
        \SetString\pagename{с.}%                      [letter]
        \SetString\seename{див.}%
        \SetString\alsoname{див.\ також}%
        \SetString\proofname{Доведення}%              [amsthm]
        \SetString\glossaryname{Словник термінів}%
        \SetString\acronymname{Абревіатури}%          [glossaries] {Acronyms}
        \SetString\lstlistingname{Лістинг}%           [listings]  (the environment) {Listing}
        \SetString\lstlistlistingname{Лістинги}%      [listings]  (the ''List of'') {Listings}
        \SetString\nomname{Позначення}%
        \SetString\notesname{Нотатки}%                [endnotes] {Notes}
```

Additional definitions for the package `nomencl`:

```
%
% ------------------------------ nomencl (Unucode case) --------------------------------
%
    \ifdefined\nomname%
        \addto\captionsukrainian{%
            \def\eqdeclaration#1{, див.\nobreakspace(#1)}%
            \def\pagedeclaration#1{, стор.\nobreakspace#1}%
        }%
    \fi
    \EndBabelCommands
\else
%
% ------------------------- Caption Names (Nonunucode case) -----------------------------
%
    \StartBabelCommands*{ukrainian}{captions}
        \SetString\prefacename{{\cyr\CYRV\cyrs\cyrt\cyru\cyrp}}%
        \SetString\refname{%
          {\cyr\CYRL\cyrii\cyrt\cyre\cyrr\cyra\cyrt\cyru\cyrr\cyra}}%
        \SetString\abstractname{%
          {\cyr\CYRA\cyrn\cyro\cyrt\cyra\cyrc\cyrii\cyrya}}%
        \SetString\bibname{%
          {\cyr\CYRB\cyrii\cyrb\cyrl\cyrii\cyro\cyrgup\cyrr\cyra\cyrf\cyrii\cyrya}}%
        \SetString\chaptername{{\cyr\CYRR\cyro\cyrz\cyrd\cyrii\cyrl}}%
        \SetString\appendixname{{\cyr\CYRD\cyro\cyrd\cyra\cyrt\cyro\cyrk}}%
        \SetString\contentsname{{\cyr\CYRZ\cyrm\cyrii\cyrs\cyrt}}%
        \SetString\tocname{\contentsname}%
        \SetString\listfigurename{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
             \ \cyrii\cyrl\cyryu\cyrs\cyrt\cyrr\cyra\cyrc\cyrii\cyrishrt}}%
        \SetString\listtablename{{\cyr\CYRP\cyre\cyrr\cyre\cyrl\cyrii\cyrk
             \ \cyrt\cyra\cyrb\cyrl\cyri\cyrc\cyrsftsn}}%
        \SetString\indexname{{\cyr\CYRP\cyrr\cyre\cyrd\cyrm\cyre\cyrt\cyrn\cyri\cyrishrt
                \ \cyrp\cyro\cyrk\cyra\cyrzh\cyrch\cyri\cyrk}}%
        \SetString\authorname{{\cyr\CYRII\cyrm\cyre\cyrn\cyrn\cyri\cyrishrt
             \ \cyrp\cyro\cyrk\cyra\cyrzh\cyrch\cyri\cyrk}}%
        \SetString\figurename{{\cyr\CYRR\cyri\cyrs.}}%
        \SetString\tablename{{\cyr\CYRT\cyra\cyrb\cyrl.}}%
        \SetString\partname{{\cyr\CYRCH\cyra\cyrs\cyrt\cyri\cyrn\cyra}}%
        \SetString\enclname{{\cyr\cyrv\cyrk\cyrl\cyra\cyrd\cyrk\cyra}}%
        \SetString\ccname{{\cyr\cyrk\cyro\cyrp\cyrii\cyrya}}%
```

```
        \SetString\headtoname{{\cyr\cyrv\cyrh.}}%
        \SetString\pagename{{\cyr\cyrs.}}%
        \SetString\seename{{\cyr\cyrd\cyri\cyrv.}}%
        \SetString\alsoname{{\cyr\cyrd\cyri\cyrv.\ \cyrt\cyra\cyrk\cyro\cyrzh}}%
        \SetString\proofname{{\cyr\CYRD\cyro\cyrv\cyre\cyrd\cyre\cyrn\cyrn\cyrya}}%
        \SetString\glossaryname{{\cyr\CYRS\cyrl\cyro\cyrv\cyrn\cyri\cyrk
             \ \cyrt\cyre\cyrr\cyrm\cyrii\cyrn\cyrii\cyrv}}%
        \SetString\acronymname{\cyr\CYRA\cyrb\cyrr\cyre\cyrv\cyrii\cyra\cyrt\cyru\cyri\cyri}%
        \SetString\lstlistingname{\cyr\CYRL\cyrii\cyrs\cyrt\cyri\cyrn\cyrg}%
        \SetString\lstlistlistingname{\cyr\CYRL\cyrii\cyrs\cyrt\cyri\cyrn\cyrg\cyri}%
        \SetString\nomname{\CYRP\cyro\cyrz\cyrn\cyra\cyrch\cyre\cyrn\cyrn\cyrya}%
        \SetString\notesname{\CYRN\cyro\cyrt\cyra\cyrt\cyrk\cyri}%
    \EndBabelCommands
\fi
```

6.5.2 Date in Ukrainian

\dateukrainian The macro \dateukrainian is used to reset the macro \today in Ukrainian.

```
%
% ---------------------------- Date (Unicode case) -----------------------------------
%
\if@ukr@uni@ode
    \PackageInfo{ukrainian.ldf}{Executing the post 3.9 branch for dates}
    \StartBabelCommands*{ukrainian}{date}[unicode, fontenc=EU1 EU2, charset=utf8]
        \SetStringLoop{month#1name}{%
            січня,%
            лютого,%
            березня,%
            квітня,%
            травня,%
            червня,%
            липня,%
            серпня,%
            вересня,%
            жовтня,%
            листопада,%
            грудня%
            }
        \SetString\abbgyear{р.}%
\else
%
% ---------------------------- Date (Nonunicode case) ---------------------------------
%
    \StartBabelCommands*{ukrainian}{date}
        \SetStringLoop{month#1name}{%
           \cyrs\cyrii\cyrch\cyrn\cyrya,%
           \cyrl\cyryu\cyrt\cyro\cyrg\cyro,%
           \cyrb\cyre\cyrr\cyre\cyrz\cyrn\cyrya,%
           \cyrk\cyrv\cyrii\cyrt\cyrn\cyrya,%
           \cyrt\cyrr\cyra\cyrv\cyrn\cyrya,%
           \cyrch\cyre\cyrr\cyrv\cyrn\cyrya,%
           \cyrl\cyri\cyrp\cyrn\cyrya,%
           \cyrs\cyre\cyrr\cyrp\cyrn\cyrya,%
           \cyrv\cyre\cyrr\cyre\cyrs\cyrn\cyrya,%
           \cyrzh\cyro\cyrv\cyrt\cyrn\cyrya,%
           \cyrl\cyri\cyrs\cyrt\cyro\cyrp\cyra\cyrd\cyra,%
           \cyrg\cyrr\cyru\cyrd\cyrn\cyrya%
           }%
        \SetString\abbgyear{\cyrr.}%
\fi
```

Typesetting date in both unicode and nonunicode cases

```
%
% ---------------------------- Date typesetting ---------------------------------------
%
```

```
\SetString\today{\number\day~\csname month\romannumeral\month name\endcsname\space
    \number\year~\abbgyear}%
\EndBabelCommands
```

6.5.3 Hyphenation patterns

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

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

6.5.4 Extra definitions

\extrasukrainian
\noextrasukrainian

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

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

```
\addto\extrasukrainian{\cyrillictext}
\addto\noextrasukrainian{\latintext}
```

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

```
\addto\extrasukrainian{\languageshorthands{ukrainian}}
\addto\extrasukrainian{\bbl@activate{’’}}
\addto\noextrasukrainian{\bbl@deactivate{’’}}
```

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

```
\addto\extrasukrainian{\bbl@frenchspacing}
\addto\noextrasukrainian{\bbl@nonfrenchspacing}
```

6.6 Alphabetic enumerations

The traditional alphabetical enumerations in Ukrainan texts use the Cyrillic alphabet (bar several letters). In principle, enumerations are a matter for class and style designers but the same can be said also about things, other than enumerations, such as names of sections and bibliography lists.

The alphabet is not the only difference, differences also in the labels format. According to Cyrillic typesetting tradition and also with ДСТУ 3008:2015, label format should be with one right parenthesis and the top level enumerate should be alphabetical, but we believe that this is not necessary for including such changes in `ukraineb.ldf`, for this purpose you can simply redefine required counters in preamle by common LaTeXway.

```
\def\theenumi{\alph{enumi}}
\def\labelenumi{\theenumi)}
\def\theenumii{\alph{enumii}}
\def\labelenumii{\theenumii)}
```

Nevertheless, the Ukrainian babel by default turns on alphabetical enumeration with Cyrillic letters. This means that enumerated lists that would be labelled with Latin letters in Latin scripts are labelled with Cyrillic ones instead.

\Alph

Starting from this virsion, we remove an macro \Asbuk (also it lowercase counterpart \asbuk). Now we redefine the macro \Alph, which is now produces (uppercase) Cyrillic letters instead of Latin ones when Ukrainian is switched on.

The letters Ґ, Є, З, І, Ї, Й, О, Ч, Ь, are skipped for such enumeration (see ДСТУ 3008:2015).

```
\addto\extrasukrainian{%
```

When Ukrainian swithsed off, the previous meaning of \@Alph will be restored

```
    \babel@save{\@Alph}%
     \if@ukr@uni@ode%
```

```
        \def\@Alph#1{%
            \ifcase#1\or%
                А\or Б\or В\or Г\or Д\or Е\or Ж\or%
                И\or К\or Л\or М\or Н\or П\or Р\or%
                С\or Т\or У\or Ф\or Х\or Ц\or Ш\or%
                Щ\or Ю\or Я%
            \else%
                \@ctrerr%
            \fi}%
    \else
        \def\@Alph#1{%
             \ifcase#1\or%
                \CYRA\or\CYRB\or\CYRV\or\CYRG\or\CYRD\or\CYRE\or\CYRZH\or%
                \CYRI\or\CYRK\or\CYRL\or\CYRM\or\CYRN\or\CYRP\or\CYRR\or%
                \CYRS\or\CYRT\or\CYRU\or\CYRF\or\CYRH\or\CYRC\or\CYRSH\or%
                \CYRSHCH\or\CYRYU\or\CYRYA%
             \else%
                 \@ctrerr%
             \fi}%
    \fi
}
```

\alph Now the macro \alph produces lowercase Cyrillic letters.

The lowercase letters ґ, є, з, і, ї, й, о, ч, ь, are also skipped such enumeration (see ДСТУ 3008:2015).

```
\addto\extrasukrainian{%
```

When Ukrainian swithsed off, the previous meaning of \@alph will be restored

```
    \babel@save{\@alph}%
    \if@ukr@uni@ode%
        \def\@alph#1{%
            \ifcase#1\or%
                а\or б\or в\or г\or д\or е\or ж\or%
                и\or к\or л\or м\or н\or п\or р\or%
                с\or т\or у\or ф\or х\or ц\or ш\or%
                щ\or ю\or я%
            \else%
                \@ctrerr%
            \fi}%
    \else
        \def\@alph#1{%
             \ifcase#1\or%
                \cyra\or\cyrb\or\cyrv\or\cyrg\or\cyrd\or\cyre\or\cyrzh\or%
                \cyri\or\cyrk\or\cyrl\or\cyrm\or\cyrn\or\cyrp\or\cyrr\or%
                \cyrs\or\cyrt\or\cyru\or\cyrf\or\cyrh\or\cyrc\or\cyrsh\or%
                \cyrshch\or\cyryu\or\cyrya%
            \else%
                \@ctrerr%
            \fi}%
    \fi
}
```

6.7 Ukrainian mathetematical typography traditions

\sh \ch \tg \ctg \arctg \arcctg \th \cth \cosec We also define few math operator names according to Ukrainian mathetematical typography traditions. Some math functions in Ukrainian math books have names different from English writings. For example, sinh in Ukrainian is called sh. Special consideration needs the macro \th that conflicts with the text symbol \th defined in Latin 1 encoding:

```
\AtBeginDocument{%
  \@ifpackagewith{babel}{russian}{\relax}{%
      \@ifpackageloaded{amsopn}
      {%
        \DeclareMathOperator{\sh}{sh}%
        \DeclareMathOperator{\ch}{ch}%
```

```
        \DeclareMathOperator{\tg}{tg}%
        \DeclareMathOperator{\ctg}{ctg}%
        \DeclareMathOperator{\arctg}{arctg}%
        \DeclareMathOperator{\arcctg}{arcctg}%
        \DeclareMathOperator{\cth}{cth}%\
        \DeclareMathOperator{\cosec}{cosec}%
        \DeclareMathOperator{\math@th}{th}%
      }{%
        \DeclareRobustCommand\sh{\mathop{\operator@font sh}\nolimits}%
        \DeclareRobustCommand\ch{\mathop{\operator@font ch}\nolimits}%
        \DeclareRobustCommand\tg{\mathop{\operator@font tg}\nolimits}%
        \DeclareRobustCommand\ctg{\mathop{\operator@font ctg}\nolimits}%
        \DeclareRobustCommand{\arctg}{\mathop{\operator@font arctg}\nolimits}%
        \DeclareRobustCommand\arcctg{\mathop{\operator@font arcctg}\nolimits}%
        \DeclareRobustCommand\cth{\mathop{\operator@font cth}\nolimits}%\MakeRobust\cth%
        \DeclareRobustCommand\cosec{\mathop{\operator@font cosec}\nolimits}%
        \DeclareRobustCommand{\math@th}{\mathop{\operator@font arctg}\nolimits}%
      }%
      \let\text@th\th%
      \DeclareRobustCommand{\th}{\TextOrMath{\text@th}{\math@th}}%
    }%
}
```

For compatibility with older Ukrainian packages we leave definition of the `\No` macro. However the Ukrainian number sign is now superseded with `\textnumero`. Moreover, it can be found on the keyboard.

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

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 `@`-sign, loading a local configuration file, and preparing the language to be activated at `\begin{document}` time.

```
\ldf@finish{ukrainian}
```