

NAME

ttf2pt1_convert – convenience font conversion script

SYNOPSIS

ttf2pt1_convert [**config-file**]

DESCRIPTION

Convert is the master conversion script provided with ttf2pt1. When installed into a public directory it's named **ttf2pt1_convert** to avoid name collisions with the other programs.

If the configuration file is not specified as an argument then the file `convert.cfg` in the current directory is used. This file contains a set of configuration variables. The distribution contains a sample file `convert.cfg.sample`. Please copy it to `convert.cfg`, look inside it and change the configuration variables. The more stable configuration variables, such as the path names of the scripts and encoding files are located in `convert` itself, they are automatically updated when installing **ttf2pt1**.

Put all the TTF fonts you want to convert into some directory (this may be just the directory that already contains all the Windows fonts on a mounted FAT filesystem). If you have fonts in different source encodings then put the fonts in each of the encodings into a separate directory. Up to 10 source directories are supported. If you (in a rather unlikely case) have more source directories then you can make two separate runs of the converter, converting up to 10 directories at a time.

The variables in the configuration file are:

- **SRCDIRS** – the list of directories (with absolute paths) with TTF fonts. Each line contains at least 3 fields: the name of the directory, the language of the fonts in it (if you have fonts for different languages you have to put them into the separate directories) and the encoding of the fonts. Again, if you have some of the TTF typefaces in one encoding, and some in another (say, CP-1251 and KOI-8), you have to put them into the separate source directories. Some lines may contain 4 fields. Then the fourth field is the name of the external map to convert the Unicode fonts into the desirable encoding. This map is used instead of the built-in map for the specified language.

8 An interesting thing is that some languages have more than one widely used character encodings. For example, the widely used encodings for Russian are IBM CP-866 (MS-DOS and Unix), KOI-8 (Unix and VAX, also the standard Internet encoding), IBM CP-1251 (MS Windows). That's why I have provided the means to generate the converted fonts in more than one encoding. See the file `encodings/README` for details about the encoding tables. Actually, if you plan to use these fonts with Netscape Navigator better use the aliases `cp-866` instead of `ibm-866` and `windows-1251` instead of `ibm-1251` because that's what Netscape wants.
- **DSTDIR** – directory for the resulting Type1 fonts. Be careful! This directory gets completely wiped out before conversion, so don't use any already existing directory for this purpose.
- **DSTENC{language}** – the list of encodings in which the destination fonts will be generated for each language. Each font of that language will be generated in each of the specified encodings. If you don't want any translation, just specify both **SRCENC** and **DSTENC** as `iso8859-1` (or if you want any other encoding specified in the `fonts.dir`, copy the description of `8859-1` with new name and use this new name for **SRCENC** and **DSTENC**).
- **FOUNDRY** – the foundry name to be used in the `fonts.dir` file. I have set it to `fromttf` to avoid name conflicts with any existing font for sure. But this foundry name is not registered in X11 standards and if you want to get the full standard compliance or have a font server that enforces such a compliance, use `'misc'`.

The next few parameters control the general behavior of the converter. They default values are set to something reasonable.

- **CORRECTWIDTH** – if the value is set to **YES** then use the converter option **-w**, otherwise don't use it. See the description of this option in the `README` file.

- **REMOVET1A** – if the value is set to **YES** then after conversion remove the un-encoded `.t1a` font files and the intermediate `.xpf` font metric files.
- **INSTALLFONTMAP** – a Ghostscript parameter, if the value is set to **YES** then install the entries for the new fonts right into the main `Fontmap` file. Otherwise just leave the file `Fontmap.ttf` in the Ghostscript configuration directory.
- **HINTSUBST** – if the value is set to **YES** use the option **-H**, otherwise don't use it. This option enables the hint substitution technique. If you have not installed the X11 patch described above, use this option with great caution. See further description of this option in the `README` file.
- **ENFORCEISO** – if the value is set to **YES** then disguise the resulting fonts as the fonts in `ISOLatin1` encoding. Historically this was necessary due to the way the installer scripts created the X11 font configuration files. It is not necessary any more for this purpose. But if you plan to use these fonts with some other application that expects `ISOLatin1` encoding then better enable this option.
- **ALLGLYPHS** – if the value is set to **YES** then include all the glyphs from the source fonts into the resulting fonts, even if these glyphs are inaccessible. If it's set to **NO** then include only the glyphs which have codes assigned to them. The glyphs without codes can not be used directly. But some clever programs, such as the Type 1 library from `XFree86 3.9` and higher can change the encoding on the fly and use another set of glyphs. If you have not installed the X11 patch described above, use this option with great caution. See further description of the option option **-a** in the `README` file.
- **GENUID** – if the value is set to **YES** then use the option **-uA** of the converter to generate UniqueIDs for the converted fonts. The standard X11 Type 1 library does not use this ID, so it may only be necessary for the other applications. The script is clever enough to generate different UniqueID for the same font converted to multiple encodings. Also after conversion it checks all the fonts generated during the session for duplicated UniqueID and shows those. Still, this does not guarantee that these UniqueIDs won't overlap with some other fonts. The UniqueIDs are generated as hash values from the font names, so it's guaranteed that if the `'convert'` script runs multiple times it will generate the same UniqueIDs during each run. See further description of this option in the `README` file.
- **GENUID** – if the value is set to **YES** then create the `.pfb` files, otherwise the `.pfa` files. The `.pfb` files are more compact but contain binary data, so you may experience some troubles when transferring them through the network.

The following parameters are used to locate the other scripts and configuration files. By default the scripts do a bit of guessing for them: they search in the `tff2pt1` installation directory if `tff2pt1` was installed or otherwise suppose that you are running `'convert'` with `'scripts'` subdirectory being the current directory.

- **ENCDIR** – directory containing the descriptions of encodings
- **MAPDIR** – directory containing the external map files

Besides that a few parameters are built into the `'convert'` script itself. You probably won't need to change them:

- **T1ASM, TTF2PT1, TRANS, T1FDIR, FORCEISO** – paths to the other script

Also there are a few parameters controlling the installation of fonts for Ghostscript. Please look at their description in the Ghostscript section of documentation or in the `tff2pt1_x2gs(1)` manual page before running `'convert'`. If these parameters are set, `'convert'` will call the `'x2gs'` script automatically to install the newly converted fonts in Ghostscript.

After creating the configuration file run the `'convert'` script. Look at the result and the log file in `DSTDIR`.

Add the directory with newly converted fonts to the configuration of X server or font server. For most of the systems this step is very straightforward. For `HP-UX` it's rather tricky and poorly documented, so the file `FONTS.hpux` gives a short description.

If you don't have the privileges of the root user, you still can configure your private font server. Just use some non-standard port number (see `FONTS.hpux` for an example, except that you won't need all the

HP-related stuff on any other system).

FILES

- TTF2PT1_SHAREDIRE/scripts/convert.cfg.sample
- TTF2PT1_SHAREDIRE/scripts/*
- TTF2PT1_SHAREDIRE/README
- TTF2PT1_SHAREDIRE/FONTS
- TTF2PT1_SHAREDIRE/*
- TTF2PT1_BINDIRE/ttf2pt1

SEE ALSO

- the *tff2pt1(1)* manpage
- the *tff2pt1_x2gs(1)* manpage
- the *t1asm(1)* manpage

BUGS

Known problems

- One catch is that the X11 Type 1 font library has a rather low limit on the font size. Because of this the fonts with more complicated outlines and the enabled hint substitution may not fit into this limit. The same applies to the fonts with very complicated outlines or with very many glyphs (especially the fonts with over 256 glyphs). So you will need to exercise caution with these options if you plan using these fonts with X11. Some vendors such as HP provide the Type 1 implementation licensed from Adobe which should have no such problem.

But there is a solution even for the generic X11. A patch located in the subdirectory 'app/X11' fixes this problem as well as some other minor problems. Its description is provided in app/X11/README.

To fix the X11 font library, you have to get the X11 sources. I can recommend the ftp sites of the XFree86 project ftp://ftp.xfree86.org or of the Open Group ftp://ftp.x.org. This patch was made on the sources of XFree86 so you may have better success with applying it to the XFree86 distribution. After you have got the sources, make sure that you can compile them. Then apply the patch as described. Make sure that it was applied properly. Compile the sources again (actually, you need only the fonts library, the fonts server, and possibly the X server). It would be prudent now to save your old font library, font server and, possibly, X server. Then install the new recently compiled versions of these files. Of course, if you know someone who already has compiled these files for the same OS as yours, you can just copy the binary files from him.

Alas, building the X11 system from the source code is not the easiest thing in the world and if you have no experience it can be quite difficult. In this case just avoid the aforementioned features or check each converted font to make sure that it works properly.

- The Type1 font library from the standard X11 distribution does not work on HP-UX (at least, up to 10.01). The font server supplied with HP-UX up to 10.01 is also broken. Starting from HP-UX 10.20 (I don't know about 10.10) they supply a proprietary font library and the converted fonts work fine with it, provided that they are configured properly (see the file FONTS.hpux).
- The fonts.scale files created by the older versions of the ttf2pt1 installation program (up to release 3.1) have conflicted with the language definitions of the xfsft font server and parts of it included into XFree86. To overcome this incompatibility the newer versions create the fonts.scale file describing all the fonts as belonging to the adobe-fontspecific encoding and the fonts.alias file with the proper names. The drawback of this solution is that xlsfonts gives the list of twice more fonts. But as a side effect the option **ENFORCEISO** in 'convert.cfg' is not required for X11 any more.

- The conversion script has no support for Eastern multi-plane fonts. Contribution of such a support would be welcome.