

# **T<sub>E</sub>XFONT** **explained**

## Introduction

It is probably a known fact that  $\text{T}_{\text{E}}\text{X}$  can handle fonts of any kind, as well as satisfy the needs of those who do not live in english speaking countries. Nevertheless, the majority of documents typeset by  $\text{T}_{\text{E}}\text{X}$ , use the more or less standard Computer Modern Typefaces in an encoding not that well suited for non english usage.

The (subtle) relations between font encodings, hyphenation patterns, input encodings, operating systems, font metrics, and alike makes font handling the most complicated (and fuzzy) part of macro packages.

Although  $\text{T}_{\text{E}}\text{X}$  distributions come with many fonts, in often obscure names, made even more obscure by encoding specific instances and/or exceptions. Browsing the font sub-tree of the `texmf` directory can drive you crazy.

As a result, installing a font, or at least making sure that the fonts installed already can be used, is not trivial. It is not enough to make your macro package aware of their existence, you also should inform the post processor (which in the case of  $\text{PDF}_{\text{E}}\text{X}$  is build in) of (re)encoding issues of shape manipulations. And of course each back-end has its own demands.

In  $\text{CON}_{\text{E}}\text{X}_{\text{T}}$ , font support is implemented in layers. One can use an individual font, define a relationship between them, and collect such relationships in typeface collections. These issues are covered in the font manual that comes with  $\text{CON}_{\text{E}}\text{X}_{\text{T}}$ .

In this manual we will describe  $\text{T}_{\text{E}}\text{X}_{\text{FONT}}$ , a PERL script that can help you to manage the files that are needed for  $\text{T}_{\text{E}}\text{X}$  and the back-ends. We hope that this tool is of help, although we cannot guarantee that life will be easy from now on. The script only covers ‘normal’ fonts. For setting up math fonts, vendors often provide the necessary files and  $\text{CON}_{\text{E}}\text{X}_{\text{T}}$  is aware of these files. Special fonts, like expert fonts, assume a more in depth knowledge of font handling. We may deal with them in the future.

The more demanding user can of course fall back on more complicated tools like `fontinst`. Although I never used this tool, my impression is that it does a good job on manipulating fonts, but most users will not need this kind of fine tuning.

*Currently this manual is directed towards  $\text{CON}_{\text{E}}\text{X}_{\text{T}}$  users who also use  $\text{PDF}_{\text{E}}\text{X}$ . Future versions may also cover DVI (to PDF) drivers.*

## Organizing fonts

In the T<sub>E</sub>X community, much comes for free. This is definitely not true for fonts, although a decent free collection is available to get you started. There are thousands of interesting fonts out there, and occasionally you may want to buy one.

So, what to do when this floppy or CDROM arrives. A quick look at the contents will show you that there are at least `pfb` and `afm` files on it. If not, then you have a problem. But, given that these files are there, where should they go?

Regular T<sub>E</sub>X distributions are organized conforming the T<sub>E</sub>X Directory Structure. There you will find the commonly used and stable components under the main tree. In this tree you will find a font sub-tree:

`tex/texmf/fonts`

Any deviation from the default components, being your own extensions or updates, should go into a local tree:

`tex/texmf-local/fonts`

The main tree is preferably just a copy of for instance the one that comes on T<sub>E</sub>X Live. The `tex/texmf-local` path is where you normally will unzip a CON<sub>T</sub>E<sub>X</sub>T update. This is also a natural place to put your fonts, given that you have access to this tree. That way, you can easily replace the main tree without spoiling your local settings.

When you buy a font for your own usage only, the local tree is a good place for them. But when you want to share them you need to be aware of licensing issues. Licences may permit installation on 5 machines, printing on 1 printer, placement on a server with restricted access for 3 simultaneous users, and alike.

So, this is why it may make sense to introduce another tree, which we can maintain separately. In this tree we put those files that are needed to use the fresh fonts. So we may have at least the following paths:

`tex/texmf-fonts/fonts`

`tex/texmf-fonts/pdftex/config`

Below the `/fonts` subpath the font specific files are organized according to vendor and collection.

```
texmf / fonts / tfm      / vendor / name / *.tfm
           / afm      / vendor / name / *.afm
           / pfm      / vendor / name / *.pfm
           / vf       / vendor / name / *.vf
           / type1    / vendor / name / *.pfb
```

The PDF $\TeX$  directory is organized as follows

```
texmf / pdftex / config /          *.cfg
           / config /          *.map
           / config / encoding / *.enc
```

Of course you need to set up a couple of environment variables in order to let your  $\TeX$  system understand this all. Here we assume that you use WEB2C.

```
TEXMFMAIN=/tex/texmf
TEXMFLOCAL=/tex/texmf-local
TEXMFFONTS=/tex/texmf-fonts
```

The order of searching these trees is determined by the following variable:

```
TEXMF={ $TEXMFFONTS, $TEXMFLOCAL, !!$TEXMFMAIN }
```

Specific settings can be taken care of in the `texmf.cnf` file. The previous definition could have gone there.

```
TEXMFCNF=/tex/texmf-local/web2c
```

Since  $\TeX$ FONT looks for both the local and dedicated font tree, you're free in your choice, although from the perspective of managing fonts it may be handy to use the dedicated tree.

Back to the question "Where should my font files go?" we can now answer: "In the font tree.". Say that you've bought Officina from ITC, you can put the files in:

```
/tex/texmf-fonts/fonts/source/itc/officina
```

The files that  $\TeX$  needs for using these fonts, will later be moved to and/or put in:

```
/tex/texmf-fonts/fonts/afm/itc/officina
/tex/texmf-fonts/fonts/tfm/itc/officina
/tex/texmf-fonts/fonts/vf/itc/officina
/tex/texmf-fonts/fonts/type1/itc/officina
```

Here, `type1` does not denote a suffix, but a more verbose naming of `pfb` files. The `tfm` files contain the metric information that  $\text{T}_{\text{E}}\text{X}$  needs during the typesetting process, while the virtual font files `vf` are needed in the postprocessing stage to sort out how glyphs are composed (if they are composed at all).

## Generating metrics

Before you can set up  $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$  to use fonts, you need to generate the metric files needed.

```
texfont --ve=itc --co=officina --ma --in
```

This command means as much as: we're going to handle the Officina collection from vendor ITC, make all directories needed, and install the files there. Installation here comes down to copying the `afm` and `pfb` files, as well as generating the `tfm` and `vf` files.

We started in the installation path with files like:

```
ovb____.afm
ovb____.pfb
```

and end up with files spread over the font tree named:

```
/tex/fonts/afm/itc/officina/ovb____.afm
/tex/fonts/type1/itc/officina/ovb____.pfb
/tex/fonts/tfm/itc/officina/raw-ovb.tfm
/tex/fonts/tfm/itc/officina/texnansi-ovb.tfm
/tex/fonts/vf/itc/officina/texnansi-raw-ovb.tfm
```

From this you can deduce that we clean up names, as well as handle an encoding vector. By default we use the `texnansi` vector, but you can specify another one if needed:

```
texfont --en=ec --ve=itc --co=officina --ma --in
```

$\text{T}_{\text{E}}\text{X}\text{FONT}$  also generates a  $\text{T}_{\text{E}}\text{X}$  file that demonstrates the usage of these fonts. Normally you will define typescripts to take care of this, or use the predefined ones that come with  $\text{CON}_{\text{T}}\text{E}_{\text{X}}\text{T}$ .

```
\definefontsynonym[OfficinaSerif-Bold][ec-ovb][encoding=ec]
```

We also need to instruct PDF<sub>T</sub><sub>E</sub>X on how to embed the font. For this purpose, a map file is generated:

```
\loadmapfile[ec-itc-officina.map]
```

In this map file, you will find lines like:

```
ec-raw-ovb OfficinaSerif-Bold 4 < ovb____.pfb ec.enc
```

In practice it means that T<sub>E</sub>X will base its typesetting decisions on the metric file, but for inclusion will fall back on the associated (to be reencoded) file. This file has references to the raw file and these references are resolved in the map file.

The default encoding is `texnansi`. For western languages, `8r` and `ec` are also a good choice. But, in any case, make sure that all the characters that you need are there by processing the generated `tex` file.

```
texexec ec-itc-officina
```

If this file processes all right, then at least you know that you have the correct files on your system. When processing this file, you can enable compact mode.

```
texexec ec-itc-officina --mode=compact
```

## Example

Normally you need more than one run of T<sub>E</sub>X<sub>FONT</sub> to create the files needed. The next sequence removes old instances and installs the new fonts. We assume that you have put the `afm` and `pfb` files on the sourcepath relative to the current location.

```
texfont --ve=itc --co=officina --re
```

```
texfont --ve=itc --co=officina --so=itc/officina --ma --in
```

```
texfont --ve=itc --co=officina --so=itc/officina --ca=* ovbk_*
```

```
texfont --ve=itc --co=officina --so=itc/officina --sl=* ovbk_*
```

```
texfont --ve=itc --co=officina --so=itc/officina --sl=* ovb_*
```

```
texfont --ve=itc --co=officina --so=itc/officina --ca=* owbk_*
```

```
texfont --ve=itc --co=officina --so=itc/officina --sl=* owbk_*
```

```
texfont --ve=itc --co=officina --so=itc/officina --sl=* owb_*
```

For setting up CON<sub>T</sub><sub>E</sub>X<sub>T</sub> to actually use these fonts, you need to take a look at the `tex` file that is generated and/or take a look at the font manual. Of course you can

ask around to see if someone already has the typescripts made. Without comment we show how such a definition looks. We assume that you put these definitions in a file called `typeface.tex`.

```
\starttypescript [map] [8r]
\loadmapfile [8r-itc-officina]
\stoptypescript

\starttypescript [serif] [officina] [name]
\definefontsynonym[Serif] [OfficinaSerif-Book]
\definefontsynonym[SerifItalic] [OfficinaSerif-BookItalic]
\definefontsynonym[SerifSlanted] [OfficinaSerif-BookSlanted]
\definefontsynonym[SerifBold] [OfficinaSerif-Bold]
\definefontsynonym[SerifBoldItalic] [OfficinaSerif-BoldItalic]
\definefontsynonym[SerifBoldSlanted] [OfficinaSerif-BoldSlanted]
\definefontsynonym[SerifCaps] [OfficinaSerif-Caps]
\stoptypescript

\starttypescript [sans] [officina] [name]
\definefontsynonym[Sans] [OfficinaSans-Book]
\definefontsynonym[SansItalic] [OfficinaSans-BookItalic]
\definefontsynonym[SansSlanted] [OfficinaSans-BookSlanted]
\definefontsynonym[SansBold] [OfficinaSans-Bold]
\definefontsynonym[SansBoldItalic] [OfficinaSans-BoldItalic]
\definefontsynonym[SansBoldSlanted] [OfficinaSans-BoldSlanted]
\definefontsynonym[SansCaps] [OfficinaSans-Caps]
\stoptypescript

\starttypescript [serif] [officina] [8r]
\definefontsynonym[OfficinaSerif-Book] [8r-ovbk] [encoding=8r]
\definefontsynonym[OfficinaSerif-BookItalic] [8r-ovwi] [encoding=8r]
\definefontsynonym[OfficinaSerif-Bold] [8r-ovb] [encoding=8r]
\definefontsynonym[OfficinaSerif-BoldItalic] [8r-ovbi] [encoding=8r]

\definefontsynonym[OfficinaSerif-BookSlanted] [8r-ovbk-slanted-167] [encoding=8r]
\definefontsynonym[OfficinaSerif-BoldSlanted] [8r-ovb-slanted-167] [encoding=8r]
\definefontsynonym[OfficinaSerif-Caps] [8r-ovbk-capitalized-800] [encoding=8r]
\stoptypescript

\starttypescript [sans] [officina] [8r]
\definefontsynonym[OfficinaSans-Book] [8r-owbk] [encoding=8r]
\definefontsynonym[OfficinaSans-BookItalic] [8r-owwi] [encoding=8r]
\definefontsynonym[OfficinaSans-Bold] [8r-owb] [encoding=8r]
\definefontsynonym[OfficinaSans-BoldItalic] [8r-owbi] [encoding=8r]
\definefontsynonym[OfficinaSans-BookSlanted] [8r-owbk-slanted-167] [encoding=8r]
\definefontsynonym[OfficinaSans-BoldSlanted] [8r-owb-slanted-167] [encoding=8r]
\definefontsynonym[OfficinaSans-Caps] [8r-owbk-capitalized-800] [encoding=8r]
\stoptypescript
```

In your document style, you can now stick to simple definitions like:

```
\usetypecriptfile[typeface]
```

```
\definetypface[officina][rm][serif][officina][default][encoding=8r]
```

```
\definetypface[officina][ss][sans][officina][default][encoding=8r]
```

```
\setupbodyfont[officina,rm,10pt]
```

Of course you can mix these fonts with other ones, in which case you may want to apply relative scaling first. The fonts manual also explains how you can set up this font to use protruding characters. Since we use symbolic names, you can also use these to access the fonts in for instance special situations, like when you construct a title page:

```
\definefont[TitleFont][Sans at 72pt]
```

```
\definefont[TitleFont][OfficinaSans-Book at 72pt]
```

Although you can also access the file name directly, these methods are more descriptive.

## Tweaking shapes

A font is seldom just one shape and file. Apart from the upright shape, there can be an italic, bold and bold-italic alternative, or, oblique, bold and bold-oblique. Italic and oblique share their forward slanted look. By applying appropriate transformations, you can slant any font or make it wider or narrower.

So, we have at our hands, either or not by manipulation, normal, italic or oblique, slanted, as well as their bold variants. The following command creates a slanted bold officina font.

```
texfont --ve=itc --co=officina --sl=* ovb
```

Instead of a `*`, you can provide a number. The default slant is `.167`. Another manipulation is to widen a font, using the extend key:

```
texfont --ve=itc --co=officina --ex=1.2 ovb
```

These commands lead to font metric files with names as:



```
texnansi-ovb-slanted-167.tfm
texnansi-ovb-extended-1200.tfm
```

Combinations are also possible. In the default T<sub>E</sub>X distributions, where the 8 character file name limit is still honoured, a less verbose naming scheme is used. Our alternative is less efficient, but opens the possibility to have more than one slanted alternative.

Since T<sub>E</sub>XFONT is mainly a wrapper around `afm2tfm`, we also provide a third manipulation: creating small caps fonts.

```
texfont --ve=itc --co=officina --ca=* ovb
```

We now get:

```
texnansi-ovb-capitalized-800.tfm
```

## Generating instances

The Computer Modern Roman typefaces were created by Donald Knuth and their shapes are described as programs. There are quite some design axis and parameters that can be tweaked in order to get different instances. This is why they qualify as meta-fonts. In its own way, Adobe has created the Multiple Master Fonts format.<sup>1</sup>

If you decide (or are forced) to use a multiple master font, you need to make sure that you get all the files needed to tweak them. For instance, for using the Myriad fonts, you need files like:

MyriadMM-LightCn.afm	MyriadMM-LightCnIt.afm
MyriadMM-BlackCn.afm	MyriadMM-BlackCnIt.afm
MyriadMM-LightSemiEx.afm	MyriadMM-LightSemiExIt.afm
MyriadMM-BlackSemiEx.afm	MyriadMM-BlackSemiExIt.afm
MyriadMM.amfm	MyriadMM-It.amfm
MyriadMM.pfb	MyriadMM-It.pfb

When the author bought this font, it came as an install binary, that needed the Adobe Type Manager. After a couple of failures, he finally found some `pfb` and `mmm` files on my system.

---

Probably due to lack of interest, Adobe is no longer advocating this format, although it will survive in

<sup>1</sup> Open Type fonts.

You need the programs `mmafm` and `mmpfb` to create the instances.<sup>2</sup> These programs need an `amfm` file, which did not come on the floppy, but a bit of emailing and renaming finally lead to the files mentioned previously.

The following commands will create an acceptable collection of normal and italic, bold and bold italic shapes.

```
texfont --we=400 --wi=600 MyriadMM
texfont --we=700 --wi=600 MyriadMM
texfont --we=400 --wi=600 MyriadMM-It
texfont --we=700 --wi=600 MyriadMM-It
```

After this you will have files like:

```
MyriadMM-weight-400-width-600.afm
MyriadMM-weight-400-width-600.pfb
```

These alternatives can now be made  $\TeX$ -ready by saying:

```
texfont --ve=adobe --co=myriad --ma --in MyriadMM-we*
texfont --ve=adobe --co=myriad --ma --in MyriadMM-It-we*
```

Slanted, boldslanted and capitalized variants can be created with:

```
texfont --ve=adobe --co=myriad --sl=* MyriadMM-we*
texfont --ve=adobe --co=myriad --ca=* MyriadMM-we*
```

## Map files

There are several ways to tell PDF $\TeX$  which map files to use. One way is to add an entry to the file `pdftex.cfg`, like:

```
map +texnansi-urw-palatino.map
```

You can also add an entry in the  $\TeX$  file, by saying:

```
\pdfmapfile{+texnansi-urw-palatino.map}
```

or in CON $\TeX$ T with:

```
\loadmapfile[texnansi-urw-palatino.map]
```

---

Currently these are only available for the Unix operating system.

This command prevents duplicate loading of files. If you want CON<sub>T</sub>E<sub>X</sub>T to load the files automatically, you can add an entry to your `cont-sys.tex` file:

```
\autoloadmapfilestrue
```

Beware: currently PDF<sub>T</sub>E<sub>X</sub> only loads map files before the first page is shipped out. If you define fonts halfway the document, you must make sure that the associated map file is loaded at the top of your document. A rather rough way out is to say:

```
\usetypescript[map][texnansi,ec,8r]
```

or whatever combination of encodings you want. If you provide the keyword `all`, CON<sub>T</sub>E<sub>X</sub>T will load all known map files.

## Switches

You can control T<sub>E</sub>X<sub>F</sub>ONT's behaviour with command line switches. You can get an overview of these by saying

```
texfont --help
```

Some switches expect a number or string. You can abbreviate switches as long as you make sure that they can be distinguished.

**fontroot=path** This is the place where the files that are generated will go to. You can either set the font root manually, or let T<sub>E</sub>X<sub>F</sub>ONT consult your path settings.

**sourcepath=path** When you install new fonts, by default the current path is taken. This switch can be used to specify an absolute or relative path. If you provide `auto` as path, T<sub>E</sub>X<sub>F</sub>ONT will try to locate the source path by means of the vendor and collection specification.

**vendor=name** When generating metrics, this key is mandatory. It is used as a directory specifier under the `/fonts` path and ends up as part of the map file name.

**collection=name** Like the `vendor` key, when generating metrics, this key is mandatory. It is used as a directory specifier under the `/vendor` path and ends up as part of the map file name.

**encoding=name** Here you specify the font encoding vector. You need to make sure that you have a correct file on your system (like `ec.enc` or `8r.enc`). The default encoding is `texnansi` but any decent alternative will do.

**slant=number** The number specified here is normally not that large. An often used value is `.167` which is also the default. If you provide a `*`, the default value is used. Values between 0.0 and 1.5 are accepted. The number, multiplied by 1000 and rounded ends up in the name of the font instance.

**extend=number** This specifier determines how much a glyph will be stretched in horizontal direction. The default value is 1.2. See also `slant`.

**caps=number** This specifier determines how much lowercase glyphs will be scaled in vertical direction. The default value is 0.8. See also `slant`.

**weight=number** When generating a multiple master instance, this number determines how bold a glyph will look. Values between 200 and 700 give acceptable results. The exact specification depends on the font. The number becomes part of the filename.

**width=number** This number defines how wide a glyph will be. See also `weight`.

**install** This switch instructs `TEXFONT` to copy files from the source path into the right locations of the font tree.

**makepath** If you provide this switch, `TEXFONT` will create the paths needed for installing the fonts. Otherwise, you have to do so yourself, otherwise `TEXFONT` will quit with a warning.

**listing** Given that your sourcepath is set to `auto`, this switch will result in a list of the metric files of the installed fonts.

**remove** Given that your sourcepath is set to `auto`, this switch will result in deletion of the metric files of the installed fonts.

**test** If you are just playing a bit with `TEXFONT`, you may not want to clobber your font path with files that you will never use. This switch sets the vendor and collection equal to `test`.

**show** If you want, `TEXFONT` can process the test file it generates during the installation.

**batch** In the next section we will discuss batch files. This switch instructs `TEXFONT` to treat the filename given as a batch file.

If you peek into the source of `TEXFONT`, you will notice a couple of more switches. These are not documented here, and may disappear in future versions.

**virtual** Create a virtual font (`vf` and `tfm` file) instead of a normal one (`tfm` only).

## Batch files

Once a decent conversion is sorted out, keying in commands like the ones mentioned in previous sections will become an annoyance. This is why T<sub>E</sub>XFONT supports a crude but effective way of processing batch files. One of the batch files that comes with T<sub>E</sub>XFONT, `type-tmf.dat`, has entries like:

```
--en=? --ve=urw --co=palatino --so=auto
--en=? --ve=urw --co=palatino --so=auto --ca=* uplr8a
--en=? --ve=urw --co=palatino --so=auto --sl=* uplr8a
--en=? --ve=urw --co=palatino --so=auto --sl=* uplb8a
```

The `?` will be replaced by the encoding passed to T<sub>E</sub>XFONT when processing this batch file:

```
texfont --encoding=8r --batch type-tmf.dat
```

If you have T<sub>E</sub>X Live installed, you can play with this file and the examples shown in this batch file, since it only uses the free fonts that are on the system. If you run the whole file, for which you can best set up a font tree first, you get a nice collection of fonts. This way you can build your own (stable) font tree. Another batch file is `type-buy.dat`, a file used at PRAGMA ADE. This file has a companion `type-buy.tex` file that contains typescripts. You need to load this file explicitly in order to get access to these scripts.

```
\usetypescriptfile[type-buy]
```

If you run a batch file with `vendor` and `collection` keys, only the section marked as such will be processed. A section is marked with:<sup>3</sup>

```
# urw palatino
```

If you make batch files (or sections) for each font collection that you install, you can always regenerate the files quickly. That way you can occasionally clean up your disk and make a fresh start.

On the next pages we will show the public Zapf Chancery font in some rather common encodings. Watch out: the current release of PDF<sub>T</sub>E<sub>X</sub> only reads map files that are defined before the first page of the document is shipped out.

---

Instead of `#`, a `%` may be used.

```
texfont --encoding=texnansi --vendor=urw --collection=zapfchan
```

```
\loadmapfile[texnansi-urw-zapfchan]
```

```
\definefontsynonym[ZapfChancery][texnansi-uzcmi8a][encoding=texnansi]
```

```
\showfont[ZapfChancery] \showligatures[ZapfChancery]
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																
000	00	001	01	002	02	003	03	004	04	005	05	006	06	007	07	010	08	011	09	012	0a	013	0b	014	0c	015	0d	016	0e	017	0f
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																
020	10	021	11	022	12	023	13	024	14	025	15	026	16	027	17	030	18	031	19	032	1a	033	1b	034	1c	035	1d	036	1e	037	1f
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47																
040	20	041	21	042	22	043	23	044	24	045	25	046	26	047	27	050	28	051	29	052	2a	053	2b	054	2c	055	2d	056	2e	057	2f
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63																
060	30	061	31	062	32	063	33	064	34	065	35	066	36	067	37	070	38	071	39	072	3a	073	3b	074	3c	075	3d	076	3e	077	3f
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79																
100	40	101	41	102	42	103	43	104	44	105	45	106	46	107	47	110	48	111	49	112	4a	113	4b	114	4c	115	4d	116	4e	117	4f
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95																
120	50	121	51	122	52	123	53	124	54	125	55	126	56	127	57	130	58	131	59	132	5a	133	5b	134	5c	135	5d	136	5e	137	5f
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111																
140	60	141	61	142	62	143	63	144	64	145	65	146	66	147	67	150	68	151	69	152	6a	153	6b	154	6c	155	6d	156	6e	157	6f
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127																
160	70	161	71	162	72	163	73	164	74	165	75	166	76	167	77	170	78	171	79	172	7a	173	7b	174	7c	175	7d	176	7e	177	7f
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143																
200	80	201	81	202	82	203	83	204	84	205	85	206	86	207	87	210	88	211	89	212	8a	213	8b	214	8c	215	8d	216	8e	217	8f
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159																
220	90	221	91	222	92	223	93	224	94	225	95	226	96	227	97	230	98	231	99	232	9a	233	9b	234	9c	235	9d	236	9e	237	9f
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175																
240	a0	241	a1	242	a2	243	a3	244	a4	245	a5	246	a6	247	a7	250	a8	251	a9	252	aa	253	ab	254	ac	255	ad	256	ae	257	af
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191																
260	b0	261	b1	262	b2	263	b3	264	b4	265	b5	266	b6	267	b7	270	b8	271	b9	272	ba	273	bb	274	bc	275	bd	276	be	277	bf
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207																
300	c0	301	c1	302	c2	303	c3	304	c4	305	c5	306	c6	307	c7	310	c8	311	c9	312	ca	313	cb	314	cc	315	cd	316	ce	317	cf
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223																
320	d0	321	d1	322	d2	323	d3	324	d4	325	d5	326	d6	327	d7	330	d8	331	d9	332	da	333	db	334	dc	335	dd	336	de	337	df
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239																
340	e0	341	e1	342	e2	343	e3	344	e4	345	e5	346	e6	347	e7	350	e8	351	e9	352	ea	353	eb	354	ec	355	ed	356	ee	357	ef
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255																
360	f0	361	f1	362	f2	363	f3	364	f4	365	f5	366	f6	367	f7	370	f8	371	f9	372	fa	373	fb	374	fc	375	fd	376	fe	377	ff

name: *texnansi-uzcmi8a* at 11.0pt encoding: *texnansi* mapping: *texnansi* handling: default

```
ff fi fl ffi ffl ‘ ’ -- ---
```

```
H H H HH HH ¨ ¨ = =
```

```
texfont --encoding=8r --vendor=urw --collection=zapfchan
```

```
\loadmapfile[8r-urw-zapfchan]
```

```
\definefontsynonym[ZapfChancery][8r-uzcmi8a][encoding=8r]
```

```
\showfont[ZapfChancery] \showligatures[ZapfChancery]
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	00	001	01	002	02	003	03	004	04	005	05	006	06	007	07
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
020	10	021	11	022	12	023	13	024	14	025	15	026	16	027	17
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
040	20	041	21	042	22	043	23	044	24	045	25	046	26	047	27
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
060	30	061	31	062	32	063	33	064	34	065	35	066	36	067	37
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
100	40	101	41	102	42	103	43	104	44	105	45	106	46	107	47
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
120	50	121	51	122	52	123	53	124	54	125	55	126	56	127	57
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
140	60	141	61	142	62	143	63	144	64	145	65	146	66	147	67
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
160	70	161	71	162	72	163	73	164	74	165	75	166	76	167	77
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
200	80	201	81	202	82	203	83	204	84	205	85	206	86	207	87
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
220	90	221	91	222	92	223	93	224	94	225	95	226	96	227	97
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
240	a0	241	a1	242	a2	243	a3	244	a4	245	a5	246	a6	247	a7
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
260	b0	261	b1	262	b2	263	b3	264	b4	265	b5	266	b6	267	b7
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
300	c0	301	c1	302	c2	303	c3	304	c4	305	c5	306	c6	307	c7
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
320	d0	321	d1	322	d2	323	d3	324	d4	325	d5	326	d6	327	d7
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
340	e0	341	e1	342	e2	343	e3	344	e4	345	e5	346	e6	347	e7
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
360	f0	361	f1	362	f2	363	f3	364	f4	365	f5	366	f6	367	f7
370	f8	371	f9	372	fa	373	fb	374	fc	375	fd	376	fe	377	ff

name: 8r-uzcmi8a at 11.0pt encoding: 8r mapping: 8r handling: default

ff fi fl ffi ffl ‘ ’ -- ---

*ff fi fl ffi ffl* ‘ ’ -- ---

```
texfont --encoding=ec --vendor=urw --collection=zapfchan
```

```
\loadmapfile[ec-urw-zapfchan]
```

```
\definefontsynonym[ZapfChancery][ec-uzcmi8a][encoding=ec]
```

```
\showfont[ZapfChancery] \showligatures[ZapfChancery]
```

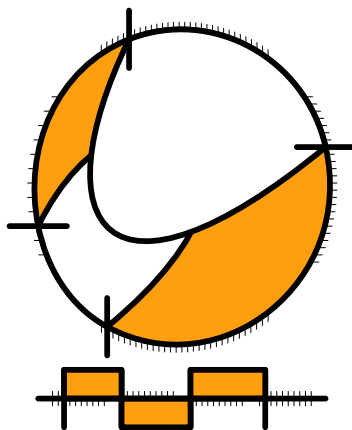
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
000	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
020	021	022	023	024	025	026	027	028	029	030	031	032	033	034	035
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
040	041	042	043	044	045	046	047	048	049	050	051	052	053	054	055
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
060	061	062	063	064	065	066	067	068	069	070	071	072	073	074	075
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	fa	fb	fc	fd	fe	ff

name: ec-uzcmi8a at 11.0pt encoding: ec mapping: ec handling: default

ff fi fl ffi ffl ‘ ’ -- ---

ff fi fl ffi ffl ‘ ’ -- ---





**PRAGMA**

Advanced Document Engineering | Ridderstraat 27 | 8061GH Hasselt NL  
tel: +31 (0)38 477 53 69 | email: [pragma@wxs.nl](mailto:pragma@wxs.nl) | internet: [www.pragma-ade.com](http://www.pragma-ade.com)