



latex-test

2022-03-22

Ken Moffat

1 Introduction, or my history with \LaTeX

A frustrating but sometimes educational experience.

It is easy to forget that $T_{\text{E}}X$ is at heart an old-school programming language, with a lot of additional macros added over the years, and many different options. Like all programming languages, it takes a long time to achieve any level of competence.

One day in February 2014, somebody noticed that our Beyond Linux From Scratch (BLFS) build of texlive did not build all of the package (and so, anybody who began by installing the binary *install-tl-unx*, which was our practice, still had programs which were not built from source).

I had no experience of this [*insert profanities here*] typesetting system, and my initial attempts to try to use it found many examples which perhaps worked when they were posted, but did not work for me. Eventually, I found a few routines which gave me a little confidence that some of it worked.

Getting a working version of xindy to build was *fun*.

Eventually I came back to this, got more of it working, and eventually got it all working from source (although on one of my machines the *binary* version of ConTeXt failed - that CPU did not support some SSE options that the contributed binary used, but such is life and anyway we prefer to build from source!). The tests are now here to check that a new version works.

Along the way, I have discovered that I really dislike much of $T_{\text{E}}X$ itself, and \LaTeX too is a bit problematic:

- The Fonts are ugly. I have no interest in mathematical formulae, only in humanities. For this document I am using X \LaTeX so that I can use my system fonts. In particular the original fonts, and variants such as cm-unicode, sometimes make very strange choices for diacritical markings, particularly the cedilla.

- It is a programming language, but very idiosyncratic and sometimes poorly documented – some parts are well documented, others less so, and the names you need to search for are not always the regular typesetting names.¹
- All the additions in \LaTeX etc. use macros, and many want to be loaded last. In practice, that means that they interact with each other.
- I have no use for outputs other than PDF files.
- As with many things from the past, basic \TeX predates unicode. There have been various language-specific versions in the past, but for me the least-painful \TeX variant is xelatex.

However, as it pulls me further in I realise that for some things it is a tolerable system. But I hope never to reach the stage of loving it.

2 About this package

I aim to test **basic** modern PDF creation in latex and its derivatives (lualatex, xelatex) using texlive. As a convenience to myself I also *minimally* test context.

These tests assume that you installed a *full* texlive system, and ideally also a few useful OTF or TTF fonts (in practice, only producing Korean text *requires* an additional font beyond what a full texlive installs.

These tests will not run, at all, on a system using tetex. I looked at the most recent Slackware, but although that has pdflatex it does not have all of the files that my test uses, particularly the stylesheet for *lorem ipsum*.

The configure script is there to check that everything it requires is present, and then to check for fonts. The package does not install anything. If some texlive programs, or some fonts, or some latex packages, are missing it will reduce the tests in the Makefile.

¹<http://mintaka.sdsu.edu/GF/bibliog/latex/gripe.html>

You can then run ‘make’ to run the tests. All being well, each will produce a PDF for you to examine in your preferred viewer. Alternatively, you can run the tests individually. Running multiple tests in parallel has NOT been regularly tested.

Many of the examples are intended to be mildly humorous, and although the files might be useful as examples of how to use some things, they should not be taken as “good” examples, apart from those which I have copied wholesale!

3 What the various tests do

3.1 biblatex-biber

A bibliography using biblatex with the biber backend. *Execs: pdflatex, biber Uses: biblatex, combelow, ragged2e*

This uses “traditional” T_EX markup for diacriticals and some “extra” latin letters such as Æ. Although I prefer to use UTF-8, apparently some people still do things in ASCII.

3.2 biblatex-bibtex

A bibliography using biblatex with the bibtex backend. *Execs: pdflatex, bibtex Uses: biblatex, combelow, ragged2e*

Essentially the same as *biblatex-biber*.

3.3 biblatex-unibiber

Biblatex and biber are a modern way of providing bibliographies and here they are used with UTF-8. *Execs: pdflatex, biber Uses: biblatex, bookmark, csquotes, inputenc, fontenc, savetrees*

This version uses biber and biblatex with pdflatex. It is “limited” in the UTF-8 it will accept for bibliography keys, although plain (latin-alphabet) text seems fine.

3.4 **bibtexidx**

Bibtex and makeindex used with ASCII input. *Execs: pdflatex, bibtex, makeindex* *Uses: makeidx, natbib, setspace, url*

Having gone to the trouble of creating a document to use bibtex in the old-fashioned way, I decided to go the extra mile and use makeindex on it.

3.5 **bibxelatex**

Bibxelatex tests biber with xelatex as well as using xeCJK. *Execs: xelatex, biber* *Uses: babel, biblatex, csquotes, fontspec, xeCJK*

This began as an alternative way of testing a bibliography. Because it is using xelatex, it has no problem with using a cyrillic key to a bib entry, nor any accented latin. It also contains the bibliography within the tex file, using the filecontents package to write it to a bib file.

Along the way I came across the xeCJK package, with its ability to separate blocks of Unified Han unicode glyphs - in the example I copied, Kana (Japanese) and Hangul (Korean) were separated. This seemed like a good idea, but further experimentation showed that not all Japanese is Kana - the ideograms shared with Chinese are known as Kanji. Chinese fonts seem to contain some of these, but coverage varies from one font to another. It is possible to use a fallback font, which should mean that everything is rendered in some form, but that either means using a Chinese font with a Japanese fallback (some things might look odd to a Japanese reader), or the converse with some things looking odd to a reader of Simplified Chinese.

In practice, I do not expect anybody to need a real bibliography with multiple languages in it. The approach I took in the lualatex file (separate fonts, selected when necessary) is probably a better approach. For this test I have a new configure switch, *-prefer-* with values of *japanese* (the default) or *chinese*.

There is no option to disable this test. As well as xelatex, it requires a general OTF/TTF font and *both* Simplified Chinese and Japanese fonts. It can also use a korean font if one is found.

3.6 bidipoem

Bidi is for bidirectional text (LTR and RTL). *Execs: xelatex Uses: bidi, bidipoem, fontspec*

This uses XeLaTeX to typeset a piece of Omar Khayyam's poetry in both English and Persian. Uses an OTF/TTF font which will handle both languages, and uses the bidi and bidipoem packages. Also uses double backslashes (insert a newline) for English poetry (I do the same thing in ABOUT.tex to stop lines in certain languages from *excessively* overflowing).

3.7 blended-circles

A test using tikz to produce a colourful picture. *Execs: pdflatex Uses: pgfbaseimage*

These are blended circles (i.e. the main RGB circles change colour where they overlap). The blend option within Tikz was fairly new when I created this. It should work in a 2014 or later texlive install.

3.8 context

ConTeXt is apparently used for desktop publishing. *Execs: context*

It is quite different in its operation from other T_EX variants - in particular, the messages from running it. It also uses different fonts from traditional T_EX and L^AT_EX.

This uses ConTeXt MkIV, so it uses lua.

3.9 cylinder

This test uses `asy` for a 3-Dimensional drawing. *Execs: asy, gs, pdflatex Uses: asymptote*

Draws a cylinder. This requires `asymptote` to have been compiled with `freelut`, I do not know of a reliable way to test for that (so if it has not been compiled like that you will need to disable this test). While this is running, it draws on the screen (typically frames, sometimes junk), and then removes this at the end - I regard that as a feature of `asy`, but it is disconcerting when you first encounter it. This test is slow, as is the rendering of the resulting PDF.

3.10 ligatures

The `LuaHBTeX` backend of `LuaTeX` with `LaTeX`. *Execs: lualatex Uses: fontspec, luacode and requires 'Linux Libertine O' font*

The `HarfBuzz` backend of `Luatex` can access features of OTF fonts, in this case ligatures (e.g. modified forms of 'fi' and 'fl'). This font has a rich set of ligatures for latin languages.

3.11 lualatex

`LuaTeX` with `LaTeX`. *Execs: lualatex Uses: fontspec, luacode, met-
alogo, microtype, savetrees, setspace*

`Lualatex` is able to cope with `fontspec` and UTF-8. I discovered that it uses `luaotfload-tool` to find *system* fonts, and runs that when it feels like it! So, my configure script runs that command to ensure that any OTF or TTF font which I have only just made known to `fontconfig` will be found.

The `lualatex` test itself provides a greeting in various languages using the latin, cyrillic, and greek alphabets. It will also do this for Simplified Chinese, Japanese, and Korean if it finds the fonts (`texlive` itself does not seem to provide a Korean OTF/TTF font, but it does provide Simplified Chinese and Japanese). It also generates a random number, because `lua` is a scripting language with many other uses - some people think it might be the future of \TeX .

3.12 mosfet

Another tikz test, this time drawing an electronic circuit diagram.
Execs: pdflatex Uses: circuitikz, siunitx, verbatim

A conventional use for Tikz, but it requires two further packages which are not always installed. This gave me a reason to use kpswhich to check for the stylesheets.

3.13 paren

A test to see if ruby works. *Execs: match_parens, pdflatex*

This test checks if ruby is working, by running match_parens which is a script supplied with texlive. Ruby has not always been installed on my systems, and many of the ruby scripts supplied by texlive did not get used by my test files.

But match_parens is useful (it does what the name suggests - check that parentheses or 'curly brackets' are matched) and it also fails (as noted in the script) when run on itself. So, the test checks for the failure, and for success on the tex source which it will then use to produce a PDF to show all is well.

3.14 pdflatex

The most basic part of the tests. *Execs: pdflatex Uses: fancyhdr, geometry, lipsum, setspace, xcolor*

This uses pdflatex to check sizing, left / centred / right alignments, produce an example formatted equation, and generate some of the usual lorem ipsum cod latin. It also uses colours.

This just checks that a PDF with some *lorem ipsum* cod latin, some maths and some colours can be produced.

3.15 triangles

This test uses asy to make a 2-Dimensional drawing. *Execs: asy, gs, pdflatex Uses: asymptote*

Draws coloured triangles.

3.16 xelatex

Xelatex is for UTF-8 and system OTF/TTF fonts. *Execs: xelatex Uses: fontspec, setspace*

This is another way of using UTF-8 and system OTF/TTF fonts to produce a PDF. This test uses separate fonts for latin / cyrillic / greek, and (if found) Simplified Chinese, Japanese, and Korean. See comment for lualatex re fonts.

3.17 xindy

Xindy, a modern way of indexing \TeX documents. *Execs: lualatex, makeglossaries Uses: bookmark, fontspec, glossaries, hyperref, imakeidx, idxlayout, setspace*

Because xindy is not built as part of the default texlive source build (it requires an existing latex and pdflatex to build it), I needed to prove that my build worked - in the end I managed to create an early version of this test which worked on the binary, but when I ran it on the self-built version it completed apparently normally, but without an index (that is, it failed).

Xindy was my motivation for creating these test files. I eventually found out what was wrong with the build, and now the test uses pdftotext to check that the correct number of pages was produced. It uses lualatex (did I mention that I like UTF-8 ?).

I have now added the bookmark package, with hyperref, to provide links to the various parts. This works in epdfview, but not in my current version of evince which can already list the contents in its own ways. As a side effect, the index links are now clickable (and that *does* work in evince). Of other viewers, mupdf does not provide a left panel for links (nor does it show where you are in the document), and for my purposes okular is similar to evince. I have not yet tested xpdf.

3.18 xindynonlua

This is intended to be a more-widely applicable way of testing xindy. *Execs: pdflatex, texindy Uses: bookmark, glossaries, hyperref, imakeidx, idxlayout, setspace, tipa*

I discovered that while xindy worked fine on my x86_64 builds, on i686 it did not work. This test is offered in the hope that it might be useful (no UTF-8, no luatex, so fewer dependencies). It is a cut-down version of the xindy test, using traditional T_EX markup. I also dropped the references to Æsir and Þórr because they were now indexed under E and O instead of A and after Z, which may have been because I used incorrect markup. More to the point, the Makefile invokes xindy directly instead of using makeglossaries. This makes it easier to edit the Makefile to add debug switches. After a lot of head-banging, I found the problem hidden among the error messages - xindy has become sensitive to how it is built, perhaps as a result of changes to coreutils.

Please note that I have long stopped building or testing anything on i686.

3.19 Tests not run by default

When I was trying to make sense of the maze of perl modules we referenced in BLFS-8.3, I discovered that if we changed our location of ca-certificates to one known to perl core module HTTP::Tidy, we could drop Mozilla::CA from the dependencies of biber: it is only a fallback intended for CPAN installs.

But I then needed to find a testcase. I eventually discovered online a couple of MWEs to use with biber when accessing remote bibliographies. For the moment these still work, but no guarantees that the links will remain. I would prefer to create my own data, but I lacked an https:// site where I can upload. These have now been altered to live on the LFS website.

3.19.1 remote1

Access to a remote bibliography which happens to be in latin1 encoding. *Execs: pdflatex, biber*

The first example I found is described as latin1 when downloaded. That needs an extra switch in biber.

3.19.2 remote2

Access to a remote bibliography in UTF-8 encoding. *Execs: pdflatex, biber*

This example downloads as UTF-8 so it is straightforward for biber to use it.

3.20 dvisvgm

I had added dvisvgm to BLFS's packages for use with asymptote. This test just proves that dvisvgm can use tikz to create an svg image. It has not been wired into the main tests. *Execs: latex, dvisvgm*

4 What is NOT tested

To be honest, *most* of texlive.

Even if I test something, it is only with the options in my testsuite. If you use texlive for real, you need to maintain your own .tex files to be able to confirm that a new version has no regressions for you.

One of the things I do not normally test is monospaced fonts (although, arguably, CJK fonts are monospaced). Here is a listing of a program which used to get referenced a lot in my JCL when I was an application programmer, IEFBR14. Taken from <http://en.wikipedia.org/wiki/IEFBR14> which is as good an explanation as any of why things need to be tested. Thanks to John Pershing for documenting this.

```
IEFBR14 START
        USING IEFBR14,15  Establish addressability
        B      GO          Skip over our name
        DC     AL1(L'ID)   Length of name
ID      DC     C'IEFBR14' Name itself
        DS     0H          Force alignment
GO      SR     15,15       Zero out register 15
        BR     14          Branch to return addr in R14
        END    IEFBR14
```

That is about the only use I ever expect to have for monospaced fonts in texlive. Getting it to render sanely was unpleasant. I suppose that Knuth expects monospaced text to only be used for short commands.

5 dependencies

5.1 Required system dependencies

The configure script is targetted at BLFS, but I have tried to make it system-agnostic. All of the following are required, and tested in this order:

bash : I use 'type -pa' to identify programs.

Once I started to test on FreeBSD-10.1 I was going to change this, but /bin/sh there appears to be a csh variant and code which supposedly works in POSIX sh did not work. But after I had installed Xorg, a desktop, and texlive, I found that bash was present at /usr/local/bin/bash. So, bash is required.

cat : a basic tool. In practice, if this is missing then configure will probably give a messy error as soon as it starts, before running the tests to check for the required programs.

find : a basic tool, used to look for the T_EXfonts

grep : a basic tool.

make : a basic tool.

In the early days I tested FreeBSD in a VM, using xfce. The version of 'make' which was present worked, but since then I have added new tests and I might have made this incompatible with that and related OS's.

rm : a basic tool.

sed : for portability, I do not use in-place edits.

X : (i.e. Xorg) : without a GUI, how can you run a PDF viewer ? When this started, Xorg was the normal way of getting a 'nix GUI. I do not have any systems running Wayland, and there does not appear to be a simple and consistent way of determining that something is running under Wayland, so for the moment the lack of X is treated as an error.

kpsewhich : I now use this to check that the packages used by each test have been installed (except for context, which I find impenetrable).

pdflatex : to me, this is the most basic part of texlive. I make it required.

5.2 Optional dependencies

These are used by various tests. If they are not present, the test will not be enabled in the Makefile. With the exceptions of *ruby*, *fc-list*, *gs*, *tr* (a basic system program) and the programs from *poppler* they are all part of a **full** texlive install.

asy : for the *cylinder* and *triangles* tests.

biber : for the *biblatex-biber*, *biblatex-unibiber* and *bibxetex* tests.

bibtex : for the *biblatex-bibtex* and *bibtexidx* tests.

context : this tests Mk IV *context*. Unfortunately, there seem to have been a number of broken binary versions of this in 2014, so I test *context -version* to see if that errors, and disable the test if it does.

fc-list : required to help find system OTF or TTF fonts if either *lualatex* or *xetex* (or tests using them) will be run. Its absence will mean that system fonts cannot be used, but I think I disable all the users if that happens, to keep the logic in *configure* somewhat manageable. In practice, it is a dependency of Xorg so it should always be present.

gs : for the *asy* tests.

kpsepath : used to look for texlive OTF/TTF fonts if either *lualatex* or *xetex* (or tests using them) will be run.

lualatex : for *ligatures*, *lualatex* and *xindy*.

luaotfload-tool : used, if present, to ensure the *lualatex* test will find system fonts when it runs. Required for *ligatures*.

makeglossaries : for *xindy*.

makeindex : for *bibtexidx*.

match_parens : for the *parens* test.

pdftinfo : (from poppler) is needed to check that the *xindy* test worked, it is unfortunately common for a broken build of *xindy* to let the test complete without error but not produce the index page.

pdftotext : (from poppler) is needed to check that the *biber* tests worked.

ruby : for the *parens* test.

TikZ : mentioned here although it is actually just a collection of packages (see the next subsection for more on packages).

tr : used to break up the output from *kpsepath* to find where the directory for *texlive* fonts should be. As with *fc-list*, if it is missing then tests using OTF and TTF fonts will be deselected.

xelatex : for the *xelatex* and *bidipoem* tests.

xindy : for the *xindy* and *xindynonlua* tests.

5.3 Required L^AT_EX packages

I eventually realised that I can, and should, use *kpsewhich* to confirm that the packages used by each of my T_EX scripts are present. After *configure* has checked for programs, it now checks for **all** of the stylesheets which I use. It then determines which tests cannot be run because a package is not installed. If an installed version is *broken*, running the tests may still drop you to a T_EX commandline.

5.4 Optional OTF/TTF font dependencies

For those tests which use fontpec (all tests using lualatex or xelatex), configure will look for usable fonts. To find texlive fonts it originally used *mkluatexfontdb* because that exists on my x86_64 installs where it updates a cache in \$HOME. Unfortunately, that command does not exist on i386 binary versions of texlive. Now I use *kpsepath* and *tr* to find the *textmf-dist/fonts/* directory, and then run *find* there.

I also use *fc-list*. Here, I list the order in which configure searches. I assume that anybody using this will have at least decent serif fonts for Latin, Cyrillic, and modern Greek. For CJK, the order of searching assumes that serif fonts are preferred to sans, (for Chinese, Kai script fonts come between Serif and Sans), and for Japanese and Korean it will use appropriate pan-CJK fonts (technically, those are Chinese) as a fallback in order to try to provide the maximum coverage. Newer, maintained, fonts are usually preferred to older unmaintained. You can alternatively specify system fonts using configure switches to override the process. For system (fontconfig/freetype) fonts I am listing the file name in case anybody wants to track these down for their own use.

The output from configure shows the name of the font it has chosen: For texlive fonts this is the filename, for system fonts it is the name by which fontconfig knows it.

For a main font for Latin, Greek, Cyrillic alphabets:

- Linux Libertine O (texlive *LinLibertine_R.otf*)
- DejaVu Serif (texlive or system, *DejaVuSerif.ttf*)
- FreeSerif (texlive or system, *FreeSerif.otf*)

For a Simplified Chinese font:

- FandolSong (texlive *FandolSong-Regular.otf*)
- AR PL UMing CN (system, *uming.ttc*)
- AR PL Sungtil GB (system, *gbsn00lp.ttf*)
- AR PL New Sung (system, *odosung.ttc*)
- AR PL UKai CN (system, *ukai.ttc*)
- AR PL KaitiM GB (system, *gkai00mp.ttf*)
- AR PL New Kai (system, *odokai.ttf*)
- FandolHei (texlive *FandolHei-Regular.otf*)
- Noto Sans CJK SC (system, *NotoSansCJKsc-Regular.otf*)
- Odo Hei (system, *odohei.ttf*)
- WenQuanYi Zen Hei (system, *wqy-zenhei.ttc*)
- Droid Sans Fallback (system, *DroidSansFallbackFull.ttf*)

For a Japanese font:

- IPAexMincho (texlive or system, *ipaexm.ttf*)
- IPAMincho (texlive or system, *ipam.ttf*)
- HanaMinA (system, *HanaMinA.ttf*)
- Kochi Mincho (system, *kochi-mincho-subst.ttf*)
- IPAexGothic (texlive or system, *ipaexg.ttf*)
- IPAGothic (texlive or system, *ipag.ttf*)
- VL Gothic (system, *VL-Gothic-Regular.ttf*)
- Noto Sans CJK JP (system, *NotoSansCJKjp-Regular.otf*)
- Kochi Gothic (system, *kochi-gothic-subst.ttf*)
- Sawarabi Gothic (system, *sawarabi-gothic-medium.ttf*)
- WenQuanYi Zen Hei (system, *wqy-zenhei.ttc*)
- Droid Sans Fallback (system, *DroidSansFallbackFull.ttf*)

Note: in texlive 2015, although Sawarabi Mincho worked in xelatex, lualatex was unable to find it. Therefore I have not included it in the tests.

For a Korean font:

- Baekmuk Batang (system, *batang.ttf*)
- NanumMyeongjo (system, *NanumMyeongjo-Regular.ttf*)
- UnBatang (system, *UnBatang.ttf*)
- Baekmuk Dotum (system, *dotum.ttf*)
- Baekmuk Gulim (system, *gulim.ttf*)
- NanumGothic (system, *NanumGothic.ttf*)
- UnDotum (system, *UnDotum.ttf*)
- UnGraphic (system, *UnGraphic.ttf*)
- Noto Sans CJK KR (system, *NotoSansCJKkr-Regular.otf*)
- Noto Sans CJK JP (system, *NotoSansCJKjp-Regular.otf*)
- Noto Sans CJK SC (system, *NotoSansCJKdc-Regular.otf*)
- Noto Sans CJK TC (system, *NotoSansCJKtc-Regular.otf*) Unusually, all four variants cover korean with identical glyphs.
- WenQuanYi Zen Hei (system, *wqy-zenhei.ttc*)

For use in bidipoem, to produce output in *both* English and Persian:

- Persian Modern (this used to be in texlive when I started these tests)
- FreeSerif (texlive or system, q.v.)
- DejaVu Sans (texlive or system, *DejaVuSans.ttf*)

Note: neither *nazli* nor *FreeFarsi* provide the apostrophe and fancy double-quotes used by the English text in my example.

6 configure options

6.1 reducing what will be tested

At times (e.g. when I know that a binary is broken) I find it useful to restrict which things will be tested. But mostly I have used these to check the logic of my changes to the configure script. Normally, these switches should not be required.

- `-without-asy` : do not test for asy, disables the cylinder and triangles tests.
- `-without-biber` : do not test for biber, disables the biblatex-biber, biblatex-unibiber and bibxelatex tests.
- `-without-bibtex` : do not test for bibtex, disables the biblatex-bibtex and bibtexidx tests.
- `-without-bidipoem` : disable the bidipoem test, do not test for poem font.
- `-without-chinese` : do not test for a simplified chinese font, removes simplified chinese text from the lualatex and xelatex tests and disables the bibxelatex test.
- `-without-context` : do not test for context, disables the context test.
- `-without-cylinder` : do not run the cylinder (3-D asymptote) test - use this if asy exists but has not been linked against freeglut.
- `-without-fonts` : do not test for OTF/TTF fonts, disables the bibxelatex, bidipoem, lualatex, xelatex and xindy tests.
- `-without-japanese` : do not test for a japanese font, removes japanese text from the lualatex and xelatex tests and disables the bibxelatex test.
- `-without-korean` : do not test for a korean font, removes korean text from the lualatex and xelatex tests.
- `-without-ligatures` : do not attempt to run the ligatures test.

- `-without-lualatex` : do not test for lualatex, disables the context, lualatex and xindy tests.
- `-without-makeindex` : do not test for makeindex, disables the bibtexidx test.
- `-without-ruby` : do not test for ruby or match_parens, disables the parens test.
- `-without-tikz` : do not test for tikz, disables the circles and mosfet tests.
- `-without-triangles` : do not run the triangles test.
- `-without-xelatex` : do not test for xelatex, disables the bibx-
elatex and xelatex tests.
- `-without-xindy` : do not test for xindy, disables the xindy and
xindynonlua tests.

6.2 specifying OTF/TTF fonts

For the tests which use fontspec (i.e. any of the tests which use lualatex or xelatex), I look for some known fonts. It is possible to override these tests, either to test a font which is normally ignored because a “better” font has been found amongst the texlive fonts or on the system, or to use a completely different font.

If configure cannot find suitable fonts, it will deselect some, or all, of the test. Thus people who do not have any CJK fonts can nevertheless see *some* results from the lualatex and xelatex tests if they have a *suitable* font for the latin, cyrillic, and greek alphabets.

To you use any of these switches, the font **must** be known to fontconfig. You *must* use the (full) filename, e.g. Sillyfont-Regular.otf, because quoting and grepping the output from fc-list give issues when there are spaces in the name known to fontconfig. For fonts which do *not* have spaces in their names known to fontconfig you may be able to just specify that name, e.g. 'Arimo'. However, the font needs to be able to produce a normal style (usually described as Regular, but sometimes regular, and I also allow Book, Medium and Light).

- `-with-main-font="SomeFont.ttf"` : specify the font to use for text in the Latin, Greek, Cyrillic alphabets (not used for the bidipoem test). This font needs to have good coverage - as well as the DejaVu and Free Sans/Serif fonts, Liberation fonts also work.
- `-with-chinese-font="SomeFont.ttf"` : my Chinese text is all in Simplified Chinese, so a Traditional font is probably not a good idea. N.B. I take special action if you specify UKai or Uming : these both provide ttc files (TrueType Collection) containing multiple variants. All of my chinese text is simplified, so for these I force the CN version to ensure that one of the Taiwanese variants is not used.
- `-with-japanese-font="SomeFont.ttf"` : use this for Japanese text - I assume that my examples may include both Kana and Kanji, so most Chinese fonts are unlikely to include all the required codepoints, quite apart from whether or not they look "strange" to someone who reads Japanese.
- `-with-korean-font="SomeFont.ttf"` : Modern Korean hardly ever uses Unified Han ideograms, so a separate Korean font is often required to show Korean text. A full texlive install does not include a Korean OTF/TTF font.
- `-with-poem-font="SomeFont.ttf"` : Use this font for bidipoem which contains English and Persian text. Experiments with inadequate fonts indicate that one or other language may be omitted, and some fonts show the English text with empty boxes where the Persian should be, or vice versa.
- `-prefer-japanese` or `-prefer-chinese` : This can be used for `bibxelatex`, to determine which language is treated as the main one, the other will be used as the fallback. Everything in the preferred language should look reasonable to a reader, anything in the other language might appear in a mixture of Chinese and Japanese style. However, if a pan-CJK font was selected, everything in Kanji will probably look Chinese.

7 Examples of what different fonts can do

People who know me will recall that I am interested in being able to render as much as possible of what I encounter on the net, even if I cannot read it. I am also interested in the different additions to the latin and cyrillic alphabets. The following examples are, if my luck has held, the first article of the Universal Declaration Of Human Rights, sometimes using different fonts to get the coverage.

For some of these languages I am forcing new lines in the text (with double backslashes at the end of the line) to get it to fit within the normal printable width of the page. I am sure that there are better ways, at least for *some* of these languages. Except where noted, these are all in the DejaVu Sans font.

For attribution, see the References. Most were found at <http://www.ohchr.org/EN/UDHR/Pages/Introduction.aspx>

N.B. for languages such as Latvian, Maltese and Romanian it is so much easier to do this in xelatex with UTF-8 : No additional packages needed.

7.1 Abkhasian [3]

Дарбанзаалак ауаџы дшоуп ихы дақәитҭны. Ауаа зегь зинлеи патулеи еиқароуп. Уртҭ ирымоуп ахшыџи аламыси, дара дарагы аешьеи аешьеи реигҭш еизықазароуп.

7.2 Adja [2]

Agbetɔwo pleṅu vanɔ gbeme ko vovodeka gbeswegbeswe, sɔto amenyinyi ko aɛwo gome; wo xɔnɔ susunywin ko jimenywi so esexwe. Wo ɔo a we nɔvi ɔaɔa wowo nɔnɔwo gbɔ.

7.3 Armenian [9]

Բոլոր մարդիկ ծնվում են ազատ ու հավասար իրենց արժանապատվությամբ ու իրավունքներով: Նրանք ունեն բանականություն ու խիղճ և միմյանց պետք է եղբայրաբար վերաբերվեն:

7.4 Cantonese using Droid Sans Fallback [17]

人人生出嚟就係自由嘅，喺尊嚴同權利上一律平等。佢哋具有理性同良心，而且應該用兄弟間嘅關係嚟互相對待。

7.5 Catalan [15]

Tots els éssers humans neixen lliures i iguals en dignitat i en drets. Són dotats de raó i de consciència, i han de comportar-se frater-nalment els uns amb els altres.

7.6 Chinese (Simplified) using Fandol Hei [9]

人人生而自由,在尊严和权利上一律平等。他们赋有理性和良心,并应以兄弟关系的精神互相对待。

7.7 Chinese (Traditional) using Fandol Hei [18]

人皆生而自由；在尊嚴及權利上均各平等。人各賦有理性良知，應和睦相處，情同手足。

7.8 Croatian [6]

Sva ljudska bića rađaju se slobodna i jednaka u dostojanstvu i pravima. Ona su obdarena razumom i sviješću i treba da jedno prema drugome postupaju u duhu bratstva.

7.9 Franco-Provençal or Arpitan [1]

Tsecoun poué ché prèvâli dè tui lè droué è dè tôte lè libèrtâ proclamâ in sta Déclarachion, chin ócóna distinchion, dè rache, dè cólóou, dè sèxe, dè linvoua, dè relijion, d'opinion poleteca ou bin dè tôte j'âtre j'opinion d'orejena nachionala ou sociala, dè fórtóna, dè nèchance ou bin dè tôte j'âtre setoachion.

7.10 Hungarian [9]

Minden. emberi lény szabadon születik és egyenlő méltósága és joga van. Az emberek, ésszel és lelkiismerettel bírván, egymással szemben testvéri szellemben kell hogy viseltessenek.

7.11 Icelandic [12]

Hver maður er borinn frjálss og jafn öðrum að virðingu og réttindum. Menn eru gæddir vitsmunum og samvizku, og ber þeim að breyta bróðurlega hverjum við annan.

7.12 Japanese using VL Gothic [13]

すべての人間は、生まれながらにして自由であり、かつ、尊厳と権利について平等である。人間は、理性と良心とを授けられており、互いに同胞の精神をもって行動しなければならない。

7.13 Korean using Baekmuk Dotum [11]

모든 인간은 태어날 때부터 자유로우며 그 존엄과 권리에 있어 동등하다. 인간은 천부적으로 이성과 양심을 부여받았으며 서로 형제애의 정신으로 행동하여야 한다.

7.14 Maltese [4]

Il-bnedmin kollha jitwiieldu ħielsa u ugwali fid-dinjità u d-drittijiet. Huma mogħnija bir-raġuni u bil-kuxjenza u għandhom iġibu ruħhom ma' xulxin bi spirtu ta' aħwa.

7.15 Polish [9]

Wszyscy ludzie rodzą się wolni i równi pod względem swej godności i swych praw. Są oni obdarzeni rozumem i sumieniem i powinni postępować wobec innych w duchu braterstwa.

7.16 Portuguese [14]

Todos os seres humanos nascem livres e iguais em dignidade e em direitos. Dotados de razão e de consciência, devem agir uns para com os outros em espírito de fraternidade.

7.17 Romanian [9]

This uses below-commas on s and t, not cedillas

Toate ființele umane se nasc libere și egale în demnitate și în drepturi. Ele înzestrate cu rațiune și conștiință și trebuie să se comporte unii față de altele în spiritul fraternității.

7.18 Russian [9]

Все люди рождаются свободными и равными в своем достоинстве и правах. Они наделены разумом и совестью и должны поступать в отношении друг друга в духе братства.

7.19 Serbian (Cyrillic) [5]

Сва људска бића рађају се слободна и једнака у достојанству и правима. Она су обдарена разумом и свешћу и требаједни према другима да поступају у духу братства.

7.20 Turkish [7]

Bütün insanlar hür, haysiyet ve haklar bakımından eşit doğarlar. Akıl ve vicdana sahiptirler ve birbirlerine karşı kardeşlik zihniyeti ile hareket etmelidirler.

7.21 Ukrainian [8]

Всі люди народжуються вільними і рівними у своїй гідності та правах. Вони наділені розумом і совістю і повинні діяти у відношенні один до одного в дусі братерства.

7.22 Vietnamese [10]

Tất cả mọi người sinh ra đều được tự do và bình đẳng về nhân phẩm và quyền lợi. Mọi con người đều được tạo hóa ban cho lý trí và lương tâm và cần phải đối xử với nhau trong tình anh em.

7.23 Walloon [16]

Tos lès-omes vinèt-st-å monde lîbes, èt so-l'minme pîd po çou qu'ènn'èst d'leu dignité èt d'leus dreûts. I n'sont nin foû rêzon èt-z-ont-i leû consyince po zèls, çou qu'èlzès deût miner a s'kidûre onk' po l'ôte tot come dès frés.

References

- [1] Alphonse Dayer
- [2] Commission béninoise des Droits de l'Homme, Benin
- [3] Office of the High Commissioner for Human Rights - Georgia Field Office
- [4] Maltese University, Malta
- [5] Permanent Mission of the Federal Republic of Yugoslavia to the United Nations Office in Geneva
- [6] Permanent Mission of the Republic of Croatia to the United Nations Office in Geneva
- [7] Secretariat of the Human Rights Coordinating Committee, Turkey
- [8] Ukrainian Union of Jurists, Ukraine
- [9] United Nations Department of Public Information, NY

- [10] United Nations Development Programme, Viet Nam
- [11] United Nations Educational, Scientific and Cultural Organization, Paris
- [12] United Nations Information Centre, Denmark
- [13] United Nations Information Centre, Japan, and the Ministry of Justice and Human Rights NGOs
- [14] United Nations Information Centre, Portugal
- [15] United Nations Information Centre, Spain
- [16] Waremme - Ville de la Paix, Wallonie, Belgique
- [17] <http://lingwiki.com/index.php?title=Cantonese>
- [18] <http://www.omniglot.com/udhr/sinotibetan.htm>

This document is © 2014-2022 Ken Moffat, and licensed under the GNU Free Documentation License v1.3.