Lout FAQ/HOWTO

by Eric Marsden *emarsden@mail.dotcom.fr* 2001-10-11

This document contains information about the Lout text formatting system. It includes a collection of questions frequently asked on the Lout mailing list.

Contents

1 Introduction

This document contains information about the Lout text formatting system. It includes a collection of questions frequently asked on the Lout mailing list. Please read through this document before posting a question to the Lout mailing list.

The latest version of this file is available at *www.chez.com/emarsden/lout/*, in Postscript and ASCII formats. There is also an online version with embedded hyperlinks.

 Feedback is welcome; please send remarks, suggestions and corrections to <emarsden@mail.dotcom.fr> (Subject: containing "lout").

2 What is Lout?

Lout is a high-level language for document formatting. It includes facilities for typesetting complex documents containing floating figures, tables, diagrams, rotated and scaled text or graphics, footnotes, running headers, footers, an index, table of contents and bibliography, cross-references, mathematical equations and statistical graphs. Lout's capabilities can be extended with definitions which are easier to write than similar markup languages.

Lout is multilingual, supporting (with hyphenation) Czech, Danish, Dutch, English, Finnish, French, German, Norwegian, Russia, Slovenian, Spanish and Swedish. Some of these languages are available as separate packages because they need more than 13 standard Adobe fonts. Further languages can be added easily.

The Basser Lout implementation is free software distributed under the GNU General Public License. It was designed and implemented by Jeffrey H. Kingston at the University of Sydney.

2.1 What platforms does Lout run on?

The Basser Lout (the lout compiler written by Jeffrey Kingston) is written in ANSI C. It runs best on a real operating system like Un*x, but has been ported to

- MS-DOS : see the next section "Where can I get Lout?"
- OS/2
- Win32
- BeOS
- QNX (thanks to Chris Herborth <*chrish@qnx.com*>)

- MacOS users have several choices. Lout is reported to compile under the MPW shell, which is now distributed free of charge by Apple. You may want to consider a Un*x such as MkLinux (a free, high-powered Un*x for PowerPC Macintoshes), or MacMiNT (a freeware Un*x lookalike which runs as an ordinary Macintosh application concurrently with others), or MachTen (similar but commercial). A standalone port is underway by Robert Marsa <rmarsa@felipe.phyast.pitt.edu>.

Lout produces Postscript, a page description language understood by most laser printers. Postscript can also be printed on pretty much any printer (and previewed on the screen) by using the Ghostscript interpreter, which is distributed under the GNU Public License. See the Postscript FAQ (available by anonymous ftp at *wilma.cs.brown.edu:pub/comp.lang.postscript/FAQ.txt*) for more details. There is also an experimental PDF (Adobe's Portable Document Format) backend, which doesn't work in the latest versions, and which doesn't support the diagramming features in Lout (since they build on the programming language features in Postscript, which are not present in PDF). Lout can also produce plain text output, though any graphics details will be lost.

2.2 Where can I get Lout?

The primary site for Lout is Jeffrey Kingston's ftp site, at *ftp.cs.esyd.edu.au:/jeff*. This site is in Australia; most users will be better off using one of the following mirror sites :

- CTAN mirrors such as *www.text.ac.uk/tex-archive/support/lout* (say thank you to the TeX people for supporting this) ;
- A mirror of Jeff's site in Sweden at *ftp.sunet.se:/pub/text-processing/lout/lout* ;
- Another mirror is *sunsite.aux.dk:/pub/text/lout* ;
- *sunsite.unc.edu:/pub/Linux/apps/wp/lout* ;
- For users of Debian GNU/Linux, there is a prepackaged binary distribution available in the `text` section on Debian mirrors all over the world ;
- Users of Red Hat Linux will find Lout packaged as an `.rpm` on Red Hat CDs or ftp mirrors ;
- A native win32 port is available from *ftp.ptc.spbu.ru:/pub/textproc/lout/binaries/win32*. This site is in Russia with a slow Internet connection, so please download overnight local time (the ftp server prints local time in its banner) ;
- Binaries for MS-DOS are available in the gnuish section of Simtelnet, available at *ftp.simtel.net:/pub/simtelnet/gnu/gnuish* ;
- Archives of the USENET group comp.sources.misc.

All distributions should include (either in the main package or separately) complete documentation : the Users' Guide with a Quick Reference, and the Expert's Guide. To build from a source distribution you will need an ANSI C compiler such as `gcc`.

2.3 So is Lout like a word processor?

Lout is not a word processor like WordPerfect or MS-Word. These programs are WYSIWYG (what you see is what you get); the text displayed on your computer's screen is an approximation to what will appear on the printed page. Lout, in contrast, is a markup language. This means that if you want **bold** or *italic* text, you write (with your favorite text editor)

```
if you want @B{bold} or @I{italic} text,
```

Other symbols are used to identify headings, paragraphs, footnotes and so on. The lout interpreter reads these commands and produces nicely formatted output. The comp.text FAQ contains a more complete discussion of the relative merits of markup and WYSIWYG ; for starters you may wish to ask yourself how many different (and incompatible) `.doc` formats Microsoft has unleashed upon the unsuspecting users of successive releases of MS-Word.

2.4 Why use Lout instead of LaTeX or troff?

Lout is similar in function to LaTeX and troff. Indeed, it borrows ideas, techniques and conventions from these typesetting systems. For example, Lout uses Knuth's (the author of TeX, on which LaTeX is based) optimal line breaking algorithm, and has extended it to paragraph breaking across pages. For simple documents, Lout, LaTeX and troff offer much the same functionality, with different syntax (see the "Simple Examples" section). Lout is much more "programmer friendly" than TeX's macros (and à fortiori than incomprehensible troff macros). See the "Advanced Examples" section.

Lout makes it easy to mix text and graphics. You can draw lines, arrows and boxes, scale and rotate objects, use color commands. While many of these things are possible in LaTeX by including Postscript files generated by utility programs such as `xfig`, you have to specify the size of each included figure, losing a lot of Lout's flexibility.

The Lout distribution is very easy to compile and maintain, which certainly is not the case of many TeX distributions. The Lout distribution is much smaller (it fits onto a floppy disk) than LaTeX, and doesn't require storing `tfm` and `pk` font outlines (since Postscript fonts are used). Lout Postscript files are more compact than those produced from the process `.tex -TeX-> .dvi -dvips-> .ps`.

Lout is multi-lingual out of the box, and understands ISO-8859 latin-1 characters.

On the other hand, TeX/LaTeX is much more widely used than Lout (TeX has been around since the late 1970s, Lout only since 1991). It will be easier to find a local TeXpert than a Louter, and there are many more user-contributed packages for LaTeX than for Lout. Many academic journals request (or require) that papers be submitted in LaTeX. Lout uses more memory than TeX, up to 10 Mb to compile large documents.

Lout is more friendly to advanced users. LaTeX style sheets are written in a twisted, diabolical manner apparently designed to make all but trivial changes to a document's appearance difficult ; it's internals are packaged as a "black box" which ordinary mortals aren't meant to understand. Lout styles are much easier to design and modify according to your needs.

The Lout formatting language is conceptually cleaner and higher level than TeX (which was designed to run efficiently on computers from an other era). For example, Lout has no built-in notion of a "page".

3 Lout Packages

3.1 Can I produce HTML from my Lout documents?

The short answer is no, not easily. The problem is that Lout is quite different from HTML. Lout is a powerful language which allows you to place text or graphics at a specific point on the page, and the ease with which this is possible is one of its nicest features. HTML is –purposefully– very poor in this respect.

This being said, in the same way that Lout can be told to produce plain text output rather than Postscript, it could also produce HTML. As Jeff Kingston has pointed out, this would require writing another backend which produces HTML equivalents for all the Lout formatting tags, ignoring those it can't render correctly.

Another way of proceeding would be to use a postprocessor (a Perl script, for example) to carry out the conversions, stripping out unconvertible information. This is the way `latex2html` works.

You may want to consider using another package if you need to export to many different formats. Other packages you might use include

- Texinfo (distributed with GNU Emacs) lets you produce info files (the GNU hypertext format), Postscript (via TeX) and HTML.
- POD is the format for documentation used in the *Perl* world. POD uses a very simple markup and can be converted to Postscript, man pages, formatted text, etc.
- SDF (Simple Document Format) is available from *www.mincom.com/mtr/sdf*.

3.2 Is there an Emacs mode for Lout?

There are two Emacs modes at *www.chez.com/emarsden/lout*.

3.3 Is there an archive of user-contributed document formats?

Nimrod Zimerman maintains an archive at *http://www.fortunecity.com/skyscraper/raytrace/358/lout/index.html*.

4 Lout Problems

4.1 Lout produces ugly output in Ghostscript

```
When I view the output from lout using Ghostview, a well known front end to Ghostscript,
the spacing is all screwed up.  Words appear run together as though they were a single
word, yet printing on a laser, postscript printer always produces beautiful results.
```

The problem comes from poor font descriptions included with older versions of Ghostscript, or from an incompatibility with them. If the font formats used by Lout for calculating word and letter spacing don't match those used by the Ghostscript interpreter, obviously you lose. Recent versions of Ghostscript use higher quality fonts and support antialiasing : it's well worth upgrading (since v6.01 the PDF support is also much improved).

4.2 Spaces using Latin2 fonts

The standard Adobe fonts are missing certain Latin2 glyphs, which results in special characters (such as `@Char odblacute` and `@Char udblacute` in Hungarian) being represented by whitespace in the generated Postscript.

The solution is to use alternative Latin2 fonts. Uwe says:

I know about Quasy Palladio and several Antykwa fonts. ParaType has Literaturnaya in TeX T1 encoding which, IIRC, comprises alphabet parts of both Latin1 and Latin2 (unfortunately they use some non-standard glyph names).

Quasy Palladio can be found on CTAN in fonts/psfonts/polish. Literaturnaya can be found at ftp://ftp.vsu.ru/pub/tex/literat* (you need the T1 fonts, there are also IS (ISO8859-5 cyrillic) and TS (misc punctuation)).

Rafal Polonski suggests the following additional fonts for Polish:

- Antykwa Toru/nska from *ftp.GUST.org.pl:/pub/TeX/fonts/psfonts/polish/antyktor* ;
- Antykwa P/o/ltawskiego (a bit BETA) from *ftp.eps.gda.pl:/antp004* ;
- Quasi Times from *ftp.GUST.org.pl:/pub/TeX/fonts/psfonts/polish/qfonts/time0100* ;
- Postscript version of TeX Computer Modern fonts, from *ftp.GUST.org.pl:/pub/TeX/fonts/psfonts/polish/plpsfont.*
- The above fonts will only be useful for Polish, since they do not support non-Polish Latin2 characters like `@Char rcaron`. Matej Cepl has worked on some full Latin2 fonts which are available at *www.volny.cz/cepls/ps-pdf/yal2f-0.9.zip.*

It is also possible to convert TrueType or type42 fonts for use in Lout; see the comp.fonts FAQ at *www.nwalsh.com/comp.fonts/FAQ/* for details.

4.3 Acrobat Reader says "Bad /BBox in font"

This is the software's way of telling you that the font is unknown to it. Unfortunately there is currently no way of embedding custom fonts in PDF documents generated by Lout, so you are limited to the *standard* fonts.

As a workaround, you could try generating PostScript then converting it to PDF. Adobe Acrobat Distiller (commercial) is able to embed fonts when converting from PostScript to PDF. Aladdin Ghostscript 6.0 (free software) is now able to embed fonts in its PDF backend, so ps2pdf should work well.

You can check which fonts are used in a PDF file from Acrobat Reader by selecting File > Document Info > Fonts. Select "List All Fonts" in the font information window: under `Used Font` is should say `Embedded` or `Embedded Subset` if you're not using a standard font.

4.4 Huge page sizes

Lout may have problems generating documents at very large page sizes: in the file externs.h you'll find MAX_SHORT_LENGTH which is 32767 internal units. Since Lout works internally with units of 1/20 pt,

$32767 = 1638.35$ pt $= 22.7(5)$ in $= 57.79$ cm

which is enough for B3 paper, but just a little bit too small for A2 (420mm x 594mm).

There is a way around this limitation: you can add a Postscript command such as

```
<</PageSize [1584 2448]>>setpagedevice
```

which sets the page size for the printing device. You can use a package called *psenlarge* to add these setpagedevice constructs to a Postscript file. It requires perl5.x and Ghostscript, and is written by `Peter.Samuel@ind.tansu.com.au`, and is available from *ftp://ftp.uniq.com.au/pub/tools/psenlarge-1.4.tar.gz.*

4.5 Unresolved cross references on Linux

You may run into problems with lout compilation not converging on recent Linux systems. Symptoms of the problem are error messages such as

```
lout file "all":
    14,1: unresolved cross reference @Roman&&1
    14,1: unresolved cross reference @Roman&&2
    14,1: unresolved cross reference @Roman&&3
```

This problem arises if you are using a non-english locale and have a glibc system. The problem arises from the collation rules under certain locales ignoring the tab character when comparing strings. Each line in the index file generated by lout consists of a key followed by position information, separated by tab characters. When writing the index lout sorts entire lines, but when reading entries it sorts according to the key; since the orders can be different, lout assumes that it hasn't done been run sufficiently often to generate all index entries.

A "solution" is to set the environment variable `LC_COLLATE` to `en_UK` (or some other locale where the tab character is collated before printable characters, and blank space and punctuation characters are not ignored during collation). Alternatively, simply run lout with the `-l` option to turn off collation.

5 How do I ... ?

5.1 How do I get quotes (") or backslashes (\) in Lout?

Rodrigo Vanegas <rv@cs.brown.edu> (former maintainer of the Lout mailing list) writes:

```
Quotes are explained in the reference manual, section 2.1.

      To get this              You type this

           "                        \"
           \                        \\


but these things only work inside "...".  So the way to print quoted strings in source
code listings is like this (for example) :

          puts("hello\n");      puts("\"hello\\n\"");
```

In other words, you replace the opening " by "\", you replace the closing quote by \"", and any \ inside you replace by \\.

5.2 How do I use accented characters in Lout?

The best way is to compile Lout in 8-bit mode and to use an editor which allows the insertion of accents. In Emacs, for example, you can use iso-accents-mode.

If this isn't possible you can use a simple script to transform "prefixed-accents" to Lout `@Char` form. For French accents, for example :

```
#!/usr/local/bin/perl
while( <> )
{
  s/'e/\@Char{eacute}/g;
  s/`e/\@Char{egrave}/g;
  s/`a/\@Char{agrave}/g;
  s/`u/\@Char{ugrave}/g;
  s/\^e/\@Char{ecircumflex}/g;
  s/\^o/\@Char{ocircumflex}/g;
  s/\^u/\@Char{ucircumflex}/g;
  # etc
```

```
    print;
}
```

A more complete set of conversions is provided by the txt2lout script by Mark Summerfield (see "Useful Tools" section).

5.3 How do I print only a selection of pages of a Lout document?

This sort of task is best left to a Postscript postprocessor. Lout outputs Postscript which conforms closely to the Adobe Document Structuring Conventions (see the file notes.dsc included with the Lout distribution). This means, among other things, that it includes page numbering information in its output, which is why Postscript previewers permit you to skip to an arbitrary page in the document, and which makes it possible to select specific pages from Lout's output.

PS Utils http://www.dcs.ed.ac.uk/~ajcd/psutils/ by Angus Duggan <angus@harlequin.co.uk> is a great postprocessor for Postscript documents. Some of the programs in the suite are

psresize

alter document paper size

psselect

selects pages and page ranges

psbook

rearranges pages into signatures

psnup

put multiple pages per physical sheet of paper

pstops

performs general page rearrangement and selection

5.4 Adding new fonts

Uwe explains how to add a new font to Lout:

You will need two files for each font: an AFM and a PFA/PFB. The AFM (Abode Font Metrics) file describes the metrics of the font, providing incformation about glyph sizes, ligatures, kerning etc which is necessary to Lout. AFM is documented in Adobe's Tech Note #5004, *Adobe Font Metrics File Format Specification*. For Type1 fonts (widely used under Windows and MacOS) the metrics are usually in PFM (Printer Metrics Format, documented in the Windows 3.x DDK). Ghostscript 6 includes a program called pf2afm which creates an AFM file from a clean PGB/PFA file. When a PFM file is available it uses it to provide better kerning information (as well as extracting other information not available in PFB/PFA files).

Font vendors generally provide both AFM and PFM metrics, so this should not be a problem.

The second file required for printing is a PFA (Printer Format ASCII), which you need to embed in your documents or download to the printer. Most Type1 fonts are in PFB format (Printer Format Binary); you can used *pfbtops* from the groff distribution or *t1ascii* from t1utils to convert PFB to PFA.

Ghostscript can use either PFA or PFB (the latter takes less space), but since you will need the PFA anyway, it's simpler to use PFA everywhere.

- Arrange for Lout to be able to find the AFM file for the new font. E.g. put it into the `$LOUTLIB/font` directory or specify the directory with -F<dir> option. AFM is the only thing Lout needs or cares about.

  If you use a non standard encoding for the font, you will have to write a LCM (Lout Character Mapping) file describing your encoding. Generally one of the existing LCMs, such as LtLatin1.LCM or LtLatin2.LCM will work (see `$LOUTLIB/maps` for how Lout looks for LCM files).

  You will then need a fontdef statement to declare the font. You can put fontdef anywhere you can put a definition, such as in `mydefs.lt` or before the `@Include` of your document setup. You could alternatively append to Lout's `include/fontdefs` file, or create a new file for site fonts and include it in your setup file. Here is an example fontdef for a hypothetical `Fooney-Regular` font:

```
fontdef Fooney Base {
    Fooney-Regular          # font name, see FontName in the AFM
    Fooney-Regular.AFM      # AFM file
    LtLatin1.LCM            # LCM file
    Recode                  # Recode (or NoRecode if you
                            #         know what you're doing)
}
```

  Here `fontdef Fooney Base` declares Lout's Base face of family `Fooney`. The name `Base` is a matter of convention, standard setups use Base to denote the regular, or roman or upright or whatever-the-base-face-is-called.

  See `$LOUTLIB/include/fontdefs` for examples of how complete families are defined. You just repeat the same steps for all the fonts in the family, only using `Slope` (i.e. italic) `Bold` and `BoldSlope` instead of `Base`.

  Lout will put appropriate `%%IncludeResource` DSC comments in its PostScript output so that your spooler can process it.

- Arrange for your PostScript environment to be able to find the font. This is highly system dependent. That's why DSC exists. Lout produces strictly DSC conforming output (well, almost; see file `notes.dsc` in the Lout distribution).

  For GhostScript you put the font somewhere where GS can find it (GS_LIB_DEFAULT compilation parameter, $GS_LIB environment variable and -I option), and add an entry or the font to the Fontmap file.

  Or you can configure your spooler to perform resource inclusion. If you use 'includeres' from PSUtils then make a symlink to the font PFA file in the directory where 'includeres' looks for resources. The name of the link should be exactly the font name without any suffixes.

5.5 Embedding Postscript fonts in a document

```
I have seen a lot of TeX users who embed Postscript fonts into their documents to make
life easier for people who read their document and don't have appropriate font.  Of
course such .ps files are much bigger, but I am wondering if it is possible to do the
same with Lout.
```

With Ghostscript 6.0 and up, Postscript fonts are automatically embedded. Users of older version will need to use 'includeres' from *PSUtils* (see the Resources section).

Uwe writes: One word of caution, though. Lout put `%%IncludeResource` comments into page setup section, only fonts of the first page goes to the document setup section. So if you use some font on every page except the first a copy of font will be inserted into every page setup section which will really bloat the result. You might want to move all the `%%IncludeResource` comments (for commonly used fonts) into document setup before running 'includeres'.

Note that embedding is not necessary if you're only printing locally: you can configure your printing software so that it knows the font. This requires you to download the necessary fonts to the printer's Postscript interpreter: see Adobe TechNote #5040, or for ghostscript you simply need to put the PFA or PFB file in the `fonts` directory and add an entry to the Ghostscript Fontmap file. Embedding only works for Postscript fonts; it is currently not implemented for PDF documents.

5.6 Flowing text across columns

```
Is it possible to split a two-column page (as in I guess, have a column
break)? The effect I am looking for is:
 - some text flow into two columns in the top third/half of a page
 - column break signified by a column-spanning ruled line
 - more text (also flowing into two columns)
```

Jeff answers (executive summary: probably not in a way that would suit you or that could be made to happen quickly):

The standard Lout packages are not set up to do this. Someone who set out to write some definitions to do it would strike a problem:

```
A      B
--------
C      D
```

If text flows into A then on into B, the only reason why it would stop flowing into A is because space there runs out. But if space there runs out there would be no space for C either. Lout can't see that it would be a good thing to make columns A and B have equal height.

The only approximation would be to decide in advance how much height to allow for A and B, and to use a `@High` symbol to restrict them to this height. Then when A fills up the text would flow on to B, but this filling would occur when the height limit was reached, allowing C (which is outside the `@High` and not affected by it) to receive other text later.

5.7 Using BibTeX databases

BibTeX is a widely used format for storing bibliographic references. A few tools to convert bibtex databases to the format required by Lout are available from *http://turtle.gis.umn.edu/people/dajm/lout/* `<turtle.gis.umn.edu/people/dajm/lout/>` (written by David Middleton).

5.8 LaTeXisms

A person coming to Lout from LaTeX will notice several differences between the systems. With its default settings, Lout takes into account all inter-word white space (as does troff). You can ask Lout to use the LaTeX conventions (multiple spaces are treated as a single space) with the `@InitialSpace` option (see the User's Guide section 1.18).

LaTeX lets you signal non-breaking spaces (as in Dr. Watson, where you don't want a line break between Dr. and Watson) with a ~ (tilde). The same effect can be obtained in Lout with the following definition (predefined in recent versions of Lout) :

```
def "~" left x right y { @OneRow { x &1s y } }
```

LaTeX considers that an empty line signals the start of a new paragraph, whereas Lout waits for an explicit @PP (indented paragraph) or @LP (non-indented paragraph) symbol.

You can simulate TeX glue with the following definitions

```
def @VFill { @VExpand {} }
def @HFill { @HExpand {} }
```

5.9 Related resources

Here is a list of packages which you may find useful in conjunction with Lout (please send me any additional suggestions):

- xxx2lout is a collection of perl scripts written by Mark Summerfield which convert from different formats such as HTML, POD, text and perl to Lout. There is also an experimental Lout to HTML converter, which works with very simple documents (tables etc are not handled very well yet). xxx2lout is available from *ftp://ftp.bora.net/pub/CPAN/authors/id/S/SU/SUMMER/*.
- MetaPost *http://plan9.bell-labs.com/cm/cs/who/hobby/MetaPost.html* is a language which resembles MetaFont, which can generate Postscript figures. Its ability to solve linear equations expressed implicitly allows figures to be "programmed" in a declarative style.
- idraw *http://www.vectaport.com/ivtools/idraw.html*
- daVinci *http://www.tzi.de/~davinci/* is a powerful system for drawing directed graphs.

The Linux Gazette, a free online monthly newsletter, has published two articles mentioning Lout:

- *www.linuxgazette.com/issue27/adelman.html*
- *www.linuxgazette.com/issue22/words.html*

5.10 Whah! I want Lout to do this! Lout shouldn't work this way!

Unlike TeX (which Don Knuth has frozen for eternity now that he considers that most bugs have been detected), Lout is evolving. For example, the treatment of whitespace was changed in 1995 (two options were added at users' request). If you have a suggestion for improvement, gripe loutly and Jeff will hear you (to quote Uwe :-).

6 Simple Examples

The "Slides" section of the documentation distributed with Lout contains *A Pratical Introduction to the Lout Document Formatting System*, by the author of Lout.

7 Advanced Examples

To be completed ... contributions welcome.

8 Further information

Valeriy Ushakov maintains a home page for Lout at *www.ptc.spbu.ru/˜uwe/lout/*. He also runs a public mailing list for discussion of Lout. To subscribe to the list, send mail to

```
lout-request@ptc.spbu.ru
```

containing the word `subscribe` in the *subject* line. To unsubscribe, send mail to `lout-request@ptc.spbu.ru` containing the word `unsubscribe` in the subject line (not in the body).

To post an item, send mail to `lout@ptc.spbu.ru`. It will be forwarded by email to all current subscribers. You can view recent messages on the list on the Web at *www.ptc.spbu.ru/˜uwe/lout/list.html*.

Freshmeat URL: *http://freshmeat.net/projects/lout/*.

End of the Lout FAQ/HOWTO.