churchslavonic package — Church Slavonic Typography in LaTeX

Aleksandr Andreev and Mike Kroutikov
version 0.1

May 3, 2016

Abstract

Package churchslavonic provides fonts, hyphenation patterns and supporting macros to typeset Church Slavonic texts.

Contents

Introduction

Church Slavonic (also called Church Slavic, Old Church Slavonic or Old Slavonic; ISO 639-2 code `cu`) is a literary language used by the Slavic peoples; presently it is used as a liturgical language by the Russian Orthodox Church, other local Orthodox Churches, as well as various Byzantine-Rite Catholic and Old Ritualist communities. The package `churchslavonic` provides fonts, hyphenation patterns and supporting macros to typeset Church Slavonic texts in TeX.

The package is designed to support Unicode text encoded in UTF-8. Texts encoded in legacy codepages (such as HIP and UCS) may be converted to Unicode using a separate bundle of utilities. See the Slavonic Computing Initiative website for more information. To use the tools in this package, you will need a Unicode-aware TeX engine such as XTeX or LuaTeX.

1 How to use the package

To use churchslavonic package one needs to include the following declarations into the document preamble:

```
\usepackage{polyglossia}
\setmainlanguage{churchslavonic}
\usepackage{churchslavonic}
```

After that, use standard polyglossia commands to control current language.

1.1 Options `color`, `gray`, and `bw`

These options control what color actually being used for text coloring commands like `\cuKinovar`.

- `color` this is the default option and it preserves the original color (a shade of red).

- `gray` replaces red color with a gray - useful if you are printing on a media that does not support colors, but is capable of reproduction shades of gray.

- **bw** replaces red color with black (effectively turning off commands like `\cuKinovar` and `\cuKinovarColor`). Use this option to generate document that will be printed in blank-and-white.

Example:

```
\usepackage[gray]{churchslavonic}
```

2 Numbers

Church Slavonic numbering system is heavily based on the Old Greek one and uses letters as digits. For more information on the matter, see the appropriate section in [1].

2.1 \cuNum

Use this command to typeset a Church Slavonic number. The command takes a single argument that should expand to a number (register name works too).

\cuNum{1}	а҃
\cuNum{12}	в҃і
\cuNum{123}	рк҃г
\cuNum{1234}	҂асл҄д
\cuNum{10345}	҂ітм҃є
\cuNum{12345}	҂в҃і тм҃є
\cuNum{123456}	҂рк҃г ун҃ѕ
\cuNum{800456}	҂ѽун҃ѕ
\cuNum{1234567}	҂҂а҃ ҂сл҄д фѯ҃з
\cuNum{1500567}	҂҂а҃ ҂ффѯ҃з
\cuNum{12345678}	҂҂в҃і ҂тм҃є хѻ҃н
\cuNum{123456789}	҂҂рк҃г ҂ун҃ѕ ψн҃ѳ҄

3 Dates

\cuDate{2016-4-21}	к҃а апрі́ллїа лѣ́та ҂ѕ҃҃кі
\cuJulianDate{2016-4-21}	и҃ апрі́ллїа лѣ́та ҂ѕ҃҃кі
\cuDate{\cuToday}	г҃ ма́їа лѣ́та ҂ѕ҃҃кі

3.1 \cuDate

Command formats the date (according to the current format). Argument is a triplet of numbers YYYY-MM-DD specifying the date. Output will be something like this: к҃в а҆прі́лла лѣ́та ҂в҃і.

Note that YYYY-MM-DD values are not being normalized or interpreted in any way. Thus, it is totally fine to call \cuDate{2016-4-32} even though April 32 is not a valid date. It will be formatted and printed as April 32. This makes it possible to use this macro in a phrase like "date \cuDate{2016-4-32} is not a valid date in any calendar".

However, if your date format uses \cuDOW (day of the week) or \cuYEARAM (year Anno Mundi), the later quantities are computed by interpreting date as a Gregorian calendar date. Also, if input date is not a valid date, it will be normalized via extrapolation. For example, April 32nd will be interpreted as May 2nd for the purpose of determining values of the day of the week and year Anno Mundi.

If your format uses \cuDOW or \cuYEARAM, and your date is Julian, not Gregorian, you must use \cuJulianDate to correctly format days of the week and year Anno Mundi.

The best practice is to always use \cuDate with Gregorian dates and use \cuJulianDate with Julian dates regardless of the current date format. This way you can later on switch date formatting style without worrying about getting wrong output.

3.2 \cuDefineDateFormat

Command allows one to define date format. It does not change how \cuDate formats its output (for that, use \cuUseDateFormat). Example:

```
\cuDefineDateFormat{long}{%
  \cuDayName{\cuDOW},
  \cuNum{\cuDAY}_гѡ~%
  \cuMonthName{\cuMONTH}~%
  лѣ́та ѿ сотворе́нїѧ мі́ра~%
  \cuNum{\cuYEARAM}%
}
```

defines a format with name long. If we use this format to print the same date as before, we will get: пѧто́къ, к҃в_гѡ а҆прі́лла лѣ́та ѿ сотворе́нїѧ мі́ра ҂зфк҃є.

Following symbolic names can be used when formatting the date:

- `\cuYEAR` — year part of date (number, like 2016)

- `\cuYEARAM`[1] — year since world creation, aka Anno Mundi (number, like 7525)

- `\cuMONTH` — month part of date (number from 1 to 12)

- `\cuDAY` — day of the month

- `\cuDOW`[1] — day of the week (number from 0 to 6, where 0 means "Sunday")

- `\cuINDICTION` — indiction year (number from 1 to 15)

3.3 `\cuUseDateFormat`

This command sets the date format to be used by the subsequent `\cuDate` and `\cuJulianDate`.

3.4 `\cuJulianDate`

Formats date, just like `\cuDate` does, but argument is interpreted as a Julian calendar date, not a Gregorian calendar one. This makes difference only if your format is using symbolic names `\cuDOW` and/or `\cuYEARAM`.

3.5 `\cuMonthName`

Command expands numeric argument (month number) into textual representation. It is typically used when defining date format. For example, date format named `default` is defined as:

```
\cuDefineDateFormat{default}{%
  \cuNum{\cuDAY}~\cuMonthName{\cuMONTH}%
  ~ лѣта~\cuNum{\cuYEAR}%
}%
```

3.6 `\cuDayName`

Expands numeric argument into a textual representation of the day of the week using nominative case.

---

[1]If your format uses this value, make sure that you format date with a correct macro: you must use `\cuDate` if passed date is Gregorian, or `\cuJulianDate` if you pass Julian date.

3.7 \cuDayNameAccusative

Expands numeric argument into a textual representation of the day of the week using accusative case.

3.8 \cuToday

This macro expands to a triplet `YYYY-MM-DD`. Date is generated using Gregorian calendar.

3.9 \cuJulianToday

This macro expands to a triplet `YYYY-MM-DD`. Date is generated using Julian calendar.

It is a shortcut for `\cuAsJulian{\cuToday}`.

3.10 \cuAsJulian

Converts Gregorian date to Julian date. Input and output use numeric triplet format `YYYY-MM-DD`.

Useful when the same date needs to be formatted as Gregorian and as Julian date.

3.11 \cuAsGegorian

Converts Julian date to Gregorian date. Input and output use numeric triplet format `YYYY-MM-DD`.

4 Kinovar

Printed and hand-written Church Slavonic texts often use color to highlight sectional and paragraph structure. Typical style would use red color to write section names, comments, marginal notes. First letter of each paragraph is also colored red.

4.1 \cuKinovar

Takes a single argument and prints it using red color. For example, explicitly specifying its argument one gets expected result:

| \cuKinovar{лйкх:} гдⷭ҇н помн́лꙋй. | лйкх: гдⷭ҇н помн́лꙋй. |

If one uses TeX mechanism of implicit argument detection, then, red color will be used for the first character of the text after this command. Non-trivial feature of this command is that it will also "collect" all diacritical marks that belong to this first character, and thus all accents will also use the red color! Use this command in the indirect parameter mode to paint red the first letter of each paragraph.[1]

| \cuKinovar Поймꙋ гꙋ҃ви пѣснь новꙋю | Поймꙋ гꙋ҃ви пѣснь новꙋю |
| \cuKinovar Ꙗкѡ тꙋ́ча на тро́скотъ | Ꙗкѡ тꙋ́ча на тро́скотъ |

4.2 \cuKinovarColor

Switches current color to red. One would typically use this command inside a group that limits the scope of red text, unless you want all subsequent text to be colored red.

5 Utilities

The mechanism that \cuKinovar macro is using to collect all accents (when argument is given implicitly) can be useful for many other purposes. One example is to typeset a dropletter at the beginning of a chapter (this is often used in Church Slavonic texts). For this purpose standard LaTeX package lettrine works fine. The only nuisance is that one have to be careful to pass to \lettrine not just the first letter, but also all diacritical marks for this letter. Naturally, we want to reuse the clever mechanism that \cuKinovar is using to do the collection job.

Here is an example how to accomplish that:

```
\def\cu@lettrine{\lettrine[lines=3,findent=0pt,nindent=0pt]}
\def\cuLettrine{\cu@tokenizeletter\cu@lettrine}
\renewcommand{\LettrineFontHook}{\cuKinovarColor}
```

Once this definition of \cuLettrine is created (put is somewhere in preamble, and do not forget to enclose in \makeatletter and \makeatother), you can create drop capitals like this:

---

[1] You can also experiment with TeX command \everypar to autoamte this, but the success or failure of this technique critically depends on the LaTeX class used and packages loaded. We found it very fragile and thus churchslavonic package does not offer any automation for this. It may be easier and more robust use Search/Replace functionality embedded in any non-trivial text editor to just automatically place \cuKinovar command before every paragraph of the source text.

\cuLettrine Иҗе дꙋха сила въ немощи совершается...

Иже дꙋха сила въ немощи совершается, ꙗкоже писано єсть, и вѣрꙋемъ: въ немощи же не тѣлесє точїю, но оубо и слова, и премꙋдрости на ѧзыцѣ лежаща. И се ꙗвѣ ѿ многихъ оубо иныхъ, паче же ѿ иже ѡ великомъ бгословѣ, и братѣ хртовѣ, благодатїю зримѣмъ.

References

[1] Aleksandr Andreev, Yuri Shardt, and Nikita Simmons. *Church Slavonic Typography in Unicode*, Uncode Technical Note 41, `http://www.unicode.org/notes/tn41/`