

ext.nw1

The LEDA Tools for Manual Production and Documentation

Evelyn Haak
Kurt Mehlhorn
Stefan Näher
Christian Uhrig
04/18/21

Contents

1 The Manual Page of the Documentation Tools

LEDA offers tools for manual production and documentation.

```

_____ weave _____ LaTeX source
|
| ----- Lman ----- LaTeX source for manual page
|  foo
| ----- Ldoc ----- LaTeX source for full
documentation
|
| ----- Fman ----- ASCII version of manual page
|
|_____ tangle _____ program

foo's ----- Mkman ----- LaTeX source of manual
```

Lman produces manual pages from header files suitably augmented by so-called manual comments. Many examples can be found in the LEDA include directory (LEDAROOT/incl/LEDA). A manual comment starts with `*\M` and ends with `*/*`. Try `Lman list` to see Lman in action. If it does not work the error is very likely to be one of the following (if not, you should refer to the LEDA installation guide):

The environment variable LEDAROOT is not set to the root directory of the LEDA system.

\$LEDAROOT/Manual/cmd is not part of your PATH.

\$LEDAROOT/Manual/tex is not part of your TEXINPUTS.

```

LEDAROOT
|_____
| | | | |
incl Cweb Noweb LwebManual
|_____
LEDA | | | |
cmd noweb tex Manual
```

shows the parts of the LEDAROOT directory relevant for manual production and documentation. The include directory contains all header files of the LEDA system, the directories Cweb, noweb, and Lweb contains examples of full documentation written with various literate programming tools, and Manual contains all the relevant commands and tex files.

The Lman utility can be applied to LEDA data types and user-defined data types. Write

`Lman T[.h] options` to apply it to the LEDA data type T and write

`Lman T[.ext] options` to apply it to file T.txt in the working directory. The extensions lw (Lweb file), nw (noweb file), w (a Cweb-file), and h (h-file) are possible. The extensions are tried in this order and the working directory is searched before the LEDA include directory. The following options are available:

```
size=12,11,10
constref=no,yes
partypes=no,yes
usesubscripts=no,yes
```

numbered=no,yes
 xdvi=yes,no
 warnings=yes,no
 ack=yes,no
 latexruns=1,2,0
 filter=all,signatures,definition,creation,operations,implementation,example,opname
 pid="",string

Options are given in assignment syntax. There must be no blank on either side of the equality sign.

The size-option selects the font-size. The option constref determines whether const-ref pairs are shown, the option partypes determines whether the type of a value parameter of an operator is suppressed if equal to the currently defined type, and the option usesubscripts determines whether indexed variables are typeset using subscripts.

A manual page can be numbered or not and the call of xdvi can be suppressed. Lman may be asked to issue warnings and warnings may require a user action or not. We recommend to run Lman with warnings=yes and ack=yes in the early stages of designing a manual page. Lman usually runs L^AT_EX twice in order to get cross-references correct. You may settle for a single run of L^AT_EX. Lman makes use of a preprocessor called ext. The preprocessor writes its output on file /tmp/pid-ext.tex where pid is either the process identification of the preprocessor run or specified by the pid-option. Mkman makes use of the pid-option. You will probably never have to use it.

There are four ways to specify options for Lman. You may either put them into the file itself by means of the Moptions command, or you can put them onto the command line, or you can put them into a configuration file Lman.cfg in either the working directory or the home directory.

The Lman-utility can also be told to extract only part of a manual page by the filter-option. The options definition, creation, operations, implementation, and example cause extraction of the corresponding section of the manual, and the option signatures extracts the signatures of all operations but suppresses the descriptions. One can also extract a single operation by using its name, e.g., Lman stack filter=push extracts the push operation of the stack type and Lman point filter=operator+ extracts the + operator of the point type (write arrop to inquire about the array operator and funop to inquire about the function operator).

Fman is an alternative to Lman. It gives low quality ASCII output but works faster.

Mkman is the tool to produce manuals like the LEDA manual.

Ldoc is the utility that supports the production of full documentations. All remarks above apply equally well to Ldoc. Ldoc applies to Lweb, noweb, and Cweb files. For use with noweb and Lweb, the LEDA dialect of noweb, you must have installed noweb and for use with Cweb you must have installed Cweb. The subdirectories Cweb, noweb, and Lweb contain full documentations of parts of the LEDA system. Copy a file from one of these directories to your working directory and run Lman on it to see it in action. If it does not work check whether you have installed the proper literate programming system.

Lweave is the weave tool for Lweb.

2 Introduction

The production of manual pages and full documentations proceeds in phases.

Lman:

```

XXX.[h|lw|nw|w] -----> XXX.tex -----> XXX.dvi
  ext.pl      latex
  
```

lweave and lw2dvi:

```

XXX.lw -----> XXX.tex -----> XXX.dvi
  
```

lweave latex

Ldoc:

```
XXX.[lw|nw|w] ----> temp.[lw|nw|w] -----> XXX.
tex -----> XXX.dvi
  | ext.pl[lw|no|c]weave | latex
  | ext.pl |
  |
XXX.man -----
      \input
```

Fman:

```
XXX.[h|lw|nw|w] -----> ASCII
Fman
```

We use standard tools for all but the first phase. The first phase is accomplished by the perl program *ext.pl*. It realizes Lman, Ldoc and Fman in a very direct way.

```
2Lman perl LEDAROOT/Manual/cmd/ext.plLman*
3Ldoc perl LEDAROOT/Manual/cmd/ext.plLdoc*
4Fman perl LEDAROOT/Manual/cmd/ext.plFman1 25lextract#!/bin/csh?
fsetinfile=1 set outfile = 2shiftshiftperlLEDAROOT/Manual/cmd/ext.pl Mkman
infileoutfile=outfile *6Mkman#!/bin/csh?fif(1 == "") then set source = LEDAROOT/
incl/LEDAsertext=helsesetsource=1 if (2=="") then set ext = h else set ext = 2 endif
endif
rm -r -f extract
mkdir extract
echo Extracting manual pages ... echo " "
foreach f (source/*.*ext) echo "extracting manual from f" lextract f extract/
ibasename f.exti.tex end
```

7LEDAROOT is an environment variable that must be set to the LEDA root directory; see the LEDA installation instructions for details.

```
\documentclass[12pt,a4paper]article
\input MANUAL.pagesize
\input MANUAL.mac
```

```
\begindocument
\beginmanual
```

```
\section*Stacks(stack)\labelstack
```

```
\definition
```

An instance \$\$\$ of the parameterized data type $\mbox{\mathitstack}\langle E \rangle$ is a sequence of elements of data type E , called the element type of $\mbox{\mathitstack}\langle E \rangle$. Insertions or deletions of elements take place only at one end of the sequence, called the top of $\mbox{\mathitstack}\langle E \rangle$. The size of $\mbox{\mathitstack}\langle E \rangle$ is the length of the sequence, a stack of size zero is called the empty stack.

```
\creation
```

```
\create \mbox{\mathitstack}\langle E \rangle
\mbox{\mathitstack}\langle E \rangle
creates an instance \mbox{\mathitstack}\langle E \rangle of type \mbox{\mathitstack}\langle E \rangle and
initializes it to the empty stack.
```

```

\setlength\typewidth2.5cm
\setlength\callwidth4cm
\computewidths
\operations

\function \mbox$\mathitE$
\mbox$\mathitS.$top\mbox$\mathit()$
returns the top element of \mbox$\mathitS$.\ \precond $$ is not empty.

\function \mbox$\mathitvoid$
\mbox$\mathitS.$push\mbox$\mathit(E\ x)$
adds $x$ as new top element to \mbox$\mathitS$.

\function \mbox$\mathitE$
\mbox$\mathitS.$pop\mbox$\mathit()$
deletes and returns the top element of \mbox$\mathitS$.\ \precond $$
is not empty.

\function \mbox$\mathitint$
\mbox$\mathitS.$empty\mbox$\mathit()$
returns true if \mbox$\mathitS$ is empty, false otherwise.

\endmanual
\enddocument

```

Figure 1: The intermediate Tex-file for the example of Figure 1.2 in [LEDA\s\do5(B)uch, Chapter Manual Pages and Documentation] LEDA\s\do5(B)uch, Chapter Manual Pages and Documentation]" \f f

Figure ? shows the intermediate TEX-file produced by Lman for the data type stack shown in Figure . The TeX file basically contains one TeX-macro call for each manual command. It also has a short preamble and postamble. The preamble reads MANUAL.pagesize and MANUAL.mac, opens the document and enters the manual environment, and the postamble closes the manual environment and the document. The files MANUAL.pagesize and MANUAL.mac are both contained in the LEDAMAN directory; they contain the definition of the pagesize used for the LEDA manual and the definition of the manual macros respectively.

Ldoc applied to the same file produces (in stack.man) the subfile starting with \beginmanual and ending with \endmanual. Ldoc also produces a temporary file temp.w which is obtained from the input file by deleting all manual comments. It applies cweave to this file to obtain XXX.tex.

```

8MANUAL.pagesize
textwidth 16cm
textheight 24 cm
topmargin -14mm
evensidemargin 3mm
oddsidemargin 3mm
sloppy

```

```

9MANUAL.mac manual environment special symbols items and such header lines
widths functions constructors and destructors old macros

```

10

3 The Manual Environment

The manual environment redefines some basic parameters of LaTeX, namely parskip, parindent, and baselineskip. parskip governs the vertical space between paragraphs and parindent governs indentation at the beginning of a paragraph. In LaTeX the first quantity is zero and the second quantity is non-zero. The TeX file of Figure ? consists of many short paragraphs, each paragraph being basically a LaTeX command. We therefore want no indentation but we want paragraphs to be separated. baselineskip controls the vertical distance between lines. We increase its value slightly.

When leaving the manual environment we set all values back to their original values. This requires no code in LaTeX. When used in Ldoc this has the effect of using the parameter settings below for the manual and to use the standard Cweb settings for the remainder of the document.

```

1 manual environment
newenvironment manual
baselineskip 3.0ex
parskip 11pt plus 1pt minus 1pt
parindent 0pt 12

```

4 Ext: An Overview

Ext.pl operates in phases. The first phase processes the command line, takes care of initialization, and produces the preamble. The second phase does most of the work. It reads the input file and produces the body of a TeX-file. The third phase adds the postamble and calls cweave and/or latex. In Ldoc-mode the third phase also removes all With the option mode=Mkman ext carries out only the first two phases, the resulting file can then be included in some larger document, and with the option Fman ext produces ASCII-output on standard output.

The production of extjs output is controlled by the so-called manual comments in the input file, see the chapter on Manual Production in the LEDA-book for further details. You must be familiar with that chapter in order to understand this program.

The details of the behavior of ext are directed by options. The options are processed in the initialization phase of ext.

```

13 ext.pl required packages
command line processing and initialization main program generate postamble and
call LaTeX or Cweb
subroutines
14

```

5 Command Line Processing and Initialization

```

15 command line processing and initialization determine mode and set options to
default values determine actual options and derived variables set up input and output
files generate preamble

```

```

16 The first argument tells ext whether it works for Lman, Ldoc, Fman or
Mkmanual. If ext receives only one argument it prints usage information for its first
argument, see section ?. We have a variable for each option. The variables are
initialized to their default values.

```

```

17 determine mode and set options to default values
mode=ARGV[0] shift;
if (!(mode));
filearg=ARGV[0] shift;
if (mode eq "Mkman") numbered = "yes"; else numbered = "no"; # Usually manual
pages are not numbered ack = "yes"; # ext asks for acknowledgments when it is
confused constref = "no"; # const-
pairs are usually suppressed partypes = "no"; # operators suppress argument types that
are identical to Mname usesubscripts = "no"; # usually we do not write indexed
variables as subscripts size = 12; # default size is 12pt xdvi = "yes"; warnings = "yes"; #
default is to give warnings filter = "all"; # usually we show everything print(s\do5(t)itle
= 1; # usually we print the title pid = ""; # we do not know the process id yet latexruns =
1; map = ""; # the substitution map delman = "yes"; # Ldoc usually removes manual
comments outfile = ""; # the default is that we determine the outfile section = "section";
# each manual page is a LaTeX section nextwarning = "yes"; # default is to show the
next warning justset = "no"; # nextwarningwasnotjustsettono.18 In Fman?
modethereisonlyonefurtherargument,thefilter.
Intheothermodeswelookforoptionsinthefile$mode.
cfginthehomedirectoryorintheworkingdirectoryonthecommandline.
Lateroptionstakeprecedence.
Thevariable$pidissettotheprocessidofthecurrentprocess(availableinvariable$$)
iftheuserleavesitundefined.Weuseaderivedvariable$showsem.
19 determine actual options and derived variables if (mode eq "Fman") if (ARGV[0]) filter

```

```

= ARGV[0];foreachcandidate ("all","signatures","definition","creation", "operations",
"implementation","example","opname") if (filtereqcandidate) goto filterfound; print
"
nFman usage is
n Fman file filter
n where filter is one of all, signatures, definition, creation, operations, implementation,
example, opname
n"; die; filterfound: print "
n
n
n"; else foreach path("HOME","PWD") if (path eq "HOME") CFG=ENVpath."
/" . mode . "
.cfg"; else CFG=mode . "
.cfg"; if (-e CFG)open(CFG);print"reading ", CFG,"
n
n"; while (<CFG>) if /(
w+)=(.*)/) eval "

```

1=

```
2"; close (CFG); eval "
```

```
1 =
2" while ARGV[0] = /(
w+)=(.*)/ shift;
if (pideq"") pid =

```