Special and common aspects of pdf/dvi/xdvi generators

Ernst Reissner (rei3ner@arcor.de)

Contents

List of Tables **1**

1 Introduction **1**

2 The options for TEX Live **2**

3 The options for MiKTeX **4**

4 Treatment of metadata for the PDF format **6**
4.1 Reproducibility 6
4.2 Security and Stability of Reproducibility 8
4.3 Reproducibility revisited 9
4.4 Security revisited 9

5 To be clarified **9**

6 References **9**

List of Tables

1 Converters and the version this document refers to 2
2 Options of TEX engines in TEX Live 3
3 Options of TEX engines in MiKTeX 6

1 Introduction

This document is created with `lualatex` or that like with output format PDF. The package `tex4ht` is not loaded.

This document is about features the three generators on latex, `pdflatex`, `lualatex` and `xelatex` have in common and discusses also aspects under which they are specific. These programs are just underlying tex engines preloading the the LATEX format. The names of the underlying engines just drop the inner sylable "`la`". For `pdftex` there is a user manual [THHB23], for `luatex` there is a reference manual [HHHS23] and for `xetex` a reference guide [RHB23].

The first aspect we cover are the options (among those to display the version). It turns out, that the options are specific for the distribution. Apart from TEX Live, there is a second important distribution, MiKTeX, which should also be treated.

We treat the options for TEX Live in Section 2 and for MiKTeX in Section 3.

The second subject is treatment of metadata available for PDF files. The motivation here is privacy as part of security and creation of reproducible documents, e.g. for tests. The results of our research is collected in Section 4.

2 The options for TEX Live

Note that in fact we use a variant of `luatex`, called `luahbtex`.

This document is valid for versions of the underlying tex engine as given in Table 1. Moreover, our research refers to a specific distribution, TEX Live.

converter	version
`pdflatex`	`pdfTeX 3.141592653-2.6-1.40.24`
`xelatex`	`XeTeX 3.141592653-2.6-0.999994`
`lualatex`	`LuaHBTeX, Version 1.15.0`

Table 1: Converters and the version this document refers to

We start with a synopsis of the options. Table 2 shows options of the converters under consideration. Note that in contrast to the other converters, `lualatex` defines options starting with `--` but it can also process the options if given with a single dash also. Conversely, converters other than `lualatex` can also deal with options starting with single dash. Options unknown to a converter never result in an error or even a warning; instead just a info message is displayed. This allows to create a configuration which works for all converters.

In Table 2 column "included", each converter is represented by the starting letter of its name, so for each option it is known which converters know about it an conversely, which options each converter has.

The table allows to furnish configurations working for all converters. Some options are common to all converters

option	included			explanation
`(-)-cnf-line=STRING`	p	x	l	parse `STRING` as a configuration file line
`--credits`	-	-	l	Display credits and exit.
`--debug-format`	-	-	l	enable format debugging
`(-)-draftmode`	p	-	l	switch on draft mode (generates no output PDF)
`-enc`	p	-	-	Enable encTeX extensions such as `\mubyte`
`-etex`	p	x	-	enable e-TeX extensions
`(-)-[no-]file-line-error`	p	x	l	disable/enable file:line:error style messages
`--[no-]file-line-error-style`	-	-	l	aliases of `--[no-]file-line-error`
`-fmt=FMTNAME`	p	x	l	use FMTNAME instead of program name or a %& line[1]
`(-)-ini`	p	x	l	for dumping formats
`-ipc`	p	-	-	send DVI output to a socket as well as the usual output file
`-ipc-start`	p	-	-	as `-ipc`, and also start the server at the other end
`(-)-halt-on-error`	p	x	l	stop processing at the first error
`(-)-help`	p	x	l	display this help and exit
`(-)-version`	p	x	l	output version information and exit
`-8bit`	p	x	-	make all characters printable by default[2]
`(-)-interaction=STRING`	p	x	l	set interaction mode (`STRING=batchmode/nonstopmode/scrollmode/errorstopmode`)

[1] in fact for `lualatex` the explanation deviates a bit: `--fmt=FORMAT: load the format file FORMAT`

[2] for `xelatex`: don't use ^^X sequences

(-)-jobname=STRING	p	x	l	set the job name to STRING
(-)-kpathsea-debug=NUMBER	p	x	l	set path searching debugging flags according to the bits of NUMBER
--lua=FILE	-	-	l	Load and execute a lua initialization script.
--luaonly	-	-	l	run a lua file, then exit
--luaconly	-	-	l	byte-compile a lua file, then exit
--luahashchars	-	-	l	the bits used by current Lua interpreter for strings hashing
(-)-[no-]mktex=FMT	p	x	l	disable/enable mktexFMT generation[3]
-mltex	p	x	-	enable MLTeX extensions such as \charsubdef
-no-pdf	-	x	-	generate XDV (extended DVI) output rather than PDF
--nosocket	-	-	l	Disable the Lua socket library.
(-)-output-comment=STRING	p	x	l	use STRING for DVI file comment instead of date (no effect for PDF)[4]
(-)-output-directory=DIR	p	x	l	use existing DIR as the directory to write files in
(-)-output-format=FORMAT	p	-	l	use FORMAT for job output; FORMAT is 'dvi' or 'pdf'[5]
-output-driver=CMD	-	x	-	use CMD as the XDV-to-PDF driver instead of xdvipdfmx
-papersize=STRING	-	x	-	set PDF media size to STRING
-[no-]parse-first-line	p	x	-	disable/enable parsing of first line of input file
(-)-progname=STRING	p	x	l	set program (and fmt) name to STRING[6]
(-)-recorder	p	x	l	enable filename recorder
--safer	-	-	l	Disable easily exploitable Lua commands.
(-)-[no-]shell-escape	p	x	l	disable/enable \write18SHELL COMMAND[7]
(-)-shell-restricted	p	x	l	enable restricted \write18[8]
-src-specials	p	x	-	insert source specials into the DVI file
-src-specials=WHERE	p	x	-	insert source specials in certain places of the DVI/XDV[9] file.
(-)-synctex=NUMBER	p	x	l	generate SyncTeX data for previewers[10]
-translate-file=TCXNAME	p			use the TCX file TCXNAME[11]
--utc			l	Init time to UTC

Table 2: Options of TEX engines in TEX Live

pdflatex:

```
Usage: pdftex [OPTION]... [TEXNAME[.tex]] [COMMANDS]
   or: pdftex [OPTION]... \FIRST-LINE
   or: pdftex [OPTION]... &FMT ARGS
  Run pdfTeX on TEXNAME, usually creating TEXNAME.pdf.
  Any remaining COMMANDS are processed as pdfTeX input, after TEXNAME is read.
  If the first line of TEXNAME is %&FMT, and FMT is an existing .fmt file,
  use it.  Else use `NAME.fmt', where NAME is the program invocation name,
  most commonly `pdftex'.

  Alternatively, if the first non-option argument begins with a backslash,
  interpret all non-option arguments as a line of pdfTeX input.

  Alternatively, if the first non-option argument begins with a &, the
```

[3] (FMT=tex/tfm/pk) for pdflatex; else (FMT=tex/tfm)
[4] For xelatex it is XDV instead of DVI and the remark (no effect for PDF) is missing
[5] xelatex offers option -no-pdf instead.
[6] lualatex does not mention (and fmt)
[7] For lualatex the explanation is disable/enable system commands
[8] For lualatex the explanation is restrict system commands to a list of commands given in texmf.cnf
[9] DVI for pdflatex; XDV for xelatex
[10] Explanation differs for lualatex
[11] TCX means TEX character translation

```
next word is taken as the FMT to read, overriding all else.  Any
remaining arguments are processed as above.

If no arguments or options are specified, prompt for input.
```

3 The options for MiKTeX

Since at the time of this writing, the author has no MiKTeX at hand, the results for MiKTeX are based on documentation, rather than experimentation. The three engines are a bit different, also in their names.

Well this section is preliminary only. It turned out that Section 2 is valid only for distribution TEX Live. So in this section we venture to find out the options for the other big distribution, MiKTeX. We shall also investigate whether there are further distributions.

Whereas the description [KB23] seems not to mention the options explicitly, the MiKTeX manual [Sch22] describes each program in Section II, 6, in particular also the tex converters. This is the source of the following tables.

The first observation is that, for MiKTeX all options start with two dashes, whereas for TEX Live this is the case only for luatex. One has to clarify, whether the maven latex plugin under consideration really works for MiKTeX.

option	included			explanation
`--alias=name`	p	x	l	Pretend to be program name, …[12]
`--aux-directory=dir`	p	x	l	Set `dir` as the directory to write auxiliary files to.
`--buf-size=n`	p	x	-	Set the the maximum number of characters …
`--c-style-errors`	p	x	l	Change the way, error messages are printed.
`--credits`	-	-	l	Display credits and exit[13].
`--disable-8bit-chars`	p	x	-	Make only 7-bit characters printable.
`--disable/enable-installer`	p	x	l	Disable/Enable automatic installation of packages.
`--disable-write18`	p	x	l	Disable the `\write18{command}` construct.
`--enable-write18`	p	x	l	Fully enable the `\write18{command}` construct[14].
`--restrict-write18`	p	x	l	Partially enable the `\write18command` construct.
`--debug-format`	-	-	l	Enable format debugging[15].
`--[dont-]parse-first-line`	p	x	-	[Dont p|P]arse first line of input file under definite conditions[16]
`--draftmode`	p	-	l	switch on draft mode (generates no output PDF)[17]
`--enable-8bit-chars`	p	x	-	Make all characters printable.
`--enable-enctex`	p	-	-	Enable encTeX extensions such as `\mubyte`[18].
`--enable-etex`	p	x	-	Enable eTeX extensions.
`--enable-installer`	p	x	l	Enable automatic installation of packages.
`--enable-mltex`	p	x	-	Enable MLTeX extensions such as `\charsubdef`.
`--error-line=n`	p	x	-	Set the width of context lines on …error messages.
`--extra-mem-bot=n`	p	x	-	Set the extra size … for large data structures …
`--extra-mem-top=n`	p	x	-	Set the extra size (in memory words) for chars, tokens, ….
`--font-max=n`	p	x	-	Set the maximum internal font number.
`--font-mem-size=n`	p	x	-	Set the size, in TeX memory words, of the font memory.
`--half-error-line=n`	p	x	-	Set the width of first lines of contexts in terminal error messages.
`--halt-on-error`	p	x	l	Quit after the first error.

[12] Using this option is equivalent to copying the program file to name and invoking name.
[13] The same as for TEX Live.
[14] Corresponds roughly to `(-)-shell-escape` in TEX Live.
[15] The same as for TEX Live.
[16] Similar for TEX Live. Note that there is also a converse option.
[17] Some differences in formulation between the converter and also between distributions
[18] Corresponds with `-enc` in TEX Live.

`--hash-extra=n`	p	x	-	Set the extra space for the hash table of control sequences …
`--help`	p	x	l	Give help and exitt[19]..
`--hhelp`	p	x	-	manual page in an HTML Help window[20]
`--include-directory=dir`	p	x	l	Add the directory `dir` to [those] to be searched for input files.
`--initialize`	p	x	l	Become the INI variant of the program.
`--interaction=mode`	p	x	l	Set the interaction mode (`mode=batchmode/nonstopmode/scrollmode/errorstopmode`).
`--job-name=name`	p	x	l	Set the name of the job (`\jobname`).
`--job-time=file`	p	x	-	Set the time-stamp of all output files equal to file's time-stamp.
`--lua=FILE`	-	-	l	load and execute a lua initialization script[21].
`--luaonly`	-	-	l	Start LuaTeX as a Lua interpreter[22].
`--luaconly`	-	-	l	byte-compile a lua file, then exit[23].
`--luahashchars`	-	-	l	the bits used by current Lua interpreter for strings hashing
`--main-memory=n`	p	x	-	Change the total size … of the main memory array.
`--max-in-open=n`	p	x	-	Set the maximum number of input files …
`--max-print-line=n`	p	x	-	Set the width of longest text lines output.
`--max-strings=n`	p	x	-	Set the maximum number of strings.
`--[no-]mktex=fmt`	-	-	l	Enable/Disable `fmt` generation, where `fmt` must be either `tex` or `tfm`.
`--nest-size=n`	p	x	-	Set the maximum number of semantic levels simultaneously active.
`--no-c-style-errors`	p	x	l	Don't change the way, error messages are printed.
`--no-pdf`	-	x	-	generate XDV (extended DVI) output rather than PDF
`--nosocket`	-	-	l	Disable the Lua socket library.
`--output-comment=string`	-	-	l	Use `string` for DVI file comment instead of date.
`--output-directory=dir`	p	x	l	Write output files in `dir`[24].
`--output-driver=CMD`	-	x	-	use CMD as the XDV-to-PDF driver instead of xdvipdfmx
`--output-format=format`	p	-	l	Use format for job output (one of: dvi, pdf)[25].
`--papersize=STRING`	-	x	-	set PDF media size to STRING
`--param-size=n`	p	x	-	Set the the maximum number of simultaneous macro parameters.
`--pool-free=n`	p	x	-	Set the minimum pool space left after loading the format.
`--pool-size=n`	p	x	-	Set the maximum number of characters in strings, …
`--quiet`	p	x	-	Suppress all output, except errors.
`--record-package-usages=file`	p	x	-	Record all package usages and write them into `file`.
`--recorder`	p	x	l	Enable the file name recorder[26].
`--safer`	-	-	l	Disable easily exploitable Lua commands[27].
`--save-size=n`	p	x	-	Set the the amount of space for saving values outside of current group.
`--src-specials`	p	x	-	Embed source file information in the DVI file[28].
`--stack-size=n`	p	x	-	Set the maximum number of simultaneous input sources.
`--string-vacancies=n`	p	x	-	Set the minimum number of characters …
`--synctex=n`	p	x	l	Generate SyncTeX data for previewers[29]
`--tcx=tcxname`	p	-	-	Use the tcxname translation table …
`--time-statistics`	p	x	-	Show processing time statistics.
`--trace[=tracestreams]`	p	x	-	Enable trace messages.
`--trie-size=n`	p	x	-	Set the amount of space for hyphenation patterns.
`--undump=name`	p	x	l	Use name as the name of the format to be used, …

[19]The same as for TEX Live.
[20]This option is only available on Windows systems.
[21]The same as for TEX Live.
[22]Could be the same as for TEX Live.
[23]Could be the same as for TEX Live.
[24]Similar as for TEX Live.
[25]`pdflatex` and `lualatex` differ a bit in text. Seems similar to TEX Live.
[26]The same as in TEX Live.
[27]The same as for TEX Live.
[28]Similar as in TEX Live.
[29]Explanation with more detail than for TEX Live.

`--utc`	-	-	l	Init time to UTC[30].
`--version`	p	x	l	Show version information and exit[31].

Table 3: Options of TEX engines in MiKTeX

Strange, there are `--enable-etex` and `--enable-mltex` but no way to disable. Maybe disable is the default.

```
  miktex-pdftex [option...] [[file] | [\command...]]

  miktex-luatex [option...] [[command...] | [file]]

The following options are ignored:

--8bit, --etex, --parse-first-line, --no-parse-first-line
These are always on.

--default-translate-file=tcxname, --translate-file=tcxname
These are always off.

  miktex-xetex [option...] [[file] | [\command...]]
```

4 Treatment of metadata for the PDF format

TBD: rework: required is a link to `headerSuppressMetaPDF.tex`. This is only possible after this is an injection. Also use references to [HHHS23] and to [THHB23].

Whereas `xelatex` always produces a XDV file internally as an intermediate step, when creating a PDF file, with the option `-no-pdf` one can eliminate creation of the PDF file and writing of the intermediate XDV file instead. For conversion, of the XDV file to PDF, the option `-output-driver=CMD` is used which defaults to the command `xdvipdfmx`. Besides direct creation of a PDF file, we consider creation via XDV file using `xdvipdfmx`. The XDV format is an extension and in fact a variant of the DVI format.

For the other compilers in contrast, the option `-output-format=dvi/pdf` determines the output format which is PDF by default and there is no intermediate format for PDF. When creating DVI files instead, these files can be converted into PDF by explicitly invoking something like `dvipdfmx`, `dvipdfmx` or `xdvipdfmx`. In my current distribution TEX Live, the programs `dvipdfm`, `dvipdfmx` and `xdvipdfmx` are all binary identical.

Nevertheless, they turn out to yield different results. One reason turning out later is, that the name with which the program is invoked goes into the result. It is likely that this is the only reason.

As a consequence of the workflow of `xelatex`, `\ifpdf` provided by package `iftex` always enters the `\else` branch for `xelatex`.

To display metadata, we use `exiftool` and `pdfinfo`.

4.1 Reproducibility

The first observation is, that invocation of `xelatex latexEngines` produces different PDF output for each run. Likewise, `xelatex -no-pdf latexEngines` produces different XDV output for each run. As turns out later, this is because the creation time goes into the result.

[30] The same as for TEX Live.
[31] The same as for TEX Live.

Thus, it is plausible that, to obtain reproducibility, we invoke the compiler as

```
SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1 xelatex          latexEngines
SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1 xelatex -no-pdf latexEngines
```

The second result is, that creating the PDF file and the XDV file that way is reproducible.

Now it is time to see differences directly and thus switch off compression which is done independent of the compiler by an according header

```
\DocumentMetadata{uncompress}
```

described in [MF23]. Seemingly, this writes additional info like metainfo date but also resurrects all other dates and also influences trailerid in a way, that the document is no longer reproducible, even when specifying `SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1`.

TBD: file a bug report. TBD: clarify whether this is true for other compilers also.

Thus, we take refuge to switching off compression via

```
\special{dvipdfmx:config z 0}
```

This is specific for `xelatex`, but it keeps reproducibility.

It turns out, that this compression setting does not refer to the XDV file, which seems always compressed, but solely to the PDF file. The PDF files differ mainly in the time stamp but also in some hashes which may depend on the time stamp.

Now let us experiment with `xdvipdfmx`. Even if we start with a reproducible XDV file, the PDF file created by `xdvipdfmx` changes with each invocation. This changes when also `xdvipdfmx` is invoked with fixed time.

```
SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1 xdvipdfmx latexEngines
```

As mentioned above, in the distribution TEX Live current at time of this writing, the programs `dvipdfm`, `dvipdfmx` and `xdvipdfmx` are all binary identical. Nevertheless, they seem to lead to different output. Possibly, the invocation name goes into the result. To find out, we do not allow compression. It turns out that the names go into the result as the producer.

Using the package `hyperref`, one can overwrite a lot of metadata. Details are found in the manual [RO22], Section 5.10. In particular, the producer can be set `unknown`. As a result, the trailer identifier is the only remaining difference. Seemingly, the producer goes into this whether displayed or not. The trailer identifier cannot be overwritten by `hyperref`, but only in a way specific for `xelatex`:

```
\special{pdf:trailerid [
  <00112233445566778899aabbccddeeff>
  <00112233445566778899aabbccddeeff>
]}
```

makes even the XDV to PDF converter transparent.

In [RO22], Section 5.10, also the creator is found, which is `LaTeX with hyperref` independent of the LATEX compiler. This shall be overwritten if there are security concerns.

As long as the tool chain and settings remain constant, invocation of latex compiler `xelatex`and backend XDV to PDF converter specifying `SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1` suffices to guarantee reproducibility. This setting refers to creation date `1970:01:01 01:00:00+01:00` and if the file exist, this is the modification date. From the point of view of reproducibility, there is ok, but it is not the truth. Thus, it makes sense to overwrite this with the string `unknown`.

If the file is overwritten, the same considerations apply to the modification date. Both can be overwritten with package `hyperref`.

One question remains: how does `hyperref` manipulate the metadata and in a second step, can we do this directly without using `hyperref`.

Now let us switch to the other two LaTeX compilers. Both write the banner information indicating above all the type of compiler and the version. As checked by switching compression of, `xelatex` does not write any banner information. Again, as long as the compiler does not change nor changes its version or its distribution, the banner does not corrupt reproducibility. On the other hand, removing it would stabilize and generalize reproducibility somewhat. Also, privacy or security is an argument in favor of The package `hyperref` offers no way to change the banner; this can be done only in a machine specific way. The details are described below. The same is true for the trailer identifier. The difference is that the trailer identifier must be explicitly overwritten to obtain reproducibility for some reason. Strictly speaking it need not be suppressed for reproducibility, but to make the result independent of the DVI to PDF converter.

Creating DVI files with `SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1` yields reproducible results. As expected, these settings are also necessary for translating DVI into PDF.

4.2 Security and Stability of Reproducibility

Security is here privacy. Hiding information makes attacks more difficult. Stability of reproducibility consists in stability as regards new versions of the same tools in the tool chain and the aspect of change of a tool.

From Section 4.1 come the recommendation to set the following pieces of information to unknown:

Creator This is uniformly `LaTeX with hyperref` as long as `hyperref` is loaded. If not it is TeX except for `xelatex`which shows compiler, and creation date. Thus, it is advisable in general for security but without `hyperref` for sake of stability of reproducibility.

Producer This is `xdvipdfmx` with version for creating DVIs and in general for `xelatex`. For creating PDF with `pdflatex` or with `lualatex`, it is something like `pdfTeX-1.40.25` or `LuaTeX-1.17.0`. This shall be hidden for sake of security and stability of reproducibility.

CreationDate This is wrong if using `SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1` and shall thus be hidden.

ModDate same as `CreationDate`.

PTEX.Fullbanner is not written by `xelatex`, but both for `lualatex`and for `pdflatex`. It can be suppressed in a compiler specific way, but not through `hyperref`. The banner exposes tools, versions and distributions. Thus, it shall not be exposed for sake of security and stability of reproducibility.

trailer identifier Suppressing this is necessary for reproducibility, except for `xelatex`. For `xelatex`, it is just independence of the XDV to PDF converter and thus a matter of stability of reproducibility. It has only a very weak aspect of security.

The downside is that, even without setting `SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1`, created PDF files stop to be individual for each creation run as is intended for PDF files. We decided that the date offers enough individuality to refrain from the trailer identifier.

4.3 Reproducibility revisited

The next question is, whether the setting `SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1` is really needed, if both `CreationDate` and `ModDate` are suppressed. If a document contains a date inserted via command `\date`, then this is only reproducible with setting `SOURCE_DATE_EPOCH=0 FORCE_SOURCE_DATE=1`. Even without `\date`, `xelatex`seems to use hashes depending on time corrupting reproducibility. To be more precise, creation of XDV is reproducible, whereas conversion to PDF is not. So, for sake of uniformity and stability of reproducibility, these settings are heavily recommended, both for the LaTeX compiler and for the converter DVI/XDV to PDF.

If reproducibility is only required for tests, then the date can be given; it is just neglected for tests. For productive documents the date is important. If reproducibility is an issue for productive documents, the date cannot be given.

4.4 Security revisited

Both `lualatex` and `pdflatex` write the filename in `PTEX.FileName`. This is a security issue, whereas the number of pages, `PTEX.PageNumber` is not. It is also useful.

For the `trapped` setting, the author wonders for `pdflatex`, how `hyperref` can switch this individually, whereas the compiler specific setting allows only setting to `unknown` in conjunction with other settings.

5 To be clarified

Note that `xelatex` works as the other engines for PDF, whereas it has XDV instead of DVI as alternative format. Thus, the format DVI is ignored. See the manual.

6 References

[HHHS23] H. Hagen, H. Henkel, T. Hoekwater, and L. Scarso. *LuaTEX Reference Manual*, 4 2023. A copy is within the documentation of this software.

[KB23] editor K. Berry. *The TeX Live Guide—2023*, 2 2023. A copy is within the documentation of this software.

[MF23] F. Mittelbach and U. Fischer. *The documentmetadata-support code*, 3 2023. A copy is within the documentation of this software, in fact two documents, documentmetadata-support-doc.pdf and documentmetadata-support-code.pdf which also comprises the implementation.

[RHB23] W. Robertson, K. Hosny, and K. Berry. *The XETEX reference guide*, 2 2023. A copy is within the documentation of this software.

[RO22] Sebastian Rahtz and Heiko Oberdiek. *Hypertext marks in LaTeX: a manual for hyperref*, 2 2022.

[Sch22] C. Schenk. MiKTeX Manual. `https://docs.miktex.org/manual/`, 2022.

[THHB23] Han The Thanh, H. Hagen, H. Henkel, and K. Berry. *The pdfTEX user manual*, 2 2023. A copy is within the documentation of this software.