

# L<sup>A</sup>T<sub>E</sub>X2man

—

## A Documentation Tool

Dr. Jürgen Vollmer

10th September 2005

Version 1.18

### Abstract

*Latex2man* is a tool to translate UNIX manual pages written with L<sup>A</sup>T<sub>E</sub>X into a format understood by the UNIX *man*(1)-command. Alternatively HTML, TexInfo, or L<sup>A</sup>T<sub>E</sub>X code can be produced too. Output of parts of the text may be suppressed using the conditional text feature (for this, LaTeX generation may be used).

## 1 Synopsis

*latex2man* [-**ttransfile**] [-**HMTL**] [-**h**] [-**V**] [-**Cname**] [-**a**char] *infile outfile*

## 2 Description

*Latex2man* reads the file *infile* and writes *outfile*. The input must be a L<sup>A</sup>T<sub>E</sub>X document using the *latex2man* L<sup>A</sup>T<sub>E</sub>X package. *Latex2man* translates that document into the *troff*(1) format using the **-man** macro package.

Using the **-H** option, HTML code can be produced, instead of *troff*(1).

Using the **-T** option, TexInfo code can be produced, instead of *troff*(1).

Using the **-M** option, *troff*(1) input is produced.

Using the **-L** option, L<sup>A</sup>T<sub>E</sub>X output can be produced, instead of *troff*(1).

## 3 Options

**-ttransfile** Translation for user defined L<sup>A</sup>T<sub>E</sub>X macros.

**-M** Produce output suitable for the *man*(1) command (default).

**-H** Instead of producing output suitable for the *man*(1) command, HTML code is produced (despite the name of the command).

- T** Instead of producing output suitable for the *man*(1) command, Tex-Info code is produced (despite the name of the command). The generated *.texi*-file may be processed with *makeinfo*(1) (to produce an *.info*-file) which in turn may be installed using *install-info*(1). The Info tags *@dircategory* and *@direntry* are provided.
- L** The L<sup>A</sup>T<sub>E</sub>X source is written to the *outfile*. This is useful in conjunction with the **-Cname** option.
- Cname** Output the conditional text for *name*. If more than one name should be given use quotes: **-C'name1 name2 ...'**  
The following names are defined automatically:
- **-H** defines HTML
  - **-T** defines TEXI
  - **-M** defines MAN
  - **-L** defines L<sup>A</sup>T<sub>E</sub>X
- achar** Is used only in conjunction with -T.  
Background:  
TexInfo ignores all blanks before the first word on a new line. In order to produce some additional space before that word (using \SP) some character has to be printed before the additional space. By default this is a . (dot). The *char* specifies an alternative for that first character. Giving a blank to **-a** supresses the indentation of a line.  
Note: only for the first \SP of a series that *char* is printed.
- h** Show a help text.
- V** Show version information.

## 4 Files

- latex2man.tex* The L<sup>A</sup>T<sub>E</sub>X file containing this Man-page.
- latex2man.sty* The L<sup>A</sup>T<sub>E</sub>X package defining the environments and commands.
- latex2man.cfg* The configuration file for *Latex2man* L<sup>A</sup>T<sub>E</sub>X-package.
- latex2man.trans* File containing example translations of user defined L<sup>A</sup>T<sub>E</sub>X macros.
- fancyheadings.sty* A L<sup>A</sup>T<sub>E</sub>X package used to typeset head- and foot lines.
- fancyhdr.sty* A L<sup>A</sup>T<sub>E</sub>X package used to typeset head- and foot lines.
- rcsinfo.sty* A L<sup>A</sup>T<sub>E</sub>X package used to extract and use RCS version control information in L<sup>A</sup>T<sub>E</sub>X documents.

## 5 See Also

L<sup>A</sup>T<sub>E</sub>X, TexInfo, *troff*(1), *groff*(1), *makeinfo*(1).

## 6 L<sup>A</sup>T<sub>E</sub>X commands

The L<sup>A</sup>T<sub>E</sub>X package `latex2man` is used to write the Man-pages with L<sup>A</sup>T<sub>E</sub>X. Since we translate into other text formats, not all L<sup>A</sup>T<sub>E</sub>X stuff can be translated.

### 6.1 Package Options

The `latex2man` package accepts the following options:

**fancy** use the L<sup>A</sup>T<sub>E</sub>X package `fancyheadings`.

**fancyhdr** use the L<sup>A</sup>T<sub>E</sub>X package `fancyhdr`.

**nofancy** neither the L<sup>A</sup>T<sub>E</sub>X package `fancyheadings` nor `fancyhdr` are used.

The default option may be specified in the file `latex2man.cfg`.

### 6.2 Package Specific Environments

The following environments are provided by the package:

`\begin{Name}{chapter}{name}{author}{info}{title}` The **Name** environment takes five arguments: 1. the Man-page chapter, 2. the name of the Man-page, 3. the author, 4. some short information about the tool printed in the footline of the Man-page, and 5. a text which is used as title, for HTML and L<sup>A</sup>T<sub>E</sub>X (it's ignored for output of the Man-page or TeXinfo). The **Name** environment must be the first environment in the document. Processing starts with this environment. Any text before this is ignored (exception: the `setVersion` and `setDate` commands). (Note: all arguments of `\begin{Name}` must be written on one line).

`\begin{Table}{columns}` The **Table** environment takes one argument: the number of columns. For example:

```
\begin{Table}{3}
Here & am & I   \\ \hline
A 1 & A 2 & A 3 \\
B 1 & B 2 & B 3 \\
\end{Table}
```

will be typeset as:

Here	am	I
A 1	A 2	A 3
B 1	B 2	B 3

`\hline` may be used. All entries are typeset left justified. If the Man-page is formatted with *troff*(1) and tables are used, the *tbl*(1) preprocessor should be called, usually by giving a `-t` to the call of *troff*(1). When viewing the generated manula page using *man*(1), *tbl*(1) is called automatically.

`\begin{Description}` is the same as `\begin{description}`

`\begin{Description}[label]` is similar to `\begin{description}`, but the item labels have at minimum the size of the (optional) word *label*. The difference is visible only in the DVI and PDF-output, not in the troff, TeXInfo or HTML output.

**a** |a `\begin{description}`

**ab** |ab

**abc** |abc

**a** |a `\begin{Description}`

**ab** |ab

**abc** |abc

**a** |a `\begin{Description}[aa]`

**ab** |ab

**abc** |abc

### 6.3 Accepted $\LaTeX$ Environments

The following environments are accepted:

- `description`
- `enumerate`
- `itemize`
- `verbatim`
- `center`

They may be nested:

- Itemize and nested center:

A centered line.  
Another centered line.

- Another item an nested enumerate

1. a
2. b

## 6.4 Package Specific Macros

The following commands are provided:

- `\Opt{option}` Option: `\Opt{-o}` will be typeset as `-o`.
- `\Arg{argument}` Argument: `\Arg{filename}` will be typeset as *filename*.
- `\OptArg{option}{argument}` Option with Argument:  
`\OptArg{-o}{filename}` will be typeset as `-ofilename`.
- `\OptoArg{option}{argument}` Option with optional Argument:  
`\OptoArg{-o}{filename}` will be typeset as `-o[filename]`.
- `\oOpt{option}` Optional option, e.g. `\oOpt{-o}` will be typeset as `[-o]`.
- `\oArg{argument}` Optional argument, e.g. `\oArg{filename}` will be typeset as *[filename]*.
- `\oOptArg{option}{argument}` Optional option with argument, e.g.  
`\oOptArg{-o}{filename}` will be typeset as `[-ofilename]`.
- `\oOptoArg{option}{argument}` Optional option with optional argument,  
e.g. `\oOptoArg{-o}{filename}` will be typeset as `[-o[filename]]`.
- `\File{filename}` used to typeset filenames, e.g. `\File{filename}` will be typeset as *filename*.
- `\Prog{prog}` used to typeset program names, e.g. `\Prog{latex2man}` will be typeset as *latex2man*.
- `\Cmd{command}{chapter}` used to typeset references to other commands,  
e.g. `\Cmd{latex2man}{1}` will be typeset as *latex2man*(1).
- `\Bar` is typeset as |.
- `\Bs` (BackSlash) is typeset as \.
- `\Tilde` is typeset as a ~.
- `\Dots` is typeset as ...
- `\Bullet` is typeset as •.
- `\setVersion{..}` set .. as version information.
- `\setVersionWord{..}` set .. for the word *Version:* in the footline. The default is `\setVersionWord{Version:}`.
- `\Version` returns the version information.
- `\setDate{..}` sets .. as date information.
- `\Date` returns the date information.
- `\Email{..}` use to mark an Email address:  
`\Email{Juergen.Vollmer@informatik-vollmer.de}` is typeset as:  
Juergen.Vollmer@informatik-vollmer.de.
- `\URL{..}` use to mark an URL: `\URL{http://www.foo.de/\Tilde vollmer}`  
is typeset as `http://www.foo.de/~vollmer`.

- \LatexManEnd** the input file is read and processed until reading end-of-file or **\LatexManEnd** (at the beginning of a line). L<sup>A</sup>T<sub>E</sub>X ignores this command.
- \Lbr**, **\Rbr** is typeset as [ and ] (these variants are needed only sometimes like in `item[FooBar\Lbr xx \Lbr]`. Usually [ ] will work.
- \LBr**, **\RBr** is typeset as { and } (these variants are needed when using { or } as arguments to macros.
- \Circum** is typeset as ^.
- \Percent** is typeset as %.
- \TEXbr** If processed with L<sup>A</sup>T<sub>E</sub>X causes a linebreak (i.e. is equivalent to `\`). In the output of *latex2man* this macro is ignored.
- \TEXIbr** If TexInfo output is generated, causes a linebreak (i.e. is equivalent to `\`), otherwise ignored.
- \MANbr** If Man-Page output is generated, causes a linebreak (i.e. is equivalent to `\`), otherwise ignored.
- \HTMLbr** If HTML output is generated, causes a linebreak (i.e. is equivalent to `\`), otherwise ignored.
- \medskip** An empty line.
- \SP** Produces some extra space, works also at the beginning of lines. The code of the second line looks like: `\SP abc \SP\SP xx\`:
- ```
abc    xx
abc    xx
abc    xx
```
- Note: Due to some “problems” with TexInfo, the lines starting with `\SP` have a leading . (dot) in the TexInfo output, see **-a<sub>char</sub>**.

## 6.5 Accepted Macros from the rcsinfo Package

- \rcsInfo \$Id ...\$** if the L<sup>A</sup>T<sub>E</sub>X package **rcsinfo** is used, this command is used to extract the date of the Man-page.
- \rcsInfoLongDate** if the L<sup>A</sup>T<sub>E</sub>X package **rcsinfo** is used, this command is used to typeset the date coded in the `$Id ...$` string.

## 6.6 Accepted L<sup>A</sup>T<sub>E</sub>X Macros

The following standard L<sup>A</sup>T<sub>E</sub>X commands are accepted:

- \section{..}** The **section** macro takes one argument: the name of the Man-page section. Each Man-page consists of several sections. Usually there are the following sections in a Man-page: *Name* (special handling as environment, c.f. above), *Synopsis*, *Description*, *Options*, *Files*, *See Also*, *Diagnostics*, *Return Values*, *Bugs*, *Author*, *version*, etc.

*Synopsis* must be the first section after the **Name** environment.

Note: Do not use L<sup>A</sup>T<sub>E</sub>X-macros in section names.

`\subsection{..}` works as well as  
`\subsubsection{..}` those.  
`\emph{..}` `\emph{example}` is typeset as *example*.  
`\textbf{..}` `\textbf{example}` is typeset as **example**.  
`\texttt{..}` `\texttt{example}` is typeset as `example`.  
`\underline{..}` `\underline{example}` is typeset as example of underline.  
`\date{..}` uses .. as date.  
`\verb+..+` but only + is allowed as delimiter.  
`$<$` is typeset as <.  
`$>$` is typeset as >.  
`$<=$` is typeset as <=.  
`$>=$` is typeset as >=.  
`$=$` is typeset as =.  
`$<>$` is typeset as <>.  
`$\ge$` is typeset as ≥.  
`$\le$` is typeset as ≤.  
`$\leftarrow$` is typeset as ←.  
`$\Leftarrow$` is typeset as ⇐.  
`$\rightarrow$` is typeset as →.  
`$\Rightarrow$` is typeset as ⇒.  
`\{` is typeset as {.  
`\}` is typeset as }.  
`\$` is typeset as \$.  
`\$` is typeset as \$, should be used inside macro arguments.  
`\_` is typeset as \_.  
`\&` is typeset as &.  
`\#` is typeset as #.  
`\%` is typeset as %.  
`\,` is typeset as smaller blank -- (between the two -)  
`\-` is used to mark hyphenation in a word.  
`\` is typeset as a linebreak or marks the end of a column in the **Table** environment.  
`\` (a `\` followed by a blank) is typeset as a blank, although it cannot be used at the beginning of a line to make indentation (see the `\SP` command).  
`~` is typeset as a blank.

`\copyright` is typeset as ©.

`\noindent`

`\hline` inside a `Table` environment.

`\item` inside a `itemize`, `enumerate`, or `description` environment.

`\today` 10th September 2005 (see also the `rctinfo` L<sup>A</sup>T<sub>E</sub>X package).

`\ss`, `\a`, ... `\ss` = ß, `\a` = ä, `\o` = ö, `\u` = ü, `\A` = Ä, `\O` = Ö, `\U` = Ü.

It is allowed to surround these macros in `{` and `}` in all places, even inside other macros, e.g.

```
\textbf{"a"o"u"A"O"U\ss}
\textbf{"a"}{"o"}{"u"}{"A"}{"O"}{"U"}{"ss"}
\textbf{äöüÄÖÜß}
```

äöüÄÖÜßäöüÄÖÜß äöüÄÖÜß

If these letters are used in their LATIN-1 8-bit coding, they are translated into the equivalent letter of the desired output format. E.g. Ä becomes `&Auml`; in HTML and `@A` in `texinfo`.

## 6.7 Conditional Text

*latex2man* preprocesses the L<sup>A</sup>T<sub>E</sub>X input to allow text to be used conditionally. A special sort of L<sup>A</sup>T<sub>E</sub>X comment is used for that purpose.

- `%% IF condition %%`
- `%% ELSE %%`
- `%% END-IF %%`

A line must contain only such a comment and nothing else. *condition* is a boolean expression containing “names” and operators. The names given with the `-Cname` option have the value “true”, while all other names occurring in the expression are assumed to be “false”. If the evaluation of the boolean expression results in the value “true”, the text in the “then”-part is used and the text in the optional “else”-part is skipped (and vice versa). The IF/ELSE/END-IF may be nested. As boolean operators the following are allowed:

||     boolean or  
 &&    boolean and  
 !     negation

( and ) for grouping are allowed.

For example:

```
%% IF abc %%
abc set
%% IF xyz %%
xyz set
```

```

%% ELSE %%
  xyz NOT set
%% END-IF %%
%% ELSE %%
  abc NOT set
%% IF xyz || !XYZ %%
  xyz OR !XYZ set
%% ELSE %%
  xyz OR !XYZ NOT set
%% END-IF %%
%% END-IF %%

```

Run this manual page through *latex2man* with e.g. **-C 'abc XYZ'** and have a look to the generated output. (If simply running the L<sup>A</sup>T<sub>E</sub>X-document through L<sup>A</sup>T<sub>E</sub>X, all lines are shown in the .dvi file).

```

abc set
xyz set
xyz NOT set
abc NOT set
xyz OR !XYZ set
xyz OR !XYZ NOT set

```

To check the conditional text feature, when *latex2man* is called with

**-CHTML** the lines 1a, 2b, 3b, and 4b;

**-CTEXI** the lines 1b, 2a, 3b, and 4b;

**-CMAN** the lines 1b, 2b, 3a, and 4b;

**-CLATEX** the lines 1b, 2b, 3b, and 4a;

**calling L<sup>A</sup>T<sub>E</sub>X without preprocessing** all lines

should be shown:

- 1a. This text occurs only when viewing the HTML output.
- 1b. The HTML conditional was not set.
- 2a. This text occurs only when viewing the TEXI output
- 2b. The TEXI conditional was not set.
- 3a. This text occurs only when viewing the MAN output
- 3b. The MAN conditional was not set.
- 4a. This text occurs only when viewing the LATEX output
- 4b. The LATEX conditional was not set.

## 6.8 Translation of User Defined Macros

The user macro translation file (given by the [-ttransfile]) contains *Perl* commands specifying the translation of L<sup>A</sup>T<sub>E</sub>X macros defined by the user. These macros may have none, one or two arguments. The following code is expected:

- Comments start with a # up to the end of the line.
- For a macro `\foo` with no arguments, the following code must be specified:

**Translation to Man-Pages**

```
$manMacro{'foo'} = '...';
```

**Translation to HTML**

```
$htmlMacro{'foo'} = '...';
```

**Translation to TexInfo**

```
$texiMacro{'foo'} = '...';
```

where ... is the translation.

- For a macro `\foo{..}` with one argument, the following code must be specified:

**Translation to Man-Pages**

```
$manMacro1a{'foo'} = '...';
```

```
$manMacro1b{'foo'} = '...';
```

**Translation to HTML**

```
$htmlMacro1a{'foo'} = '...';
```

```
$htmlMacro1b{'foo'} = '...';
```

**Translation to TexInfo**

```
$texiMacro1a{'foo'} = '...';
```

```
$texiMacro1b{'foo'} = '...';
```

where ... is the translation. The 1a code is used before the argument, while 1b is typeset after the argument is set.

- For a macro `\foo{..}{..}` with two arguments, the following code must be specified:

**Translation to Man-Pages**

```
$manMacro2a{'foo'} = '...';
```

```
$manMacro2b{'foo'} = '...';
```

```
$manMacro2c{'foo'} = '...';
```

**Translation to HTML**

```
$htmlMacro2a{'foo'} = '...';
```

```
$htmlMacro2b{'foo'} = '...';
```

```
$htmlMacro2c{'foo'} = '...';
```

**Translation to TexInfo**

```
$texiMacro2a{'foo'} = '...';
```

```
$texiMacro2b{'foo'} = '...';
```

```
$texiMacro2c{'foo'} = '...';
```

where ... is the translation. The 2a code is used before the first argument, 2b between the two arguments and 2c is typeset after the second argument is set.

- The file `latex2man.trans` contains some example code.

## 6.9 Verbatim Environment

```
This
    {is}
      \texttt{a}
        $test$
          _of_
verbatim
<this is no HTML tag> and no @* TexInfo command
```

## 6.10 Subsection works

This is a \subsection.

### 6.10.1 Subsubsection works

This is a \subsubsection.

### 6.10.2 Subsubsection still works

This is another \subsubsection.

## 6.11 General Remarks

1. Empty lines are typeset as paragraph separators.
2. The arguments of the L<sup>A</sup>T<sub>E</sub>X commands must not be split over several lines.
3. Do not nest calls to macros.
4. Except the mentioned environment and macros, the usage of other L<sup>A</sup>T<sub>E</sub>X environments or macros are not translated. Their usage will cause garbage in the output.
5. *latex2man* requires Perl version  $\geq 5.0004.03$ .
6. If you want to install the system with the distributed **Makefile**, you need GNU-*make*. If you don't have it, you should execute the steps shown in the **Makefile** manually.

## 6.12 Some Bug Fix Tests

**Leading . and '** Now leading . and ' in generation troff output should work properly, since a \& is added. Therefore the \Dot macro has been deleted.  
Thanks to Frank.Schilder@Mathematik.Tu-Ilmenau.De.  
Testcase 1:

```
'\n' ...
```

```
Testcase 2:
.foobar Testcase 3:
...
abc ...abc . efg ' 123
```

**% in verbatim** A % in a `\verb` and verbatim-environment was not emitted correctly. Thanks to Aleksey Nogin `nogin@cs.caltech.edu` for the bug report and bug fix.

```
% abc
```

```
% abc %
```

but ignore comments following this:

## 7 Requirements

**Perl** *latex2man* requires Perl version `>= 5.0004.03`.

**Make** If you want to install the system with the distributed **Makefile**, you need GNU-*make*. If you don't have it, you should execute the steps shown in the **Makefile** manually.

**L<sup>A</sup>T<sub>E</sub>X** L<sup>A</sup>T<sub>E</sub>X2e is required.

## 8 Changes

### Change history of the latex2man package

=====

```
$Id: CHANGES,v 1.52 2005/07/20 13:30:42 vollmer Exp $
1.18  20 July 2005    - latex2man.sty: Removed option pdf.
1.17  14 Oct  2004    - Fixed order of generating & for HTML
    (Thanks to Aleksey Nogin <nogin@cs.caltech.edu>)
1.16  06 Sep  2004    - Fixed handling of % in \verb+...+ and verbatim
    environments
    (Thanks to Aleksey Nogin <nogin@cs.caltech.edu>)
    - added \subsubsection
1.15  12 Dec  2001    - added extra stuff in the headline of man pages
    (thanks to Andrew Anderson <aja@emulab.ee.mu.oz.au>)
1.14  18 June 2001    - TROFF generation: fixed handling of leading . and '
    - removed macro \Dot
1.13  08 May  2001    - Added \begin{center} / \end{center}.
    - Reimplemented font changing stuff in TROFF
    generation (use now troff instead of -man commands).
1.12.3 10 Apr  2001   - Fixed \oOptArg in man output.
1.12.2 12 Mar  2001   - Fixed handling of conditional translation to LaTeX
```

- (if in a IF .. ELSE .. END-IF, the condition evaluates to true, the ELSE and END-IF are written to the result file, but not the IF...  
Now all %% directives are written.)
- 1.12.1 13 Sep 2000 - In \setDate and \setVersion a ~ is replaced by the output specific blank.
  - 1.12 28 Aug 2000 - Fixed error in htmlVerb (thanks to Glunz Wolfgang <Wolfgang.Glunz@icn.siemens.de>)
  - Fixed error in conditional text handling: (%%ELSE%% was not handled correct).
  - 1.11 25 Aug 2000 - Fixed error in conditional text handling: (MAN was not enabled by default, when calling with -M).
  - Added error message, if arguments of \begin{Name}, \section, and \subsection are not contained on a single line.
  - 1.10.1 21 Aug 2000 - Better handling of \Tilde.
  - 1.10 13 Jun 2000 - added \Begin{Description}[label] (based on an idea of Andi Winter <winter@cocolab.de>)
  - Thanks to Andi Winter <winter@cocolab.de>:
  - fixed: removed wrong line break in date for TROFF output
  - fixed wrong line breaks in some TROFF macros
  - fixed wrong output of optional arguments in TROFF
  - allow spaces between '\item' and following '['
  - 1.9.1 20 Mar 2000 - added \OptoArg \oOptoArg Option with optional argument (Thanks to Andi Winter <winter@cocolab.de>)
  - 1.9 13 Mar 2000 - latex2man.sty: Added option pdf.
  - latex2man: define names HTML TEXI MAN LATEX automatically.
  - 1.8 11 Feb 2000 - Added verbatim environment (many thanks to "Olaf Maibaum" <Maibaum@Informatik.Uni-Oldenburg.DE> who provided me the troff generation code)
  - Added better handling of german Umlaut
  - Fixed minor HTML tag mismatch.
  - 1.7.1 17 Nov 1999 - Fixed END\_IF handling of -L option.
  - 1.7 25 July 1999 - Allow german Umlaute to be written as \"a, etc.
  - 1.6.4 30 May 1999 - Added fix of Frank.Schilder@Mathematik.TU-Ilmenau.DE
- ```

line 418: from
        if ($manRS == 0 && $list_nest > 0) {
to
        if ($manRS == 0 && $list_nest > 1) {
line 471: from

        #    manParagraph;
to
        manParagraph;

```
- 1.6.3 25 Mar 1999 - Added \, Math stuff is matched now after macro arguments
  - 1.6.2 21 Nov 1998 - Added \LBr / \RBr, added \setlength{}{} and \itemsep

			<code>\Dollar \$\ge\$, \$\le\$, \$\leftarrow\$, \$\Leftarrow\$, \$\rightarrow\$, \$\Rightarrow\$.</code>
1.6.1	14 Sep	1998	- Added <code>\$&lt;=\$</code> , <code>\$&gt;=\$</code> , <code>\$=\$ \$&lt;&gt;\$</code> .
1.6	08 Sep	1998	- Added <code>\SP</code> macro. - Hyphenation <code>\-</code> is translated for TexInfo into <code>@-</code> . - Runs now under older Versions of Perl (e.g. 5.03).
1.5	22 Aug	1998	- Support for the <code>fanyhdr</code> package, added <code>latex2man.cfg</code> - Now <code>\%</code> and <code>\-</code> work.
1.4	23 July	1998	- Fixed typos. - Fixed generated HTML The following is wrong: <code>&lt;a ..&gt; &lt;h..&gt;...&lt;/h..&gt;&lt;/a&gt;</code> Correct is: <code>&lt;h..&gt;&lt;a ..&gt;...&lt;/a&gt;&lt;/h..&gt;</code>
1.3	03 June	1998	- Add conditional text feature
1.2	18 Jan	1998	- Enumerate/itemize/description environments may be nested now. - Output specific line break macros.
1.1	14 Jan	1998	- Fixed bug in HTML generation (no blanks emitted after <code>\hline</code> )
1.0	11 Jan	1998	- Initial version

Please check the file `latex2man-CHANGES` for the list of changes.

## 9 Version

Version: 1.18 of 10th September 2005.

## 10 License and Copyright

**Copyright** © 1998, Dr. Jürgen Vollmer, Viktoriastraße 15, D-76133 Karlsruhe, Germany, [Juergen.Vollmer@informatik-vollmer.de](mailto:Juergen.Vollmer@informatik-vollmer.de)

**License** This program can be redistributed and/or modified under the terms of the LaTeX Project Public License Distributed from CTAN archives in directory `macros/latex/base/lppl.txt`; either version 1 of the License, or any later version.

**Misc** If you find this software useful, please send me a postcard from the place where you are living.

## 11 Author

Dr. Jürgen Vollmer  
Viktoriastraße 15  
D-76133 Karlsruhe  
Email: [Juergen.Vollmer@informatik-vollmer.de](mailto:Juergen.Vollmer@informatik-vollmer.de)  
WWW: <http://www.informatik-vollmer.de>.