

Kp-fonts: OpenType version

Daniel FLIPO

18th November 2020

This bundle provides OpenType versions of Type1 Kp-fonts designed by Christophe Caignaert. See [Kpfonts-Doc-English.pdf](#) for the full documentation of the original Type1 fonts.

It is usable only with LuaTeX or XeTeX engines; it consists of sixteen Text OpenType fonts, a Roman family **KpRoman** in eight shapes and weights—*Regular, Italic, Bold, BoldItalic, Light, LightItalic, Semibold, SemiboldItalic*—, a Sans-Serif family **KpSans** and a TypeWriter family **KpMono**, each of them in four shapes and weights—*Regular, Italic, Bold* and *BoldItalic*— and five Math OpenType fonts **KpMath-Regular**, **KpMath-Bold**, **KpMath-Light**, **KpMath-Semibold** and **KpMath-Sans**.

KpRoman and **KpSans** families have small caps available in two sizes (SmallCaps and PetitesCaps), upper and lowercase digits (0123456789), ancient ligatures ct , st and Q a long-tailed capital Q. Superior and inferior digits and letters have been added to the OpenType **KpRoman** and **KpSans** fonts for footnotes' calls and abbreviations 1st, 2nd...

Latin and Greek letters are available in Upright and Italic shapes, in Bold and Regular weights, for all Math fonts: $\alpha \beta \Gamma \Delta$, $\alpha \beta \Gamma \Delta$, $\alpha \beta \Gamma \Delta$, $\alpha \beta \Gamma \Delta$.

Blackboard Bold capitals are available in two shapes, Serif and Sans: `\mathbb{ABC}` prints either $\mathbb{A}\mathbb{B}\mathbb{C}$ (option `[Style=bbsans]`) or $\mathbb{A}\mathbb{B}\mathbb{C}$ (default). Commands `\mathcal{ABC}` and `\mathscr{ABC}` print either $\mathcal{A}\mathcal{B}\mathcal{C}$ (default) or $\mathscr{A}\mathscr{B}\mathscr{C}$ with option `[Style=mathcal]`, .

File [unimath-kpfonts.pdf](#) shows the full list of Unicode Math symbols provided by Kp-fonts, compared with other common Math fonts. More symbols, specific to Kp-fonts, are listed in section ??.

A style file `kpfont-otf.sty` is provided to load Kp-fonts easily. It is derived from `kpfont.sty` but options differ.

Please beware that the current version (0.32) is *experimental*.

All fonts are covered by OFL licence, style file and documentation are under LPPL-1.3 licence.

oldstyle: provides lowercase digits as a default, long-tailed Q (Quebec) and (for Roman and Sans-Serif fonts only) old style ligatures « ct » et « st ».

Examples:

- *Quest* for an attractive font!
- *Quest* for an attractive font!
- **Quest** for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!
- *Quest* for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!

veryoldstyle: same as option **oldstyle** but the round ‘s’ is replaced by the long one ‘f’, unless it ends a word (then it remains ‘s’)² and ancient ligatures fi, fl, ft are activated. Coding **s=** prints a round ‘s’ anytime; in most cases this coding is not necessary with LuaTeX³, f.i. `\textit{some of Highlands' mysterious castles...}` will print *some of Highlands' mysterious castles...* which is correct; with XeTeX an = sign must be added at the end of **mysterious=** to get a round ‘s’ there.

largesmallcaps: prints larger SMALL CAPS than the default ones (PETITES CAPS).

1.3 Options for Math fonts *only*

lightmath: uses *light* Math fonts.

bbsans: command `\mathbb` prints Sans-Serif Blackboard Bold capitals with Serif fonts too: compare $\mathbb{C}, \mathbb{K}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}$, with $\mathbb{C}, \mathbb{K}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}$ (default).

mathcal: forces commands `\mathcal` and `\mathscr` to print $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ instead of $\mathscr{A}, \mathscr{B}, \mathscr{C}, \mathscr{D}$ (default).

frenchstyle: Latin uppercase letters and all Greek letters are printed upright, only lowercase latin letters are printed in italics; this follows the French typographic usage.

oldstylenumsmath: prints lowercase digits in Maths (default is uppercase).

narrowiints: prints condensed repeated integrals :

$\int\int\int$ et \iiint instead of $\int\int\int$ et \iiint (default).

partialup: the `\partial` symbol is printed upright ∂ instead of ∂ .

fancyReIm: commands `\Re` et `\Im` print \Re et \Im respectively instead of \mathbb{R} et \mathbb{I} .

tight: horizontal spaces tightened in math mode (same settings as **fourier-otf**).

noDcommand: do not define `\D` to avoid incompatibilities with other packages.

2 Another way to load Kp-fonts

Loading Kp-fonts through `kpfonts-otf.sty` offers only a limited choice of options; the standard commands `\setmainfont`, `\setsansfont`, `\setmonofont`, `\setmathfont` offer much more flexibility.

2. The OpenType **calt** feature is used to catch ‘s’ ending a word.

3. TeX’s ~ char (nobreakspace) fails to be recognised as ending a word: `boys~band` prints boyf band.

On the other hand, `kpfonts-otf.sty` defines a lot of useful commands to access AMS and specific Kp-fonts symbols. Loading `kpfonts-otf` with the `symbols` option enables to get all these commands defined without loading any font:

```
\usepackage[symbols]{kpfonts-otf}
```

Please note that `unicode-math`⁴ (and `fontspec`) are loaded by this procedure, no need to do it again, unless specific options are required, then `unicode-math` has to be loaded before `kpfonts-otf`, f.i.:

```
\usepackage[math-style=ISO,bold-style=upright]{unicode-math}
\usepackage[symbols]{kpfonts-otf}
```

Then, it is up to the user to load Kp-fonts with whatever option he/she likes using commands

```
\set...font{font}[options].
```

For documents requiring no Math fonts, loading `fontspec` and using the `\set...font` commands is enough.

2.1 Options for Text fonts

Here are the options available for Text Kp-fonts:

Numbers=Lowercase to get lowercase digits 1,2,3 instead of 1,2,3; the default is

Numbers=Lining.

SmallCapsFeatures = {Letters=SmallCaps} the `\textsc{}` command will print larger SMALL CAPS than the default PETITES CAPS.

The default setting is **SmallCapsFeatures = {Letters=PetiteCaps}**.

Ligatures=TeX (default) ' !` ?` -- --- print respectively ' ; ÿ - —.

Ligatures=Common (default) automatic ligatures ff ffi ffl fi fl.

Alternate=1 provides an alternative for glyphs ffi ffl fi fl (ff is unchanged).

Ligatures=Required: adds ft et tt ligatures.

Ligatures=Rare adds ct et st ligatures.

Style=Swash to get the long-tailed capital Q: Queen, also in small caps (both sizes):

QUEEN and QUEEN.

Style=Historic replaces any instance of 's' by the long variant f. It is still possible to get a round 's' coding it as 's='. As the long variant is never used at words' ends the feature `calc` is also activated: it tries to catch end of words, see `veryoldstyle` p. ?? for more details.

Ligatures=Historic switches specific ligatures for the long f: fi, fl, ft.

Options may be are chosen for each font, say:

4. A carefull reading of both manuals `unicode-math.pdf` and `fontspec.pdf` (available in all TeX distributions) is required in order to take full advantage of these packages.

```
\setmonofont{KpMono}[Numbers=Lowercase,Style=Historic]
```

but can also be shared by different typefaces:

```
\defaultfontfeatures+[KpRoman,KpSans,KpMono]{Numbers=Lowercase}
\defaultfontfeatures+[KpRoman,KpSans]{%
  Ligatures = Rare,
  Style = Swash,
  SmallCapsFeatures = {Letters=SmallCaps},
}
\setmainfont{KpRoman}
\setsansfont{KpSans}
\setmonofont{KpMono}
```

Notes : 1. `\setmonofont{KpMono}`, `\setsansfont{KpSans}`, `\setmainfont{KpRoman}` rely on files `KpMono.fontspec`, `KpSans.fontspec` and `KpRoman.fontspec` installed by `Kpfonts`.

2. Note the + ending `\defaultfontfeatures+` : options are *added*, not overwriting any other (including those of `fontspec.cfg`).

3. Options can be gathered: `Ligatures={Rare,Historic}` (with braces) is equivalent to `Ligatures=Rare` and `Ligatures=Historic`.

4. These options can also be switched on and off *locally* using `\addfontfeatures` inside a group, f.i. to print lowercase digits 1234576890 with a font loaded with option `Numbers=Lining`:

```
{\addfontfeatures{Numbers=Lowercase}1234576890}
```

Actually, a shortcut is available in this case: `\oldstylenums{1234576890}`.

2.2 Options for Math fonts

The following options can be passed either to `unicode-math`⁵ or to `\setmathfont{}`:

`math-style = ISO, TeX` (défaut), `french, upright`;

`bold-style = ISO, TeX` (défaut), `upright`;

`partial = upright` (default italic);

`nabla = italic` (default upright);

Seven ‘Style Variants’ are available with Kp-fonts, here are the first three:

`Style=mathcal` (+ss01) commands `\mathcal{}` and `\mathscr{}` print ABC instead of \mathcal{ABC} (default);

`Style=bbsans` (+ss02) `\mathbb{}` prints Sans-Serif Blackboard bold capitals ABC for Serif Math fonts `KpMath-Regular` and `KpMath-light` instead of \mathbb{ABC} ;

`Style=narrowiints` (+ss03) provides condensed repeated integrals: \iiint instead of \iiint (default).

5. See the manual `unicode-math.pdf`.

The next four tables present the other Style Variants available:

Table 1 – Style=leqslant (+ss04)

Command	Default	Variant
<code>\leq</code>	\leq	\leqslant
<code>\geq</code>	\geq	\geqslant
<code>\nleq</code>	$\not\leq$	$\not\leqslant$
<code>\ngeq</code>	$\not\geq$	$\not\geqslant$
<code>\leqq</code>	$\leq\leq$	$\leq\leqslant$
<code>\geqq</code>	$\geq\geq$	$\geq\geqslant$
<code>\leqless</code>	$\leq\lessdot$	$\leq\lessdotslant$
<code>\leqgtr</code>	$\leq\gtrdot$	$\leq\gtrdotslant$
<code>\lesseqgtr</code>	$\leq\gtrdot\lessdot$	$\leq\gtrdot\lessdotslant$
<code>\gtreqless</code>	$\gtrdot\lessdot$	$\gtrdot\lessdotslant$
<code>\lesseqqgtr</code>	$\leq\gtrdot\lessdot\lessdot$	$\leq\gtrdot\lessdot\lessdotslant$
<code>\gtreqqless</code>	$\gtrdot\lessdot\lessdot$	$\gtrdot\lessdot\lessdotslant$

Table 2 – Style=smaller (+ss05)

Command	Default	Variant
<code>\mid</code>	$ $	$\! $
<code>\nmid</code>	$\! $	$\! $
<code>\parallel</code>	\parallel	\parallel
<code>\nparallel</code>	$\not\parallel$	$\not\parallel$
<code>\parallelslant</code>	\parallel	\parallel
<code>\nparallelslant</code>	$\not\parallel$	$\not\parallel$

Table 3 – Style=subsetneq (+ss06)

Command	Default	Variant
<code>\subsetneq</code>	\subsetneq	\subsetneq
<code>\supsetneq</code>	\supsetneq	\supsetneq
<code>\subsetneqq</code>	\subsetneqq	\subsetneqq
<code>\supsetneqq</code>	\supsetneqq	\supsetneqq

Table 4 – Style=parallelslant (+ss07)

Command	Default	Variant
<code>\parallel</code>	\parallel	\parallel
<code>\nparallel</code>	$\not\parallel$	$\not\parallel$
<code>\shortparallel</code>	\parallel	\parallel
<code>\nshortparallel</code>	$\not\parallel$	$\not\parallel$

Example: switching styles 4 (leqslant) and 6 (subsetneq) can be achieved coding either `\setmathfont{KpMath-Regular.otf}[StylisticSet={4,6}]` or `\setmathfont{KpMath-Regular.otf}[Style={leqslant,subsetneq}]` but this second syntax is available only if `kpfonts-otf.sty` has been loaded (eventually with the `symbols` option).

Table ?? shows the available ‘Glyphs Variants’:

Example: with `\setmathfont{KpMath-Regular.otf}[CharacterVariant={3,6}]` commands `\epsilon` and `\phi` print ϵ and ϕ instead of ϵ et ϕ . The same is true of course for all shapes and and weights (upright, bold, bolditalic, sans-derif, etc.): f.i. with `math-syle=french`, `\epsilon` and `\phi` print ϵ and ϕ .

Note about `\hbar`: `unicode-math` defines `\hbar` as `\hslash` (U+210F) while `amsmath` provides two different glyphs (italic h with horizontal or diagonal stroke). `kpfonts-otf` now follows `unicode-math`; the italic h with horizontal stroke can be printed using `\hslash` or `\hbar` together with character variant `cv01` or with `\mithbar` (replacement for AMS’ command `\hbar`).

Table 5 – Glyphs Variants

	Default		Variant		Command	
cv00	\Re	\Im	\Re	\Im	<code>\Re</code>	<code>\Im</code>
cv01	\hbar		\hbar		<code>\hslash</code> or <code>\hbar</code>	
cv02	\emptyset		\emptyset		<code>\emptyset</code>	
cv03	ϵ		ϵ		<code>\epsilon</code>	
cv04	κ		κ		<code>\kappa</code>	
cv05	π		ϖ		<code>\pi</code>	
cv06	ϕ		φ		<code>\phi</code>	
cv07	ρ		ρ		<code>\rho</code>	
cv08	σ		ς		<code>\sigma</code>	
cv09	θ		ϑ		<code>\theta</code>	
cv10	Θ		Θ		<code>\Theta</code>	

3 Kp-fonts specific commands

3.1 Integrals

Kp-fonts offers variants for integral symbols suitable for indefinite integrals, they are coded as `\varint`, `\variint`, `\variiint`, `\variiiiint` and `\varidotsint`. Compare $\int f(t) dt$ and $\int f(t) dt$ and also

$$\int f(t) dt \quad \text{and} \quad \int f(t) dt$$

`\D{...}` prints an upright ‘d’ and improves kernings around the differential element: `\displaystyle\varint f(t)\D{t}` donne $\int f(t) dt$.

3.2 Specific Math symbols

The next tables present symbols unavailable as Unicode characters, they are coded in Kp-fonts’ private zone.

<code>\mmapsto</code>	\mapsto	<code>\longmmapsto</code>	\mapsto
<code>\mmapsfrom</code>	\mapsfrom	<code>\longmmapsfrom</code>	\mapsfrom
<code>\Mmapsto</code>	\Mmapsto	<code>\Longmmapsto</code>	\Mmapsto
<code>\Mmapsfrom</code>	\Mmapsfrom	<code>\Longmmapsfrom</code>	\Mmapsfrom
<code>\leftrightdasharrow</code>	\leftrightarrow	<code>\leadsto</code>	\leadsto
<code>\boxright</code>	\boxrightarrow	<code>\boxleft</code>	\boxleftarrow
<code>\circleright</code>	\circrightarrow	<code>\circleleft</code>	\circleftarrow
<code>\Diamondright</code>	\diamondrightarrow	<code>\Diamondleft</code>	\diamondleftarrow
<code>\boxdotright</code>	\boxdotrightarrow	<code>\boxdotleft</code>	\boxdotleftarrow
<code>\circledotright</code>	\circledrightarrow	<code>\circledotleft</code>	\circledleftarrow
<code>\Diamonddotright</code>	\diamonddotrightarrow	<code>\Diamonddotleft</code>	\diamonddotleftarrow

<code>\boxRight</code>	$\square \rightarrow$	<code>\boxLeft</code>	$\leftarrow \square$		
<code>\boxdotRight</code>	$\square \dot{\rightarrow}$	<code>\boxdotLeft</code>	$\dot{\leftarrow} \square$		
<code>\DiamondRight</code>	$\diamond \rightarrow$	<code>\DiamondLeft</code>	$\leftarrow \diamond$		
<code>\DiamonddotRight</code>	$\diamond \dot{\rightarrow}$	<code>\DiamonddotLeft</code>	$\dot{\leftarrow} \diamond$		
<code>\multimapdot</code>	$\multimap \cdot$	<code>\multimapdotinv</code>	$\cdot \multimap$		
<code>\multimapdotboth</code>	$\multimap \cdot \cdot$	<code>\multimapdotbothvert</code>	$\cdot \cdot \multimap$		
<code>\multimapbothvert</code>	$\cdot \cdot \multimap$	<code>\multimapdotbothBvert</code>	$\cdot \cdot \multimap$		
<code>\multimapdotbothAvert</code>	$\cdot \cdot \multimap$	<code>\multimapdotbothBvert</code>	$\cdot \cdot \multimap$		
<code>\capplus</code>	$\cap \oplus$	<code>\sqcupplus</code>	$\sqcup \oplus$	<code>\sqcapplus</code>	$\sqcap \oplus$
<code>\parallelslant</code>	$\parallel \text{slant}$	<code>\colonsim</code>	\sim	<code>\colonapprox</code>	\approx
<code>\parallelbackslant</code>	$\parallel \text{backslant}$	<code>\Colonsim</code>	\sim	<code>\Colonapprox</code>	\approx
<code>\eqqColon</code>	\equiv	<code>\Colondash</code>	\dashv	<code>\dashColon</code>	\dashv
<code>\strictif</code>	\rightarrow	<code>\strictfi</code>	\leftarrow	<code>\strictiff</code>	\leftrightarrow
<code>\circledvee</code>	$\vee \circ$	<code>\circledwedge</code>	$\wedge \circ$	<code>\circledbar</code>	$\bar{\circ}$
<code>\openJoin</code>	\times	<code>\opentimes</code>	\times	<code>\VvDash</code>	\Vdash
<code>\lambdaslash</code>	$\lambda \text{ slash}$	<code>\lambdabar</code>	$\lambda \text{ bar}$	<code>\Wr</code>	\wr
<code>\idotsint</code>	$\int \dots \int$	<code>\idotsint</code>	$\int \dots \int$	<code>\idotsint</code>	$\int \dots \int$
<code>\ointclockwise</code>	$\oint \text{clockwise}$	<code>\ointclockwise</code>	$\oint \text{clockwise}$	<code>\varointctrlockwise</code>	$\varoint \text{ctrlockwise}$
<code>\oiintclockwise</code>	$\oiint \text{clockwise}$	<code>\oiintclockwise</code>	$\oiint \text{clockwise}$	<code>\oiintctrlockwise</code>	$\oiint \text{ctrlockwise}$
<code>\varoiintclockwise</code>	$\varoiint \text{clockwise}$	<code>\varoiintclockwise</code>	$\varoiint \text{clockwise}$	<code>\varoiintctrlockwise</code>	$\varoiint \text{ctrlockwise}$
<code>\oiiintclockwise</code>	$\oiiint \text{clockwise}$	<code>\oiiintclockwise</code>	$\oiiint \text{clockwise}$	<code>\oiiintctrlockwise</code>	$\oiiint \text{ctrlockwise}$
<code>\varoiiintclockwise</code>	$\varoiiint \text{clockwise}$	<code>\varoiiintclockwise</code>	$\varoiiint \text{clockwise}$	<code>\varoiiintctrlockwise</code>	$\varoiiint \text{ctrlockwise}$
<code>\sqiint</code>	\sqiint	<code>\sqiint</code>	\sqiint	<code>\sqiint</code>	\sqiint

The full list of Unicode symbols available with Kp-fonts is shown in file [unimath-kpfonts.pdf](#).

3.3 Wide accents

- `\widehat` and `\widetilde`

\hat{x} \widehat{xx} \widehat{xxx} \widehat{xxxx} \widehat{xxxxx} \widehat{xxxxxx} \tilde{x} \widetilde{xx} \widetilde{xxx} \widetilde{xxxx} \widetilde{xxxxx} \widetilde{xxxxxx}

- `\overline` and `\underline`

\overline{x} \overline{xy} \overline{xyz} $\overline{A \cup B}$ $\overline{A \cup (B \cap C) \cup D}$ $\underline{m+n+p}$

- `\wideoverbar`, `\widecheck` et `\widebreve`

\wideoverbar{x} \wideoverbar{xy} \wideoverbar{xyz} \widecheck{x} \widecheck{xxxx} \widecheck{xxxxxx} \widebreve{x} \widebreve{xxx} \widebreve{xxxxxx}

- `\overparen` and `\underparen`

\overparen{x} \overparen{xy} \overparen{xyz} $\overparen{A \cup B}$ $\overparen{A \cup (B \cap C) \cup D}$ $\underparen{x+y}$ $\underparen{a+b+\dots+z}$

\underline{x} \underline{xz} \underline{xyz} $\underline{x+z}$ $\underline{a+b+\dots+z}$

- `\overbrace` and `\underbrace`

\overbrace{a} \overbrace{ab} \overbrace{abc} \overbrace{abcd} \overbrace{abcde} $\overbrace{a+b+c}^3$ $\overbrace{a+b+\dots+z}^{26}$

\underbrace{a} \underbrace{ab} \underbrace{abc} \underbrace{abcd} \underbrace{abcde} $\underbrace{a+b+c}_3$ $\underbrace{a+b+\dots+z}_{26}$

- `\overrightarrow` and `\overleftarrow`

\overrightarrow{v} \overrightarrow{M} \overrightarrow{vv} \overrightarrow{AB} \overrightarrow{ABC} \overrightarrow{ABCD} $\overrightarrow{ABCDEFGH}$

\overleftarrow{v} \overleftarrow{M} \overleftarrow{vv} \overleftarrow{AB} \overleftarrow{ABC} \overleftarrow{ABCD} $\overleftarrow{ABCDEFGH}$

- Enfin `\widearc` and `\widearcarrow` (ou `\overrightarrowarc`)

\widearc{AMB} \widearcarrow{AMB}

3.4 Math Versions

Different versions of the `KpMath` fonts may be defined in the document's preamble:

`\setmathfont{KpMath-Regular.otf}[version=base, options]`

`\setmathfont{KpMath-Bold.otf}[version=bold, options]`

`\setmathfont{KpMath-Semibold.otf}[version=semibold, options]`

`\setmathfont{KpMath-Sans.otf}[version=sans, options]`

`\setmathfont{KpMath-Light.otf}[version=light, options]`

then, it is easy to switch from one version to another one with `\mathversion{name}`.

Example⁶ :

```
\setmathfont{KpMath-Regular.otf}[Style=leqslant, CharacterVariant=3]
\setmathfont{KpMath-Bold.otf}[version=bold,
    Style=leqslant, CharacterVariant=3]
\setmathfont{KpMath-Sans.otf}[version=sans,
    Style=leqslant, CharacterVariant=3]
```

Here is the same equation in three versions, normal, bold and Sans-Serif:

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{bold}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{sans}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

6. Option `CharacterVariant=3` changes ϵ into ε .