

The XCharter Font Package

Michael Sharpe

May 29, 2021

1 Package Features

The *XCharter* fonts are extensions of the Bitstream Charter fonts, adding oldstyle figures (proportionally spaced only), 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 serified fonts for viewing on screen, and bears a number of similarities to Charter, though the latter is heavier.)

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 (o123456789) with numeral one like a shortened 1, but math mode will always use lining figures;
- `oldstyleI` (or `osfI`) sets the figure style in text mode to oldstyle (oI23456789) with numeral one like a shortened I, but math mode will always use lining figures;
- `sup` 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.
- As of version 1.22, there is a theorem font option similar to those in `newtx` and `newpx`. A new font family, `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`.

SPECIAL MACROS:

- `\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}`.
- `\textin` prints its argument in inferior figures, e.g., `\textin{12}` results in ₁₂. The effect is the same with `{\instyle 12}`.
- `\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 using whatever oldstyle option is in force.
- `\textosf` prints its argument in oldstyle figures using, in effect, the `osf` option—e.g., `\textosf{12}` results in 12.
- `\textosfI` prints its argument in oldstyle figures using, in effect, the `osfI` option—e.g., `\textosfI{12}` results in 12.
- 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.
- 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 2³¹/₃₂. (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{X}` or `\textcircled{x}` render as \textcircled{X} . The macro works also for numerals: `\textcircled{2}` renders as $\textcircled{2}$.

- `\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) and `\textbf{\textth{Theorem font (01234):!}}` as ***Theorem font (01234):!***. The related font switch `\thfamily` is defined so that, in essence, `{\thfamily A12!}` and `\textth{A12!}` are equivalent. In opentype processing, the `StylisticSet 05` controls whether figures and punctuation are upright in italic shaped faces.

Three 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 ν . Moreover, it is not easy to redefine `\mathcal` to get a better math calligraphic alphabet—e.g., the `mathalfa` 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 ν , and is completely compatible with `mathalfa`. 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 ν . (The option `noxchvw` to `newtxmath` changes the *v* and *w* to be the original Charter italic glyphs, which may lead to issues with ν . The option `alty` to `newtxmath/charter`, new as of version 1.203, substitutes *y* for the default *y* which, IMO, works better in combination with other math symbols because it lacks the problematic tail of *y*.)

EXAMPLE 1:

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

EXAMPLE 2:

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

EXAMPLE 3:

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

EXAMPLE 4:

```
\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]{cabin}% sans serif
```

```
\usepackage[uprightsript,charter,vvarbb,scaled=1.05]{newtxmath}
\linespread{1.04}
```

EXAMPLE 5:

```
% Adds instructions to produce a pdf conforming to PDF/A-1b
\pdfcompresslevel=0
\pdfgentounicode=1
\input glyphtounicode.tex
\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]{cabin}% sans serif
\usepackage[uprightsript,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, \dots be a sequence of iid random variables with mean 0 and variance 1 on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Then

$$\mathbb{P}\left(\frac{X_1 + \dots + X_n}{\sqrt{n}} \leq y\right) \rightarrow \mathfrak{N}(y) := \int_{-\infty}^y \frac{e^{-v^2/2}}{\sqrt{2\pi}} dv \quad \text{as } n \rightarrow \infty,$$

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

$$\mathbb{E}f\left(\frac{S_n}{\sqrt{n}}\right) \rightarrow \int_{-\infty}^{\infty} f(v) \frac{e^{-v^2/2}}{\sqrt{2\pi}} dv \quad \text{as } n \rightarrow \infty, \text{ for every } f \in \text{bC}(\mathbb{R}).$$

Greek letters in version 1.11:

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

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

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

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

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

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

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

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

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

3 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 ($\text{\textcircled{R}}$, \textinterrobang).

XCHARTER-ROMAN-TS1

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	` 0	' 1	^ 2	~ 3	¨ 4	" 5	° 6	˘ 7	"0x
'01x	˘ 8	¯ 9	· 10	, 11	¸ 12	, 13	14	15	
'02x	16	17	„ 18	19	20	— 21	— 22	23	"1x
'03x	← 24	→ 25	ˆ 26	ˆ 27	ˆ 28	ˆ 29	30	31	
'04x	‡ 32	33	34	35	\$ 36	37	38	' 39	"2x
'05x	40	41	* 42	43	, 44	= 45	. 46	/ 47	
'06x	0 48	I 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	U 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	⊙ 109	110	111	
'16x	112	113	114	115	116	117	118	119	"7x
'17x	120	121	122	123	124	125	~ 126	= 127	
'20x	˘ 128	˘ 129	" 130	131	† 132	‡ 133	134	%o 135	"8x
'21x	• 136	°C 137	\$ 138	¢ 139	f 140	¢ 141	W 142	N 143	
'22x	€ 144	£ 145	£ 146	₠ 147	¥ 148	¤ 149	đ 150	™ 151	"9x
'23x	%000 152	¶ 153	₠ 154	№ 155	‰ 156	e 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	

$\overset{\cdot}{3}2x$	208	209	210	211	212	213	\times 214	215	"Dx
$\overset{\cdot}{3}3x$	216	217	218	219	220	221	222	223	
$\overset{\cdot}{3}6x$	240	241	242	243	244	245	\div 246	247	"Fx
$\overset{\cdot}{3}7x$	248	249	250	251	252	253	254	255	
	"8	"9	"A	"B	"C	"D	"E	"F	

4 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.

5 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. Most documents produced using `XCharter` failed.

6 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	Ѓ ₀	Ѕ ₁	Ц ₂	Э ₃	І ₄	Є ₅	Ђ ₆	Ѓ ₇	"0x
´01x	Ѓ ₈	Ѕ ₉	Ц ₁₀	Э ₁₁	і ₁₂	є ₁₃	ђ ₁₄	ѓ ₁₅	
´02x	Ю ₁₆	Ж ₁₇	Й ₁₈	Ё ₁₉	Ѳ ₂₀	Ө ₂₁	Ѕ ₂₂	Я ₂₃	"1x
´03x	ю ₂₄	ж ₂₅	й ₂₆	ё ₂₇	ѳ ₂₈	ө ₂₉	ѕ ₃₀	я ₃₁	
´04x	¨ ₃₂	! ₃₃	” ₃₄	Ђ ₃₅	˘ ₃₆	% ₃₇	’ ₃₈	’ ₃₉	"2x
´05x	(₄₀)) ₄₁	* ₄₂	Ђ 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	˘ 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	
	"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 $\text{III}(A/K)$.

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

NOTE: Currently, the \TeX support files for T2A encoded XCharter cover only the figure styles lining, oldstyle and superiors.

7 Serbian Cyrillic

Serbian and Russian Cyrillic differ in the following ways.

Character	Shape	Russian	Serbian
U+0431	Upright	б	б̄
U+0431	Italic	б	б̄
U+0433	Italic	з	ī
U+0434	Italic	д	g
U+043F	Italic	н	ū
U+0442	Italic	м	ū̄

Usage under XeLaTeX 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 \TeX 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 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 . Under LaTeX, the T1 encoding contained S_S. With unicode tex processing:

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

typesets, as in LaTeX processing, to

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.)