

churchslavonic package — Church Slavonic Typography in L^AT_EX

Aleksandr Andreev and Mike Kroutikov
version 0.2

December 18, 2016

Abstract

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

Contents

1	How to use the package	1
1.1	Options <code>color</code> , <code>gray</code> (= <code>grey</code>), and <code>bw</code>	2
1.2	Underscore	2
2	Numbers	3
2.1	<code>\cuNum</code>	3
3	Dates	4
3.1	<code>\cuDate</code>	4
3.2	<code>\cuDateJulian</code>	4
3.3	<code>\cuDefineDateFormat</code>	4
3.4	<code>\cuUseDateFormat</code>	5
3.5	<code>\cuMonthName</code>	6
3.6	<code>\cuDayName</code>	6
3.7	<code>\cuDayNameAccusative</code>	6
3.8	<code>\cuToday</code>	6
3.9	<code>\cuTodayJulian</code>	6
3.10	<code>\cuAsJulian</code>	6
3.11	<code>\cuAsGegorian</code>	7

4	Kinovar	7
4.1	<code>\cuKinovar</code>	7
4.2	<code>\cuKinovarColor</code>	7
5	Miscellaneous	8
5.1	<code>\cuMarginMark</code> , <code>\cuMarginMarkSkip</code> , & <code>\cuMarginMarkText</code>	8
5.2	Dropcaps	9

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 \XeTeX or $\text{Lua}\TeX$.

1 How to use the package

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

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

This makes the Church Slavonic hyphenation patterns immediately available. After that, use the standard `polyglossia` commands to control current language. Church Slavonic fonts are provided by the `fonts-churchslavonic` package, which should have been installed automatically when you installed this package. See the `fonts-churchslavonic` documentation for information about fonts.

1.1 Options `color`, `gray` (= `grey`), and `bw`

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

- `color` this is the default option; it indicates the original color (a shade of red).
- `gray` (`grey`) replaces the red color with gray - useful if you are printing on media that does not support color, but is capable of grayscale.
- `bw` replaces the red color with black (effectively turning off commands like `\cuKinovar` and `\cuKinovarColor`). Use this option to generate a document that will be printed in black-and-white.

Example:

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

1.2 Underscore

The underscore symbol (`_`, U+005F LOW LINE) is a valid text symbol in Church Slavonic (it has roughly the same role as the hyphen in English). The package `churchslavonic` redefines the underscore in a way that it can be directly entered in text mode, for example:

ПОСЛѢДОВАНИЕ МОЛѢБНАГО ПѢНІА С҃БІМЪ МѢКАМЪ К҃ГѦ ВѢКА,
ВЪ С҃НКТЪ_ПЕТРѢВЪРЖИТѢЙ ДѢХОВНѢЙ АКАДЕМІИ
НАЧАЛЬСТВОКАШИМЪ, ОУЧѢШИМЪ И ОУЧѢШИМЦА

Attention: if you have an older version of the `fontspec` package installed on your system, the redefined underscore symbol cannot be used in font names and font options in `fontspec` commands like `\setXXXfont` and `\newfontfamily`.

Typically you need to set underscore as the hyphenation character for Church Slavonic fonts: `HyphenChar=``_`. With older versions of `fontspec` this will cause errors. This problem exists in `TEX Live 2013` and in `fontspec v2.3c`.

We recommend upgrading your `TEX` distribution to at least `TEX Live 2015`. Alternatively you can selectively upgrade the `fontspec` package to version `v2.4c` or better.

If upgrading is not an option, you can work around this problem by either specifying the hexadecimal code for the hyphenation character: `HyphenChar="005F`, or by declaring all fonts before loading `churchslavonic`.

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

If your format uses `\cuDOW` or `\cuYEARAM`, and you specify a date according to the Julian, not Gregorian, calendar, you must use `\cuDateJulian` to correctly format days of the week and year Anno Mundi.

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

3.2 `\cuDateJulian`

Formats the date, just like `\cuDate` does, but the argument is interpreted as a date on the Julian calendar (“old style”) instead of the Gregorian calendar. This makes a difference only if your format is using symbolic names `\cuDOW` and/or `\cuYEARAM`.

3.3 `\cuDefineDateFormat`

This command allows you to define your own date format. It does not change how `\cuDate` formats its output (for that, use `\cuUseDateFormat`). Example:

```
\cuDefineDateFormat{long}{%
  \cuDayName{\cuDOW},
  \cuNum{\cuDAY}_ρω~%
  \cuMonthName{\cuMONTH},~%
  ΛΕΪΤΑ Ψ ΙΟΥΚΟΡΕΝΙΑ ΜΙΡΑ~%
  \cuNum{\cuYEARAM}%
}
```

defines a format with name `long`. If we use this format to print the same date as before, we will get: ΠΑΤΡΟΚΣ, ΚΒ_ΓΩ ΑΠΡΙΛΛΙΑ, ΛΕΪΤΑ Ψ ΙΟΥΚΟΡΕΝΙΑ ΜΙΡΑ ,ΩΦΚΣ.

The following symbolic names can be used when formatting the date:

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

- `\cuYEARAM`¹ — the year Anno Mundi, that is, since the creation of the world according to the Byzantine reckoning (aka “the Byzantine era”; a number, like 7525)
- `\cuMONTH` — the month part of a date (a number from 1 to 12, with January set to 1)
- `\cuDAY` — the day of the month
- `\cuDOW`¹ — the day of the week (number from 0 to 6, where 0 means “Sunday”)
- `\cuINDICTION` — the indiction² (a number from 1 to 15)

3.4 `\cuUseDateFormat`

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

3.5 `\cuMonthName`

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

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

3.6 `\cuDayName`

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

3.7 `\cuDayNameAccusative`

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

¹If your format uses this value, make sure that you format the date with the correct macro: you must use `\cuDate` for dates on the Gregorian calendar and `\cuDateJulian` for dates on the Julian calendar.

²See <https://en.wikipedia.org/wiki/Indiction>.

3.8 `\cuToday`

This macro expands to a triplet YYYY-MM-DD. The date is generated according to the Gregorian calendar.

3.9 `\cuTodayJulian`

This macro expands to a triplet YYYY-MM-DD. The date is generated according to the Julian calendar.

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

3.10 `\cuAsJulian`

Converts a date on the Gregorian calendar to a date on the Julian calendar. Input and output use numeric triplet format YYYY-MM-DD.

Useful when the same date needs to be formatted both according to the Gregorian and Julian calendars.

3.11 `\cuAsGegorian`

Converts a date according to the Julian calendar to a date according to the Gregorian calendar. 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 and to indicate liturgical rubrics, section names, comments, and marginal notes. The first letter of each paragraph is also often colored red.

4.1 `\cuKinovar`

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

<code>\cuKinovar{лѣкз:}</code> гѣи помѣлѣи.	лѣкз: гѣи помѣлѣи.
---	--------------------

If one uses the \TeX mechanism of implicit argument detection, then the first character of the text after this command will be printed in red. However, a non-trivial feature of this command is that it will also “collect” all of the

diacritical marks that belong to this first character, and thus all accents will also be colored in red! Use this command in the indirect parameter mode to print in red the first letter of each paragraph.¹

<code>\cuKinovar</code> Поі́мъ гѣвн пѣснь нѣвѣю	Поі́мъ гѣвн пѣснь нѣвѣю
<code>\cuKinovar</code> ѿ́кв тѣча на трѣкоцѣ	ѿ́кв тѣча на трѣкоцѣ

4.2 `\cuKinovarColor`

Switches the 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 Miscellaneous

5.1 `\cuMarginMark`, `\cuMarginMarkSkip`, & `\cuMarginMarkText`

The command `\cuMarginMark` is used to place short text in the margin at the same level as the line where the macro is placed. Example:

```

\cuMarginMark{б}\cuKinovar Пѣснь гѣвн моѡ гѣвн ѡ не забвѣѡ
вѣѣхъ воздѡвнѣѡ ѡгѡ. \cuMarginMark{а}\cuKinovar Ѡчищающаго
вѣѡ беззакѡнѣѡ твоѡ, ѡчищающаго вѣѡ неѡчищенѣѡ твоѡ:
\cuMarginMark{б}\cuKinovar Ѡчищающаго ѡчищающаго ѡчищающаго живѡтъ твоѡ,
вѣѣѡчищающаго тѣѡ млѣѣѡ ѡ чиѡрѡтамн: \cuMarginMark{а}\cuKinovar
Ѡчищающаго во вѣѣѣѡхъ желѡнѣѡ твоѡ, ѡчищающаго ѿѡвѡнѣѡ ѿѡвѡнѣѡ
ѡчищающаго твоѡ.

```

Will result in:

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

Ꙗ БѢГОСЛОВИ ДѢШЕ МОѦ ГДА Н НЕ ЗАБЫВАЙ КСѢХЪ ВОЗДААНІИ
 Я СГѦ. Ѡчищающаго вѣѦ беззаконїѦ твоѦ, исцѣляющаго
 Ꙗ вѣѦ недѣги твоѦ: НЗБАВЛЯЮЩАГО Ѡ исцѣлїнїѦ жнвотѣх
 Я ТВОИ, вѣнчающаго тѦ млтїю н цедротами: НПОЛНЯЮЩАГО
 ВО БЛГІХЪ ЖЕЛАНІЕ ТВОЕ, ѠБНОВІТЕЛ ЯКѦ Ѡрла ѡноутѣ
 ТВОѦ.

The marginal mark is placed on the right margin for odd pages and on the left margin for even pages (e.g. the mark is placed on the “outer” margin, not the spine margin), which is usually the desired behavior.

The distance between the mark and the text is controlled by the value of `\cuMarginMarkSkip`. The default is:

```
\def\cuMarginMarkSkip{0.6em}
```

To globally customize the font and color of the margin mark use `\cuMarginMarkText`. For example, to make margin marks appear in red color, redefine `\cuMarginMarkText` in the preamble of your document like this:

```
\def\cuMarginMarkText#1{\cuKinovar{#1}}
```

If you need to change the font/size/color just for a single mark, you can do it directly with `\cuMarginMark`:

```
\cuMarginMark{{\tiny *}}
```

5.2 Dropcaps

The mechanism that the `\cuKinovar` macro is using to collect all accents (when the argument is specified 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, the standard L^AT_EX package `lettrine` works just fine. The only nuisance is that one has to be careful to pass to `\lettrine` not just the first letter, but also any diacritical marks that attach to this letter. Naturally, we want to reuse the clever mechanism that `\cuKinovar` uses, and automatically collect the diacritical marks!

Here is an example of how to accomplish this:

```

\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 (somewhere in the preamble, between the declarations `\makeatletter` and `\makeatother`), you can create drop capitals like this:

```
\cuLettrine Њже дѣла сила въ нѣмоци совершáетсѧ...
```

Њже дѣла сила въ нѣмоци совершáетсѧ, ѣкоже пѣсано
ѣсть, ѡ вѣрѣмъ: въ нѣмоци же не чѣлесѧ члвчю,
но ѡбѣ и глагола, и премѣдрости на азѣцѣ лежáца.
Њ еѣ ѣвѣ ѡ многых ѡбѣ иныих, паче же ѡ ѡже ѡ
великомъ бѣословѣ, и братѣ хрѣтовѣ, благодáтїю зрѣмѣмъ.

References

- [1] Aleksandr Andreev, Yuri Shardt, and Nikita Simmons. *Church Slavonic Typography in Unicode*, Unicode Technical Note 41. 2015. <http://www.unicode.org/notes/tn41/>