The XCharter Font Package

Michael Sharpe

June 8, 2024

1 Package Features

The *XCharter* fonts are extensions of the Bitstream Charter fonts, adding oldstyle figures, superior figures and small caps in all styles. The original Charter fonts were created by famed font designer Matthew Carter in the late 1980's to enhance legibility of the output from printers of that era (laser, dot matrix, thermal and inkjet) with resolutions that would now be considered low—not far from modern screen resolutions. Their low contrasts, high x-heights and use of piecewise linear outlines where possible may make them interesting again as fonts that will render well on small devices and perhaps projected slides.

It's worth noting that the same designer provided Georgia for Microsoft. It is widely considered to be one of the clearest serifed fonts for viewing on screen, and bears many similarities to Charter, though with rounder finials, flatter serifs and curvier transitions. Their x-heights are the same for all practical purposes but Georgia has Cap-height and Ascender-height about 2% more than Charter.

(This paragraph is processed using Georgia reduced by 1%.) The vertical stem widths of Georgia are about 10% thicker than those of Charter and Georgia has distinctly greater contrast. As you can see, it appears to render blacker. and occupies about 3% less horizontal space than Charter.

(This paragraph is processed using Gelasio-Regular reduced by 1%. Gelasio is a free, open source font that is metrically equivalent to Georgia. Like Georgia, it has no small caps, but work is in progress to add these via virtual fonts based on XCharter, and to add math fonts based on `newtx`.)

As of version 1.09 (2017-06-25) there is a new collection of Cyrillic glyphs in *XCharter*, copied from Andrey Panov's *Khartiya*, an extension of the free Charter fonts, with small caps included. Some new figure styles were also copied from *Khartiya*—inferiors, numerators and denominators. Along with these additions, there are now slanted versions for those who wish to have both slanted and italic text available to meet distinct semantic purposes. Note that figures and uppercase slanted and italic are almost identical (except for slanted *Q* and italic *Q*) but lower-case forms are distinct.

Starting with version 1.1, support has been added for typesetting in the Serbian variant of Cyrillic, with some changes to the Italic and BoldItalic Cyrillic glyphs and a new option in the main package. These are described more fully below.

Support files are provided for T1, TS1, LY1, T2A and OT2 encodings, the last two being to support the Cyrillic component of *XCharter*. The package has a number of options:

- `scaled=.98`, for example, scales all text to 98% of specified size;

- `lining` (or just `lf`) makes lining figures (0123456789) the default for text—this is set automatically and does not need to be entered explicitly;
- `oldstyle` (or `osf`) sets the figure style in text mode to oldstyle (0123456789) with numeral one like a shortened 1, but math mode will always use lining figures;
- `proportional` (or `p`), new as of version 1.23, changes the default tabular figure style to proportional.
- `oldstyleI` (or `osfI`) sets the figure style in text mode to oldstyle (0123456789) with numeral one like a shortened I, but math mode will always use lining figures;
- `sups` sets the style for superscript figures (e.g., footnote markers) to XCharter's superior figures rather than using the default text inserts in mathematical superscripts. This option has no effect if a KOMA class is in force.
- `scosf` makes oldstyle figures the default in small cap text, no matter what the global figure setting may be.
- `serbianc` is useful only with the T2A encoding. It modifies one slot in upright and slanted shapes and five slots in italic shapes, as expected in Serbian Cyrillic. See the last section for examples.

Changes in version 1.23

There are some substantial additions in version 1.23, some requiring `newtx`, version 1.71 or higher:

- `XCharter.sty` now works with all flavors of LaTeX—unicode and non-unicode—but there may be some small differences in output. Essentially all previous options and macros are supported and there are new ones available, some of which are limited to unicode engines.
- Previous versions of `XCharter` had only two normal figure styles: `tabular lining` (the default) and `proportional oldstyle`. Version 1.23 adds two more so there are separate `TLF (tabular lining figures)`, `LF (proportional lining figures)`, `TOsF (tabular oldstyle figures)` and `OsF (proportional oldstyle figures)`. Two new options have been added to globally select the default figures style. Option `p` (or `proportional`) and `t` (or `tabular`). A new command `\useproportional` (preamble only) has the same effect as option `proportional`.
- With the new figures came new macros to select them, no matter what the defaults may be. There are two forms, one that switches the figures until further notice and the other a macro with an argument.

Switch	Command	Effect	Example
`\tlfstyle`	`\texttlf`	TLF	`{\tlfstyle123}, \texttlf{123}`
`\lfstyle`	`\textlf`	LF	`{\lfstyle123},\textlf{123}`
`\tosfstyle`	`\texttosf`	TOsF	`{\tosfstyle123},\texttosf{123}`
`\osfstyle`	`\textosf`	OsF	`{\osfstyle123}, \textosf{123}`

There are also the text font switches `\liningnums`, `\tabularnums`, `\oldstylenums` and `\proportionalnums`, each of which changes only one attribute of the figure style and alignment. For example, `\liningnums` changes the style to `lining` and `\tabularnums` changes

the figure alignment to `tabular`.

- There is a theorem font option similar to those in newtx and newpx. A new font family named `XCharterTH` is made from the italic and bold italic faces of `XCharter`, but having upright figures and punctuation that, IMO, look better than slanted ones in theorem statements and the like. For details, consult the brief descriptions below and the more discursive version in the documentation to `newtx`. There is a `theoremfont` option to `XCharter` that works exactly the same as in `newtx`.
- The figure style in `theoremfont` will by default be the same as your chosen figure style. Option `thmlining` will ensure that lining figures are always used.
- `oldSS` specifies the preference for the old Capital Sharp S rather than the newer form, `U+1E9E`, ß.
- There are new options that affect only unicode engines:
  - `type1text` (or `type1`) specifies processing the text font using `type1` mode. This does not prevent `fontspec` from loading.
  - `defaultfeatures=` gives you a place to set the default text font features for `fontspec`.

Special Macros:

- `\useproportional` (usable only in the preamble) may be used for changing the text figure alignment to `proportional` though math mode will use tabular lining figures. (New in 1.23.)
- `\useosf` (usable only in the preamble) may be used for changing the text figure style to `osf` though math mode will use lining figures.
- `\useosfI` (usable only in the preamble) may be used for changing the text figure style to `osfI` though math mode will use lining figures.
- `\textsu` prints its argument in superior figures, e.g., `\textsu{12}` results in ^{12}. The effect is the same with `{\sustyle 12}`.
- `\textinf` prints its argument in inferior figures, e.g., `\textinf{12}` results in $_{12}$. The effect is the same with `{\instyle 12}`. (In versions of `XCharter` prior to 1.221, `\textinf` was named `\textin`, but the latter conflicts with `hyperref` which redefines it to point to U+2208.)
- `\textlf` prints its argument in lining figures, e.g., `\textlf{12}` results in 12. The effect is the same with `{\lfstyle 12}`.
- `{\osfstyle 23}` prints 23 (`OldStyle,Proportional`) while `{\liningnums 23}` prints 23, `Lining` with whatever figure alignment is in force. There are also macros `\tabularnums`, `\proportionalnums`, `\oldstylenums`, `\tosfstyle` and `\tlfstyle` with the expected behaviors.
- Numerators and denominators are normally used only for constructing fractions, but may if needed be called using `\textnumerator` and `\textdenominator`. They are about 7% smaller than superiors and inferiors. You may use `\textde` and `\textnu` as abbreviations, though the latter will not be available if `babel` is loaded with `greek` option. As of version 1.24, you may prevent `\textnu` from overwriting the babel/greek definition by using the new option `notextnu` to `XCharter`. In any case, a new command `\textnum` takes the place of the old `XCharter` `\textnu`.

- The `\textfrac` macro allows you to write, e.g., `\textfrac{31}{32}` to get the simple fraction 31⁄32, and `\textfrac[2]{31}{32}` to get 2 31⁄32. (The optional argument, 2 in the latter case, is always typeset in lining figures.)
- The `\textsfrac` macro, available only when you use the `newtx` package with option `xcharter` to load `XCharter` with `newtxmath`, allows you to write, e.g., `\textsfrac{31}{32}` to get the simple stacked fraction $\frac{31}{32}$,and `\textsfrac[2]{31}{32}` to get $2\frac{31}{32}$. (The optional argument, 2 in the latter case, is always typeset in lining figures.)
- `\textcircled` renders its argument in raised and reduced small caps encircled by the `bigcircle` glyph. E.g., `\textcircled{M}` and `\textcircled{m}` both render as Ⓜ. The macro works also for numerals: `\textcircled{2}` renders as ②.
- `\textth` (and also `\textthit`) render their arguments using the theorem fonts. For example: `\textth{Theorem font (01234):!}` renders as *Theorem font* (01234):!—note the upright figures and punctuation. (There is no Bold theorem font—if you attempt it using the specification `\textbf{\textth{Theorem font (01234):!}}`, you will find that it renders as ***Theorem font (01234):!***.) The related font switch `\thfamily` is defined so that `{\thfamily A12!}` and `\textth{A12!}` are equivalent. In opentype processing, the `StylisticSet 05` controls whether figures and punctuation are upright in italic shaped faces.

Greek letters in version 1.11:

$$\Gamma\ \Delta\ \Theta\ \Lambda\ \Xi\ \Pi\ \Sigma\ \Upsilon\ \Phi\ \Psi\ \Omega$$

$$\mathit{\Gamma\ \Delta\ \Theta\ \Lambda\ \Xi\ \Pi\ \Sigma\ \Upsilon\ \Phi\ \Psi\ \Omega}$$

$$\alpha\ \beta\ \gamma\ \delta\ \epsilon\ \zeta\ \eta\ \theta\ \iota\ \kappa\ \lambda\ \mu\ \nu\ \xi\ o\ \pi\ \rho\ \sigma\ \tau\ \upsilon\ \phi\ \chi\ \psi\ \omega\ \varepsilon\ \vartheta\ \varpi\ \varrho\ \varsigma\ \varphi\ \varkappa\ \ell\ \wp$$

$$\mathrm{\alpha\ \beta\ \gamma\ \delta\ \epsilon\ \zeta\ \eta\ \theta\ \iota\ \kappa\ \lambda\ \mu\ \nu\ \xi\ o\ \pi\ \rho\ \sigma\ \tau\ \upsilon\ \phi\ \chi\ \psi\ \omega\ \varepsilon\ \vartheta\ \varpi\ \varrho\ \varsigma\ \varphi\ \varkappa}$$

$$\mathbf{\Gamma\ \Delta\ \Theta\ \Lambda\ \Xi\ \Pi\ \Sigma\ \Upsilon\ \Phi\ \Psi\ \Omega}$$

$$\boldsymbol{\Gamma\ \Delta\ \Theta\ \Lambda\ \Xi\ \Pi\ \Sigma\ \Upsilon\ \Phi\ \Psi\ \Omega}$$

$$\boldsymbol{\alpha\ \beta\ \gamma\ \delta\ \epsilon\ \zeta\ \eta\ \theta\ \iota\ \kappa\ \lambda\ \mu\ \nu\ \xi\ o\ \pi\ \rho\ \sigma\ \tau\ \upsilon\ \phi\ \chi\ \psi\ \omega\ \varepsilon\ \vartheta\ \varpi\ \varrho\ \varsigma\ \varphi\ \varkappa\ \ell\ \wp}$$

$$\mathbf{\alpha\ \beta\ \gamma\ \delta\ \epsilon\ \zeta\ \eta\ \theta\ \iota\ \kappa\ \lambda\ \mu\ \nu\ \xi\ o\ \pi\ \rho\ \sigma\ \tau\ \upsilon\ \phi\ \chi\ \psi\ \omega\ \varepsilon\ \vartheta\ \varpi\ \varrho\ \varsigma\ \varphi\ \varkappa}$$

2 Math/text package choices

All unicode

There is now a unicode math package, `XCharter-Math` that may be run with a simple preamble containing

```
\usepackage{fontspec}
\setmainfont{XCharter} % reads XCharter.fontspec
\usepackage{unicode-math}
\setmathfont{XCharter-Math.otf}
```

or, even better, as described in the documentation for `XCharter-Math`,

```
\usepackage{xcharter-otf}
```

but in order to get the options and macros described in this documentation, you should use instead, for the same effect

```
\usepackage[otfmath]{XCharter}
% loads fontspec, unicode-math, and sets XCharter-Math.otf
```

Notes on the last preamble fragment:

- Unless option `otfmath` is specified, math will be processed by `newtxmath` with `xcharter` option. (See examples 2–6 below.)
- All options passed to `XCharter` that are not understood by `XCharter` will be passed along to `xcharter-otf` provided option `otfmath` was specified.

Non-unicode only

Three non-unicode math packages seem to provide reasonable companions for XCharter. The first example uses Charter italics as math italics, but doesn't provide arbitrary scaling and doesn't sufficiently distinguish math italic v from mathematical Greek `\nu`. Moreover, it is not easy to redefine `\mathcal` to get a better math calligraphic alphabet—e.g., the `mathalpha` package fails. The second uses libertine italics and Greek in math mode, which is a good match to Charter in style and weight after scaling up, is arbitrarily scalable, has distinct math italic v and mathematical Greek `\nu`, and is completely compatible with `mathalpha`. The third is a new revision of `newtxmath` with option `charter` (or, equivalently, `xcharter`), which substitutes Charter italics as math italics and, as of version 1.11, uses a newly developed family of Greek symbols in {regular, bold} × {upright, italic} to match the style and italic angle of XCharter. This version is scalable and has a math italic v (plus a matching w) distinct from `\nu`. (The option `noxchvw` to `newtxmath` changes the v and w to be the original Charter italic glyphs, which may lead to issues with `\nu`.)

Example 1:

```
% [pdf]latex only
\usepackage[charter,expert]{mathdesign}
\usepackage[scaled=.96,osf]{XCharter}% matches the size used in math
\linespread{1.04}
```

Example 2:

```
% [pdf]latex only
\usepackage[scaled=.98,sups,osf]{XCharter}% lining figures in math, osf in text
\usepackage[scaled=1.04,varqu,varl]{inconsolata}% inconsolata typewriter
\usepackage[type1]{sourcesanspro}% sans serif
\usepackage[uprightscript,libertine,vvarbb,scaled=1.05]{newtxmath}
\linespread{1.04}
```

Non-unicode using `newtxmath`

Example 3:

```
% [pdf]latex only
\usepackage[scaled=.98,sups,osf]{XCharter}% lining figures in math, osf in text
\usepackage[scaled=1.04,varqu,varl]{inconsolata}% inconsolata typewriter
```

```
\usepackage[type1]{sourcesanspro}% sans serif
\usepackage[uprightscript,charter,vvarbb,scaled=1.05]{newtxmath}
\linespread{1.04}
```

Example 4:

```
% [pdf]latex only
\usepackage[<specify babel languages>]{babel}% load before XCharter
\usepackage[scaled=.98,sups,osf]{XCharter}% osf in text, lining figures in math
\usepackage[scaled=1.04,varqu,varl]{inconsolata}% inconsolata typewriter
\usepackage[type1]{sourcesanspro}% sans serif
\usepackage[uprightscript,charter,vvarbb,scaled=1.05]{newtxmath}
\linespread{1.04}
```

Example 5:

```
% an example using newtx.sty, works with all latex engines
\usepackage[<specify babel languages>]{babel}% load before newtx
\usepackage[scaled=1.04,varqu,varl]{inconsolata}% inconsolata tt
\usepackage[type1]{sourcesanspro}% sans serif for math
\usepackage[T1]{fontenc} % encoding to use for mathtt, etc
\usepackage[xcharter,osf,p,mathscale=1.05,textscale=0,uprightscript,vvarbb]{newtx}
% loads newtxmath
% newtx loads fontspec with unicode engines
\setmonofont{lmmono10-regular.otf}[Scale=1.08] % typewriter for text
\linespread{1.04}
% load polyglossia after newtx, if using
```

Example 6:

```
% Adds instructions to produce a pdf conforming to PDF/A-1b
%\pdfcompresslevel=0 % uncomment for debugging the pdf
%\pdfgentounicode=1
%\input glyphtounicode.tex  % now part of latex
\InputIfFileExists{glyphtounicode-cmr.tex}{}{}
\InputIfFileExists{glyphtounicode-ntx.tex}{}{}
\usepackage[a-1b]{pdfx} % version 1.6.4 or higher
\usepackage[<specify babel languages>]{babel}% load before XCharter
\usepackage[scaled=.98,sups,osf]{XCharter}% osf in text, lining figures in math
\usepackage[scaled=1.04,varqu,varl]{inconsolata}% inconsolata typewriter
\usepackage[type1]{sourcesanspro}% sans serif
\usepackage[uprightscript,charter,vvarbb,scaled=1.05]{newtxmath}
\linespread{1.04}
```

Here is a short sample based on the preamble of Example 3:

The typeset math below follows the ISO recommendations that only variables be set in italic. Note the use of upright shapes for d, e and π. (The first two are entered as `\mathrm{d}` and `\mathrm{e}`, and in fonts derived from `newtxmath` or `mtpro2`, the latter is entered as `\uppi`.)

Simplest form of the *Central Limit Theorem*: *Let* $X_1, X_2, \cdots$ *be a sequence of iid random variables*

with mean 0 *and variance* 1 *on a probability space* $(\Omega, \mathcal{F}, \mathbb{P})$. *Then*

$$\mathbb{P}\left(\frac{X_1 + \cdots + X_n}{\sqrt{n}} \le y\right) \to \mathfrak{N}(y) := \int_{-\infty}^{y} \frac{\mathrm{e}^{-v^2/2}}{\sqrt{2\pi}}\,\mathrm{d}v \quad \text{as } n \to \infty,$$

or, equivalently, letting $S_n := \sum_1^n X_k$,

$$\mathbb{E}f\left(S_n/\sqrt{n}\right) \to \int_{-\infty}^{\infty} f(v)\frac{\mathrm{e}^{-v^2/2}}{\sqrt{2\pi}}\,\mathrm{d}v \quad \text{as } n \to \infty, \text{ for every } f \in \mathrm{b}\mathcal{C}(\mathbb{R}).$$

Unicode text, non-unicode math

The following examples cover using a unicode latex engine (`lualatex` or `xelatex`) with XCharter otf fonts for text accompanied by newtxmath. They are simple variants of examples 5 and 6 from the prior subsection.

Example 7:

```
% an example using newtx.sty, works with all latex engines
\usepackage[<specify babel languages>]{babel}% load before newtx
\usepackage[scaled=1.04,varqu,varl]{inconsolata}% inconsolata tt
\usepackage[type1]{sourcesanspro}% sans serif for math
\usepackage[T1]{fontenc} % encoding to use for mathtt, etc
\usepackage[xcharter,osf,p,mathscale=1.05,textscale=0,uprightscript,vvarbb]{newtx}
% loads newtxmath
% newtx loads fontspec with unicode engines
\setmonofont{lmmono10-regular.otf}[Scale=1.08] % typewriter for text
\linespread{1.04}
% load polyglossia after newtx, if using
```

Example 8:

```
% Adds instructions to produce a pdf conforming to PDF/A-1b
%\pdfcompresslevel=0 % uncomment for debugging the pdf
%\pdfgentounicode=1
%\input glyphtounicode.tex  % now part of latex
\InputIfFileExists{glyphtounicode-cmr.tex}{}{}
\InputIfFileExists{glyphtounicode-ntx.tex}{}{}
\usepackage[a-1b]{pdfx} % version 1.6.4 or higher
\usepackage[<specify babel languages>]{babel}% load before XCharter
\usepackage[scaled=.98,sups,osf]{XCharter}% osf in text, lining figures in math
\usepackage[scaled=1.04,varqu,varl]{inconsolata}% inconsolata typewriter
\usepackage[type1]{sourcesanspro}% sans serif
\usepackage[uprightscript,charter,vvarbb,scaled=1.05]{newtxmath}
\linespread{1.04}
```

3 Text effects under `fontaxes`

This package loads the `fontaxes` package in order to access italic and slanted small caps. You should pay attention to the fact that `fontaxes` modifies the behavior of some basic LaTeX text macros such

as \textsc and \textup. Under normal LaTeX, some text effects are combined, so that, for example, \textbf{\textit{a}} produces bold italic a, while other effects are not, e.g., \textsc{\textup{a}} has the same effect as \textup{a}, producing the letter a in upright, not small cap, style. With fontaxes, \textsc{\textup{a}} produces instead upright small cap a. It offers a macro \textulc that undoes small caps, so that, e.g., \textsc{\textulc{a}} produces a in non-small cap mode, with whatever other style choices were in force, such as bold or italics.

4 Text Companion Issues

As of version 1.206, XCharter has essentially full support for textcomp and there should be no issues in using any of the macros like \textregistered and \textinterrobang (®, ⸘.)

XCharter-Roman-ts1

	′0	′1	′2	′3	′4	′5	′6	′7	
′00x	` 0	´ 1	ˆ 2	˜ 3	¨ 4	˝ 5	° 6	ˇ 7	″0x
′01x	˘ 8	¯ 9	˙ 10	¸ 11	˛ 12	, 13	14	15	
′02x	16	17	„ 18	19	20	— 21	— 22	23	″1x
′03x	← 24	→ 25	‿ 26	‿ 27	‿ 28	‿ 29	30	31	
′04x	ƀ 32	33	34	35	$ 36	37	38	' 39	″2x
′05x	40	41	* 42	43	, 44	= 45	. 46	/ 47	
′06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	″3x
′07x	8 56	9 57	58	59	⟨ 60	− 61	⟩ 62	63	
′10x	64	65	66	67	68	69	70	71	″4x
′11x	72	73	74	75	76	Ʊ 77	78	◯ 79	
′12x	80	81	82	83	84	85	86	Ω 87	″5x
′13x	88	89	90	⟦ 91	92	⟧ 93	↑ 94	↓ 95	
′14x	` 96	97	⋆ 98	o|o 99	† 100	101	102	103	″6x
′15x	104	105	106	107	108	ꝏ 109	110	111	
′16x	112	113	114	115	116	117	118	119	″7x
′17x	120	121	122	123	124	125	~ 126	= 127	
′20x	˘ 128	ˇ 129	˝ 130	131	† 132	‡ 133	|| 134	‰ 135	″8x
′21x	• 136	℃ 137	$ 138	¢ 139	ƒ 140	₵ 141	₩ 142	₦ 143	
′22x	₲ 144	₱ 145	£ 146	₨ 147	⸮ 148	ȧ 149	đ 150	™ 151	″9x
′23x	‰ 152	¶ 153	฿ 154	№ 155	℅ 156	℮ 157	° 158	SM 159	
′24x	[160	] 161	¢ 162	£ 163	¤ 164	¥ 165	¦ 166	§ 167	″Ax
′25x	¨ 168	© 169	a 170	ⓒ 171	¬ 172	Ⓟ 173	® 174	¯ 175	
′26x	° 176	± 177	2 178	3 179	´ 180	µ 181	¶ 182	· 183	″Bx
′27x	※ 184	1 185	o 186	√ 187	¼ 188	½ 189	¾ 190	€ 191	
′30x	192	193	194	195	196	197	198	199	″Cx
′31x	200	201	202	203	204	205	206	207	
′32x	208	209	210	211	212	213	× 214	215	″Dx
′33x	216	217	218	219	220	221	222	223	

′34x	224	225	226	227	228	229	230	231	″Ex
′35x	232	233	234	235	236	237	238	239	
′36x	240	241	242	243	244	245	÷ 246	247	″Fx
′37x	248	249	250	251	252	253	254	255	
	″8	″9	″A	″B	″C	″D	″E	″F	

5 Usage with fontspec

Because the package supplies a file named `XCharter.fontspec` whose contents list the `otf` files that correspond to each of Regular, Bold, Italic, BoldItalic, Slanted and BoldSlanted, you may load XCharter with just

```
\usepackage{fontspec}
\setmainfont{XCharter}
```

With unicode-encoded text, you will, in particular, have complete access to the Cyrillic glyphs.

6 XCharter and PDF/A

There are a number of PDF/A validators available, though their outputs can and do differ when applied to the same document. I've tried the following.

- Adobe Acrobat Pro. Almost all XCharter documents validate PDF/A-1b.
- `https://www.pdf-online.com/osa/validate.aspx` is a free online validator. Almost all XCharter documents validate PDF/A-1b.
- The free `veraPDF` validator is much stricter. Recent documents produced using XCharter since version 1.24 have validated correctly.

7 Using Cyrillic with pdflatex

The OT2 encoding, now considered as obsolete because it is 7-bit, is nonetheless useful to scholars who wish to write short segments using a Cyrillic script from a Western keyboard. There are two means of doing this, one using control sequences for the characters (e.g., `\CYRA` for Cyrillic A) and the other using ligatures to access the characters. Tables setting out the substitutions available may be consulted at `http://herbert.the-little-red-haired-girl.org/dvi/pdf/cyrillic.pdf`.

Note that, while the OT2 encoded font is complete, there are many gaps in the T2A encoded version, so that only Modern Russian and Ukrainian are fully covered, along with a number of characters from Old Russian and other Slavic languages.

XCharter-Roman-tlf-ot2.tfm:

	′0	*′1*	*′2*	*′3*	*′4*	*′5*	*′6*	*′7*	
′00x	Њ 0	Љ 1	Ц 2	Э 3	І 4	Є 5	Ђ 6	Ћ 7	″0x
′01x	њ 8	љ 9	ц 10	э 11	і 12	є 13	ђ 14	ћ 15	
′02x	Ю 16	Ж 17	Й 18	Ё 19	Ѵ 20	Ѳ 21	Ѕ 22	Я 23	″1x
′03x	ю 24	ж 25	й 26	ё 27	ѵ 28	ѳ 29	ѕ 30	я 31	
′04x	¨ 32	! 33	” 34	Ѣ 35	˘ 36	% 37	´ 38	’ 39	″2x
′05x	(40	) 41	* 42	ѣ 43	, 44	- 45	. 46	/ 47	
′06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	″3x
′07x	8 56	9 57	: 58	; 59	« 60	1 61	» 62	? 63	
′10x	ˇ 64	А 65	Б 66	Ц 67	Д 68	Е 69	Ф 70	Г 71	″4x
′11x	Х 72	И 73	Ј 74	К 75	Л 76	М 77	Н 78	О 79	
′12x	П 80	Ч 81	Р 82	С 83	Т 84	У 85	В 86	Щ 87	″5x
′13x	Ш 88	Ы 89	З 90	[91	“ 92	] 93	Ь 94	Ъ 95	
′14x	‘ 96	а 97	б 98	ц 99	д 100	е 101	ф 102	г 103	″6x
′15x	х 104	и 105	ј 106	к 107	л 108	м 109	н 110	о 111	
′16x	п 112	ч 113	р 114	с 115	т 116	у 117	в 118	щ 119	″7x
′17x	ш 120	ы 121	з 122	– 123	— 124	№ 125	ь 126	ъ 127	
′20x	128	129	130	131	132	133	134	135	″8x
′21x	136	137	138	139	140	141	142	143	
′22x	144	145	146	147	148	149	150	151	″9x
′23x	152	153	154	155	156	157	158	159	
′24x	160	161	162	163	164	165	166	167	″Ax
′25x	168	169	170	171	172	173	174	175	
′26x	176	177	178	179	180	181	182	183	″Bx
′27x	184	185	186	187	188	189	190	191	
′30x	192	193	194	195	196	197	198	199	″Cx
′31x	200	201	202	203	204	205	206	207	
′32x	208	209	210	211	212	213	214	215	″Dx
′33x	216	217	218	219	220	221	222	223	
′34x	224	225	226	227	228	229	230	231	″Ex
′35x	232	233	234	235	236	237	238	239	

′36x	240	241	242	243	244	245	246	247	″Fx
′37x	248	249	250	251	252	253	254	255	
	″8	″9	″A	″B	″C	″D	″E	″F	

This encoding contains the upright Sha glyph in slot 88. This may be used in mathematical formulas by defining `\def\Sha{{\usefont{OT2}{XCharter-TLF}{m}{n}\char88 }}` so that one may write `$\text{\Sha}(A/K)$` for the Tate–Shafarevich group Ш(A/K).

Example OT2 Preamble:

```
\documentclass{article}
\usepackage[OT2,T1]{fontenc} % loads ot2enc.def
\newcommand\cyrtext{%
\fontfamily{XCharter-TLF}\fontencoding{OT2}\selectfont} % declaration
\DeclareTextFontCommand{\textcyr}{\cyrtext}  %macro with argument
```

The Russian part of the following sentence is entered as `\textcyr{a e1to --- po-russki}`.

This is text in English, then Russian: а это — по-русски.

Using T2A with T1:

Here's an example of using XCharter text and math, set up to allow the use of Russian with English as the main language.

```
\usepackage[OT2,T2A,T1]{fontenc} % spell out all text encodings to be used
\usepackage[utf8]{inputenc} %
\usepackage{substitutefont} % so we can use fonts other than those specified in babel
\usepackage[russian,english]{babel}
\usepackage{XCharter} %
\usepackage[charter,vvarbb,scaled=1.07]{newtxmath}
\useosf % use oldstyle figures except in math
\substitutefont{T2A}{\rmdefault}{XCharter} % use XCharter to render Russian
%\substitutefont{OT2}{\rmdefault}{XCharter} % poor man's version
```

Any utf8-encoded text typed outside of a `\foreignlanguage{}{}` block will be rendered as T1-encoded XCharter, while that within `\foreignlanguage{russian}{}` will render as T2A-encoded Cyrillic.

XCharter-Roman-tlf-t2a.tfm:

	′0	′1	′2	′3	′4	′5	′6	′7	
′00x	` 0	´ 1	ˆ 2	˜ 3	¨ 4	″ 5	° 6	ˇ 7	″0x
′01x	˘ 8	¯ 9	˙ 10	¸ 11	˛ 12	I 13	fj 14	fb 15	
′02x	“ 16	” 17	fh 18	19	˘ 20	– 21	— 22	23	″1x
′03x	24	ı 25	ȷ 26	ff 27	fi 28	fl 29	ffi 30	ffl 31	
′04x	␣ 32	! 33	" 34	# 35	$ 36	% 37	& 38	’ 39	″2x
′05x	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47	
′06x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	″3x
′07x	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63	
′10x	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	″4x
′11x	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79	
′12x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	″5x
′13x	X 88	Y 89	Z 90	[91	\ 92	] 93	^ 94	_ 95	
′14x	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	″6x
′15x	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111	
′16x	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	″7x
′17x	x 120	y 121	z 122	{ 123	| 124	} 125	~ 126	- 127	
′20x	Ґ 128	129	Ђ 130	Ћ 131	132	133	134	Љ 135	″8x
′21x	Ї 136	137	138	139	Æ 140	141	142	S 143	
′22x	144	Ç 145	Ў 146	Y 147	148	149	Ц 150	151	″9x
′23x	152	Є 153	154	Нь 155	Ё 156	№ 157	¤ 158	§ 159	
′24x	ґ 160	161	ђ 162	ћ 163	h 164	165	166	љ 167	″Ax
′25x	ї 168	169	170	171	æ 172	173	174	s 175	
′26x	176	ç 177	ў 178	179	180	181	ц 182	183	″Bx
′27x	184	є 185	186	нь 187	ё 188	„ 189	« 190	» 191	
′30x	А 192	Б 193	В 194	Г 195	Д 196	Е 197	Ж 198	З 199	″Cx
′31x	И 200	Й 201	К 202	Л 203	М 204	Н 205	О 206	П 207	
′32x	Р 208	С 209	Т 210	У 211	Ф 212	Х 213	Ц 214	Ч 215	″Dx
′33x	Ш 216	Щ 217	Ъ 218	Ы 219	Ь 220	Э 221	Ю 222	Я 223	
′34x	а 224	б 225	в 226	г 227	д 228	е 229	ж 230	з 231	″Ex
′35x	и 232	й 233	к 234	л 235	м 236	н 237	о 238	п 239	
′36x	р 240	с 241	т 242	у 243	ф 244	х 245	ц 246	ч 247	″Fx
′37x	ш 248	щ 249	ъ 250	ы 251	ь 252	э 253	ю 254	я 255	
	″8	″9	″A	″B	″C	″D	″E	″F	

XCharter-Italic-tlf-t2a.tfm:

	′0	′1	′2	′3	′4	′5	′6	′7	
′00x	̀ 0	́ 1	̂ 2	̃ 3	̈ 4	̋ 5	̊ 6	̌ 7	″0x
′01x	̆ 8	̄ 9	̇ 10	̧ 11	̛ 12	*I* 13	*fb* 14	*fh* 15	
′02x	“ 16	” 17	*fj* 18	19	̨ 20	– 21	— 22	23	″1x
′03x	24	*ı* 25	*ȷ* 26	*ff* 27	*fi* 28	*fl* 29	*ffi* 30	*ffl* 31	
′04x	␣ 32	*!* 33	" 34	*#* 35	*$* 36	*%* 37	*&* 38	’ 39	″2x
′05x	*(* 40	*)* 41	* 42	*+* 43	*,* 44	- 45	. 46	⁄ 47	
′06x	*0* 48	*1* 49	*2* 50	*3* 51	*4* 52	*5* 53	*6* 54	*7* 55	″3x
′07x	*8* 56	*9* 57	*:* 58	*;* 59	*<* 60	*=* 61	*>* 62	*?* 63	
′10x	*@* 64	*A* 65	*B* 66	*C* 67	*D* 68	*E* 69	*F* 70	*G* 71	″4x
′11x	*H* 72	*I* 73	*J* 74	*K* 75	*L* 76	*M* 77	*N* 78	*O* 79	
′12x	*P* 80	*Q* 81	*R* 82	*S* 83	*T* 84	*U* 85	*V* 86	*W* 87	″5x
′13x	*X* 88	*Y* 89	*Z* 90	*[* 91	\ 92	*]* 93	^ 94	_ 95	
′14x	‘ 96	*a* 97	*b* 98	*c* 99	*d* 100	*e* 101	*f* 102	*g* 103	″6x
′15x	*h* 104	*i* 105	*j* 106	*k* 107	*l* 108	*m* 109	*n* 110	*o* 111	
′16x	*p* 112	*q* 113	*r* 114	*s* 115	*t* 116	*u* 117	*v* 118	*w* 119	″7x
′17x	*x* 120	*y* 121	*z* 122	{ 123	| 124	} 125	~ 126	- 127	
′20x	*Ґ* 128	129	*Ђ* 130	*Ћ* 131	132	133	134	*Љ* 135	″8x
′21x	*Ї* 136	137	138	139	*Æ* 140	141	142	*S* 143	
′22x	144	*Ç* 145	*Ў* 146	*Y* 147	148	149	*Ц* 150	151	″9x
′23x	152	*Є* 153	154	*Нь* 155	*Ё* 156	*№* 157	¤ 158	*§* 159	
′24x	*ґ* 160	161	*ђ* 162	*ћ* 163	*h* 164	165	166	*љ* 167	″Ax
′25x	*ї* 168	169	170	171	*æ* 172	173	174	*s* 175	
′26x	176	*ç* 177	*ў* 178	179	180	181	*ц* 182	183	″Bx
′27x	184	*є* 185	186	*нь* 187	*ё* 188	„ 189	« 190	» 191	
′30x	*А* 192	*Б* 193	*В* 194	*Г* 195	*Д* 196	*Е* 197	*Ж* 198	*З* 199	″Cx
′31x	*И* 200	*Й* 201	*К* 202	*Л* 203	*М* 204	*Н* 205	*О* 206	*П* 207	
′32x	*Р* 208	*С* 209	*Т* 210	*У* 211	*Ф* 212	*Х* 213	*Ц* 214	*Ч* 215	″Dx
′33x	*Ш* 216	*Щ* 217	*Ъ* 218	*Ы* 219	*Ь* 220	*Э* 221	*Ю* 222	*Я* 223	
′34x	*а* 224	*б* 225	*в* 226	*г* 227	*д* 228	*е* 229	*ж* 230	*з* 231	″Ex
′35x	*и* 232	*й* 233	*к* 234	*л* 235	*м* 236	*н* 237	*о* 238	*п* 239	
′36x	*р* 240	*с* 241	*т* 242	*у* 243	*ф* 244	*х* 245	*ц* 246	*ч* 247	″Fx
′37x	*ш* 248	*щ* 249	*ъ* 250	*ы* 251	*ь* 252	*э* 253	*ю* 254	*я* 255	
	″8	″9	″A	″B	″C	″D	″E	″F	

8 Serbian Cyrillic

Serbian and Russian Cyrillic differ in the following ways.

Character	Shape	Russian	Serbian
U+0431	Upright	б	б
U+0431	Italic	*б*	*б*
U+0433	Italic	*г*	*ѓ*
U+0434	Italic	*д*	*ђ*
U+043F	Italic	*п*	*ћ*
U+0442	Italic	*т*	*ќ*

Usage under LuaLaTeX and XeLaTeX is simple. Your preamble should include

```
\usepackage{polyglossia}
\usepackage{fontspec}
\setmainfont{XCharter}[%
Language=Serbian,
Script=Cyrillic
]
```

Then all (unicode) input characters will be typeset using the above substitution table.

The story is a bit more complicated with LaTeX processing.

Example 1: Serbian Cyrillic as default text.

```
\usepackage[utf8x]{inputenc}
\usepackage[serbianc]{babel}
\usepackage[serbianc]{XCharter}
\usepackage[T2A]{fontenc}
```

This will produce essentially the same output as the `fontspec` example above, with unicode input.

It may be be possible to work out a scheme that would allow multiple scripts and languages to be used with `serbianc` as the main or as a secondary language in `babel`, but I have not succeeded in doing this with XCharter, and know of no other example that I might crib from.

Opentype processing and German orthography

Prior to version `1.12`, `XCharter` offered only basic support for German orthography, having all required accented glyphs and the lower case ß, as well as a small caps ß. Under non-unicode LaTeX, the T1 encoding contained `S_S`. With unicode tex processing:

```
{\addfontfeature{StylisticSet=1}\ss\ \textsc{\ss}}
```

typesets, as in non-unicode LaTeX processing, to

ß ß

Note also that in unicode processing, in order to obtain the expected case change behavior, it may be necessary to add in your preamble:

```
\uccode`ß="1E9E
```

As of version 1.12 of XCharter, there are now glyphs in each style for U+1E9E and for its small caps version, accessible under unicode TeX. The glyphs may be used as the uppercase and small caps versions of germandbls. Currently, the new glyphs are not available in any of the LaTeX encodings and must be used via unicode TeX.

The following tables show how to access the new glyphs in unicode TeX. Note that you will need to set StylisticSet=1 if you wish not to use the new sharp-s glyphs.

New symbols in XCharter:

Glyph name	glyph	macro
uni1E9E	ß	\symbol{"1E9E}
uni1E9E.ss01	SS	{\addfontfeature{StylisticSet=1}\symbol{"1E9E}}
germandbls.sc	ß	{\textsc{\ss}}
germandbls.sc.ss01	ss	{\addfontfeature{StylisticSet=1}\textsc{\ss}}

Effect of choice of StylisticSet:

StylisticSet	\ss	\SS	\MakeUppercase{\ss}	\textsc{\ss}
None	ß	ß	ß	ß
=1	ß	SS	SS	ss

Choosing the shape of the oldstyle figure "one":

The default is 1. The choice is controlled by the feature cv01. To change this globally when you load XCharter, you may make oldstyle numbers the default with the alternate shape for "one":

```
\setmainfont{XCharter}[CharacterVariant={1:0},Numbers={OldStyle}]
```

To set this for one-time use, you can write:

```
{\addfontfeature{CharacterVariant={1:0},Numbers={OldStyle}}1}
```

to make 1 render as 1. (According to its documentation, the first index 1 in {1:0} refers to the CharacterVariant index, and the second index, 0, refers to the first choice specified among the variant forms.)

If you choose to load XCharter-*.otf using XCharter.sty or newtx, you may make use of the options osf, osfI or the macros \useosf, \useosfI to the same effect.