

tex-test 2015-01-10

Ken Moffat

1 Introduction, or my history with T_EX

A frustrating but sometimes educational experience.

It is easy to forget that T_EX 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 (BLFS) build of texlive did not build all of the package (and so, anybody who began by installing the binary *install-tl-unx* 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 does 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_EX itself, and L^AT_EX 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 X₃L_AT_EX 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.

One thing I have not yet managed to understand is why the first text within a **section** starts at the left, but all subsequent paragraphs are indented. As with many things, it is easier to just reword the text to (mostly) accomodate that.

2 About this package

I aim to test **basic** modern PDF creation using texlive.

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, are missing it will reduce the tests in the Makefile.

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.

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

Alternatively, you can run the tests individually. Running multiple tests in parallel has NOT been 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 pdflatex

The most basic part of the tests.

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.

3.2 asy

Asymptote is for drawing things.

In this case, a coloured venn diagram. Uses pdflatex and asy.

3.3 biblatex

Biblatex and biber are a modern way of providing bibliographies.

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 bibxelatex

Bibxelatex tests biber with xelatex as well as using 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.

3.5 bidipoem

Bidi is for bidirectional text (LTR and RTL).

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).

3.6 context

Context is apparently used for desktop publishing.

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.

3.7 lualatex

LuaTeX with LaTeX

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 T_EX.

3.8 paren

A test to see if ruby works.

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.

3.9 xelatex

Xelatex is for UTF-8 and system OTF/TTF fonts.

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.

3.10 xindy

Xindy, a modern way of indexing T_EX documents.

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 pdfinfo 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.

3.11 xindynonlua

This is intended to be a more-widely applicable way of testing xindy.

I discovered that while xindy works fine on my x86_64 builds, on i686 it does 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, but in practice these seem to not get passed to texindy, which was still invoked as:

```
runsystem(texindy -M lang/english/utf8-lang
xindyonlua-xindyonlua.idx)
```

4 What is NOT tested

To be honest, *most* of texlive.

Even if I test something, it is only with the options in my test-suite. 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 Beyond Linux From Scratch, but I have tried to make it system-agnostic. All of the following are required, and tested in this order :

`/bin/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 `cs` 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` : basic.

`grep` : similar.

`make` : similar.

`rm` : similar.

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

`touch` : again, basic.

`X` : (i.e. Xorg) : if you don't have that, how will you view the PDFs which this package produces ? Yes, I know some people have PDF readers which will render the text in a `tty`, but that would be a very poor way of looking at \TeX output.

`pdflatex` : to me, this is the most basic part of `texlive`. I make it required because if you do not have it then I assume neither `lualatex` nor `xelatex` were installed, so none of these tests are likely to run.

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 exception of *ruby* they are all part of a **full** `texlive` install.

`asy` : for the `asy` test.

biber : for the *biblatex* test.

lualatex : for *lualatex* and *xindy*.

luaotfload-tool : used to ensure the *lualatex* test will find system fonts when it runs.

makeglossaries : for *xindy*.

xindy : for the *xindy* test.

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

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

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

ruby : for the *parens* test, which tests if ruby is working.

match_parens : for the *parens* test.

fc-list : required to help find system fonts if either *lualatex* or *xelatex* (or tests using them) will be run. Its absence would mean that system fonts cannot be used and therefore trying to specify them will fail. In practice, it is a dependency of Xorg so it should always be present.

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

tr : used to break up the output from *kpsepath* to find where the directory for texlive fonts should be.

5.3 Optional OTF/TTF font dependencies

For those tests which use fontspec (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.

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

- LinLibertine (texlive)
- DejaVu Serif (texlive or system, *DejaVuSerif.ttf*)
- FreeSerif (texlive or system, *FreeSerif.otf*)

For a Simplified Chinese font:

- FandolSong (texlive)
- AR PL UMinG CN (system, *uming.ttc*)
- AR PL SunGtil GB (system, *gbsn00lp.ttf*)
- AR PL New Sung (system, *fireflysung.ttf*)
- AR PL UKai CN (system, *ukai.ttc*)
- AR PL KaitiM GB (system, *gkai00mp.ttf*)
- AR PL New Kai (system, *odokai.ttf*)
- FandolHei (texlive)
- WenQuanYi Zen Hei (system, *wqy-zenhei.ttc*) This plays badly with the English from some Japanese fonts when used in e.g. libreoffice writer.

- Droid Sans Fallback (system, *DroidSansFallbackFull.ttf*)

For a Japanese font:

- IPAexMincho (texlive or system, *ipaexm.ttf*)
- IPAMincho (texlive or system, *ipam.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*)
- Kochi Gothic (system, *kochi-gothic-subst.ttf*)
- WenQuanYi Zen Hei (system, *wqy-zenhei.ttc*)
- Droid Sans Fallback (system, *DroidSansFallbackFull.ttf*)

For a Korean font:

- Baekmuk Batang (system, *batang.ttf*)
- NanumMyeongjo (system, *NanumMyeongjo-Regular.ttf*)
- Baekmuk Dotum (system, *dotum.ttf*)
- Baekmuk Gulim (system, *gulim.ttf*)
- NanumGothic (system, *NanumGothic-Regular.ttf*)
- WenQuanYi Zen Hei (system, *wqy-zenhei.ttc*)

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

- Persian Modern (texlive)
- FreeSerif (texlive or system, q.v.)
- DejaVu Sans (texlive or system, *DejaVuSans.ttf*)

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 asy test.
- `--without-biber` : do not test for biber, disables the biblatex and bibxelatex tests.
- `--without-bidipoem` : disable the bidipoem test, do not test for poem font.
- `--without-context` : do not test for context, disables the context test.
- `--without-lualatex` : do not test for lualatex, disables the context, lualatex and xindy tests.
- `--without-ruby` : do not test for ruby or match_parens, disables the parens test.
- `--without-xelatex` : do not test for xelatex, disables the bibxelatex and xelatex tests.
- `--without-xindy` : do not test for xindy, disables the xindy test.
- `--without-fonts` : do not test for OTF/TTF fonts, disables the bibxelatex, bidipoem, lualatex, xelatex and xindy tests.
- `--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-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.

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, and you **must** use the *name* known to fontconfig, e.g. "DejaVu Sans". Currently (the 2014 release of texlive), trying to use at least the Japanese version of the NotoSans fonts breaks lualatex and causes xelatex to fail, reporting a Freetype error.

- --with-main-font="Some Font" : 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="Some Font" : my Chinese text is all in Simplified Chinese, so a Traditional font is probably not a good idea.
- --with-japanese-font="Some Font" : 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="Some Font" : Modern Korean hardly-ever uses Unified Han ideograms, so a separate Korean font is required to show Korean text. A full texlive install does not appear to include a Korean OTF/TTF font.
- --with-poem-font="Some Font" : Use this font for bidipoem which contains English and Persian text. Experiments with

inadequate fonts indicate that one or other language will 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>

7.1 Abkhasian[3]

Дарбанзаалак ауаџы дшоуп ихы дақәиҭны. Ауаа зегъ зинлеи патулеи еиқароуп. Урҭ ирымоуп ахшыџи аламыси, дара дарагъ аешьеи аешьеи реиҭш еизықазароуп.

7.2 Adja[2]

Agbetɔwo plɛɲu vanɔ gbɛmɛ ko vonoɔka gbeswɛgbeswɛ, sɔto amenyinyi ko acewo gome; wo xɔɔ susunywin ko jimenywi so esexwe. Wo ɔo a wɛ nɔvi ɔaɔa wowo nɔɔwo gbɔ.

7.3 Armenian[8]

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[16]

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[14]

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[8]

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[17]

人皆生而自由；在尊詢及輕利上均各平等。人各囊有理性良知, 嘗義和睦相熾, 情同手足。

7.8 Croatian[5]

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 pou chèn prèvâli dè tui lè drouê 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 bîn dè tôte j'âtre j'opinion d'orejena nachionala ou sociala, dè fórtóna, dè nèchance ou bîn dè tôte j'âtre setoachion.

7.10 Hungarian[8]

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[11]

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[12]

すべての人間は、生まれながらにして自由であり、かつ、尊厳と権利と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 [10]

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

7.14 Polish[8]

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.

7.15 Portuguese[13]

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.

7.16 Romanian[8]

(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.

7.17 Russian[8]

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.

7.18 Serbian (Cyrillic)[4]

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.

7.19 Swedish using Sweden Sans[11]

Alla människor äro födda fria och lika i värde och rättigheter. De äro utrustade med förnuft och samvete och böra handla gentemot varandra i en anda av broderskap.

7.20 Turkish[6]

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

7.21 Ukrainian[7]

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[9]

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[15]

Tos lès-omes vinèt-st-â 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] Permanent Mission of the Federal Republic of Yugoslavia to the United Nations Office in Geneva
- [5] Permanent Mission of the Republic of Croatia to the United Nations Office in Geneva

- [6] Secretariat of the Human Rights Coordinating Committee, Turkey
- [7] Ukrainian Union of Jurists, Ukraine
- [8] United Nations Department of Public Information, NY
- [9] United Nations Development Programme, Viet Nam
- [10] United Nations Educational, Scientific and Cultural Organization, Paris
- [11] United Nations Information Centre, Denmark
- [12] United Nations Information Centre, Japan, and the Ministry of Justice and Human Rights NGOs
- [13] United Nations Information Centre, Portugal
- [14] United Nations Information Centre, Spain
- [15] Waremme - Ville de la Paix, Wallonie, Belgique
- [16] <http://lingwiki.com/index.php?title=Cantonese>
- [17] <http://www.omniglot.com/udhr/sinotibetan.htm>

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