

# Getting Started with Neatroff

*A.G. Rudi*

The present document explains the steps necessary for setting up and using Neatroff. It uses neatroff\_make Git repository, which contains a set of standard macro packages and a top-level makefile to obtain and build Neatroff and its helper programs, which are referred to as neat\* throughout this document. More details about Neatroff and the programs that accompany it are available in its homepage at <http://litcave.rudi.ir/>.

## Using Neatroff Without Installation

To use Neatroff without installing it, the “master” branch of neatroff\_make can be retrieved as follows:

```
$ git clone git://repo.or.cz/neatroff_make.git
```

This branch assumes that the resulting directory will not be removed and shall contain Neatroff fonts, macros, and binaries when using Neatroff. The “init” make target, clones the necessary Git repositories and obtains Ghostscript fonts. The “neat” target compiles the programs and generates Neatroff font descriptions. Finally, the demo/ subdirectory contains a small example and a Makefile to demonstrate how to use Neatroff.

```
$ make init
$ make neat
$ cd demo && make
```

To add new fonts, simply place them in the fonts/ subdirectory and remake the “neat” target. To use the new font in Neatroff, the file name without its extension may be mounted. For instance, if the name of the font is NewFont.ttf, the following troff code mounts it:

```
.fp 11 F1 NewFont
```

```
.ft F1
Text in NewFont
```

## Installing Neatroff

This section describes how to install Neatroff in system directories. The following commands fetch neatroff\_make and obtain the latest versions of neat\*:

```
$ git clone -b install git://repo.or.cz/neatroff_make.git
$ cd neatroff_make/
$ make pull
```

This obtains Neatroff, neatpost, neateqn, neatrefer, and a port of Plan 9 troff to use its pic and tbl preprocessors (sadly there is no neatpic and neattbl!).

To build neat\*, neatroff\_make/makefile should be modified to set the values of GSFONTS and PREFIX macros. GSFONTS should point to the directory containing the standard Ghostscript fonts (ghostscript-fonts package, containing files such as n021003l.afm). In most environments these files are installed in /usr/share/ghostscript/fonts or /usr/share/fonts/type1/gsfonts; you can manually obtain the ghostscript-fonts package and extract it to a temporary directory, if they are missing. Also PREFIX specifies the installation prefix. The following commands build and install neat\*.

```
$ make
$ make install
```

Note that the second command may need to be executed by a superuser depending on the directory specified as PREFIX.

At this point neat\* should be installed. If Neatroff is set up properly, the following command should create test.pdf from the input troff source test.tr (you need to add other preprocessors if you use them).

```
$ echo "Hello Neatroff!" >test.tr
$ cat test.tr | neatroff | neatpost | ps2pdf - test.pdf
```

## Adding Fonts

A remarkable design decision in troff was the separation of output devices, for instance for Postscript, from the troff typesetting program. This separation requires generating device-independent font descriptions, listing available glyphs for each font and their metrics. Neatroff's font descriptions can be generated with the neatmkfn program as follows:

```
$ neatmkfn -b -a <fontpath.afm \  
    >PREFIX/share/neatroff/font/devutf/fontname  
$ neatmkfn -b -o <fontpath.ttf \  
    >PREFIX/share/neatroff/font/devutf/fontname
```

After generating font description, the new font can be mounted in troff just as other fonts with .fp request:

```
.fp 12 F2 fontname
```

Alternatively, you can place your fonts in the directory specified as GSFONTS when building neatroff\_make; the makefile generates and installs font descriptions for all fonts in that directory automatically. This is specially convenient when the number of fonts is large.

There is another method of using fonts in Neatroff that creates the font descriptions on the fly. Despite its overhead, this method may be convenient when testing new fonts. It uses the fp macro package, which is included in neatroff\_make. Assuming that the new fonts are in /path/to/fonts, the following command informs the macros defined in this package and Ghostscript about the location of the fonts (note that -mfp is passed to Neatroff to read this package).

```
$ cat test.tr | neatroff -dfp.src=/path/to/fonts -mfp | \  
    neatpost | ps2pdf -sFONTPATH=/path/to/fonts - test.pdf
```

The package defines .fp.ttf, .fp.otf, and .fp.afm macros, whose behaviour is quite similar to the standard .fp request, except that the third argument should be the name of the font file without its extension. Thus, for mounting /path/to/fonts/NewFont.ttf, test.tr can contain:

```
.fp.ttf 13 F3 NewFont  
.ft FN
```

Testing the new font...

## More Information

### *Neatroff Introduction*

Explains the differences between Neatroff and other troff implementations.  
Available at <http://litcave.rudi.ir/neatroff.pdf>.

### *Typesetting Mathematics with Neateqn*

Introduces the neateqn preprocessor for typesetting mathematical equations.  
Available at <http://litcave.rudi.ir/neateqn.pdf>.

### *Neatroff Introduction in Farsi*

Explains specifying text direction in right-to-left languages and Keshideh adjustment in Farsi. Available at <http://litcave.rudi.ir/neatfarsi.pdf>.