

# vpp View and (selectively) Print PDF and PostScript

doc generated from the script with vpp

bash script, version=3.07

## Synopsis

vpp [options] [file]

## Options

-h,--help	print a help message and exit
-H,--Help	print print full documentation via less and exit
-V,--version	print version and exit
-b,--batch	run in batch using STRING for print command
--view	view the document (this is the default)
--noview	do not view the document
--viewer	specifies the pdf viewer to use
--print	offer printing interaction (this is the default)
--noprint	do not offer printing interaction
-p,--printer	print to printer named STRING
-d,--doublesided	printer is doublesided
-v,--verbose	be verbose
--noverbose	don't be verbose (this is the default)
-r,--rc	use STRING as an rc file
--norc	before handling the options, don't read the ~/.vpprc file

## Description

**vpp** is a Bash script that displays a PDF or PostScript document (after conversion to PDF) using `xpdf`, `gv`, or any other PDF viewer of your choice. The user can use the viewer to print the document or, alternatively, leave the viewer and use `vpp`'s facilities to print selected pages to a one- or two-sided hardcopy or an A5-booklet: see the section *Page selection* for the details. Instead of printing your selections, you can also save them into PDF files.

If `file` is specified with a `.ps` or a `.pdf` extension, **vpp** will simply use that `file`. Otherwise, **vpp** will look for `file.pdf`, `file.ps`, and `file`, in that order, and will use the first existing file. If `file` lacks, standard input is used.

In any case, the first few characters *in* the file determine whether it is treated as a PDF or as a PostScript file.

**vpp** has four possible exit values:

- 0 OK
- 1 error
- 2 edit, which is a signal to the calling program that a new edit session is at order; this is used by `mk`.
- 3 re-compile; this is used by `mk`

## Dependencies

<code>kpsewhich</code>	from <code>texlive</code>
<code>pdflatex</code>	from <code>texlive</code>
<code>pdfpages.sty</code>	from <code>texlive</code>
<code>pdfinfo</code>	from <code>poppler-utils</code>
<code>ps2pdf</code>	from <code>ghostscript</code>
<code>texi2dvi</code>	version 1.152 or greater, from <code>texinfo</code>
<code>mktemp</code>	from <code>coreutils</code>
<code>readlink</code>	from <code>coreutils</code>
<code>getopt</code>	from <code>util-linux</code>
<code>lpr</code>	from <code>cups-bsd</code>

```
lpoptions,lpstat from cups-client
file,less
```

## Options

**vpp** comes with several options. Before evaluating any options, **vpp** will try to read the user rc-file, `~/.vpprc`, where you can set defaults for most options, by assigning values to variable named after the long form of the options. For example:

```
printer=k550 doublesided=true
```

sets the printer to the printer named `k550` and tells that it can print `doublesided`. This is equivalent to calling **vpp** with:

```
--printer=k550 --doublesided
```

These are the variables that can be set in `~/.vpprc`:

```
batch      (string) sets the --batch option
print      (true or false) sets printing interaction on of off
printer    (string) sets the --printer option
doublesided (true or false) sets the --doublesided option
verbose    (true or false) sets the --verbose option
view       (true or false) sets viewing on or off
viewer     (string) set the viewer; arguments may be added; example:
           viewer='acroread -geometry 1450x1150+0+0'
           You should use a basename here, i.e. the name of the viewer should contain no slashes, and
           it should be in your PATH.
```

```
--help      Prints synopsis, then quits.
--Help      Prints this documentation, via less.
--version   Prints version, then quits.
--verbose   Prints messages about the progress vpp is making. Can be reverted with --noverbose.
--rc=rc-file Read the specified rc-file before processing, but after any other rc files. The contents
           of this rc-file will override previously specified options, but they will overridden in turn
           by options following it.
--norc      prevents reading the user rc-file.
--batch=string Prevents the -print option to interrogate the user about pages to be printed. Instead
           the document is printed according to the mandatory string. Also sets viewing off. Thus
           the command
           vpp --batch '2-3 x3' test.pdf
           prints 3 copies of pages 2 and 3 of test.pdf without viewing.
--print     Present the print prompt. This is the default. Can be reverted with --noprint, normally
           used to suppress the print prompt, for example when using vpp from other scripts that
           generate PDF or PostScript documents that have only to be displayed or printed without
           even being displayed.
--view      Run the file viewer. This is the default. Can be reverted with --noview, normally used
           to suppress starting the viewer, for example when using vpp from other scripts that
           generate PDF or PostScript documents that have only to be printed.
--printer=key Specifies the printer to be used instead of the system default printer. This script defines
           an associative array containing no printers at all, so by default, the system defined
           printer is used, and it is supposed to have no doublesided facilities, see the
           --doublesided option. You can however define your own set of printers in the ~/.vpprc
           file, by re-defining the variable printers, using the names of the printers as keys, and
           the corresponding values as true of false, depending on whether the printer can print
           doublesided on not. For example, if you have a doublesided printer named color and a
           singlesided printer called bw, you could defined the printers variable as follows:
           printers=([bw]=false [color]=true)
           printer=color
           Of course, the printers named bw and color must be known to your system.
--doublesided Tells that the printer is able, and configured, to do doublesided printing.
--viewer=key Specifies the viewer to use. This script defines an associative array viewers containing
           4 viewers as follows:
```

```

viewers=( [xp]=xpdf [ev]=evince [gv]=gv [ac]=acroread)
and the viewer is set to xp by default. However, you can define your own set of viewers
in the ~/.vpprc file; for example:
viewers=(
[xp]="xpdf -g 970x1050+0+0 -font 8x13bold -z page -cont"
[ac]="acroread -geometry 850x890+0+0"
[ev]="evince --fullscreen --presentation"
)
viewer=xp

```

## Page selection

When you select the `--print` option, and you did not also use the `--batch` option, **vpp** interrogates you about the pages you want to print. To that end the following prompt appears:

```
vpp command (? for help):
```

upon typing `?` or `h`, **vpp** displays examples of possible commands:

Command Examples:

```

5          to print page 5
5-         to print pages 5 through the end
5-7       to print pages 5, 6 and 7
7-5 ox    write the same pages, in reversed order, to x.pdf
-7        to print the first 7 pages
5-7,19-   to print pages 5, 6, 7 and 19 through the end
a         to print the whole document
-         to print the whole document
a x3      to print 3 copies of the document
x3        the same
5 x3      to print 3 copies of page 5
t         print the whole document twosided
t 2-      print twosided starting at page 2
b         to print the whole document as an a5 size booklet
b -12     to print the first 12 pages as an a5 size booklet

```

Other commands:

```

e         (if called by mk) edit the tex source and rerun mk
c         (if called by mk) rerun mk
v         (re)view the ps/pdf file
w         list errors and warnings from the log file
oxyz     send pdf output to file xyz.pdf instead of printer
pxyz     print to printer xyz
dx       tell vpp printer is doublesided (x=t) or singlesided (x=f)
h        display this help
?        display this help
q        quit

```

With these descriptions, no further explanation should be necessary, except for the following:

When twosided (`t`) or booklet (`b`) printing is selected, printing will be performed in two shifts, one for the front side and one for the backside. Between the shifts, another prompt appears:

```
printer ready? then turn stack and type return
```

You will have to arrange your printer such that, with the printed sides up, the first page printed will be at the bottom of the stack, and the last page printed will be on top. Normally you will then have your output come out the back of your printer. *Turn the stack* then means: rotate it over the long side of the paper and feed it back into the printer for the other side to be printed.

When you use the `oxyz` subcommand, your selection will not be printed but instead will be saved in a PDF file named `xyz.pdf`. When you use a `t` or `b` selection, you will not, of course, be prompted to turn the paper stack. Instead, the odd and even pages of your selection will be saved in separate PDF files, `xyz_odd.pdf` and `xyz_even.pdf`.

## Environment

Two environment variables may be useful in scripts using vpp:

VPPOUTDIR	The directory where PDF files generated with the o command will be saved; the default is the working directory.
VPPCHECKSAVED	If non-empty, <b>vpp</b> will check on exit that the inspected file has been saved into a pdf file and will issue a warning if it hasn't.

## Examples

Since **vpp** can read from standard input, it can be used to print (parts of) manpages. This example (we assume a printer which cannot print double sided) prints the full `ls` manpage first, followed by an A5 booklet of the first 8 pages:

```
$ man -t ls | vpp # (xpdf shows preview and is left with q)
vpp command (? for help): a
vpp command (? for help): b 1-8
printer ready? then turn pack over the long side and type enter (^D skips)
vpp command (? for help): q
$
```

If you don't need a preview, because you have seen the man page already, you can print it immediately as an A5 booklet with:

```
$ man -t ls | vpp --batch=b
```

or, to make an A5 booklet of the first 8 pages:

```
$ man -t ls |vpp --batch='-8 b'
```

If you just want to save a PDF copy of the man page, you can say:

```
$ man -t ls |vpp -b ols
```

Some PDF-documents, like the CVS manual (`cvs.pdf`), have their Table of Contents in their back instead of behind the title page. You can use **vpp** to rearrange such documents:

```
$ vpp --batch='1,2,153-160,3-152 ocv's' cvs.pdf
```

This overwrites the input document. Note that any links in the file will get broken, so that is only useful for documents that have to be printed. It would have been more sensible in this case to say:

```
$ vpp --batch=b 1,2,153-160,3-152' cvs
```

which prints the reordered document as an A5 booklet without replacing it. You can even print or output page ranges in reverse order:

```
$ vpp --batch='12-1 otest' cvs.pdf
```

## Author

Wybo Dekker

## Copyright

Released under the [GNU General Public License](#)

## Functions used:

### check\_needs

---

parameters:	-
description:	Verify the availability of executables and tex files
globals set:	-
globals used:	neededex neededtx
returns:	1 if something is missing, 0 otherwise

---

## find\_viewer

parameters: -  
description: Find pdf viewer  
globals set: viewer  
globals used: viewer viewers  
returns: 0

---

## handle\_options

parameters: the script's arguments  
description: Handles the options  
globals set: batch doublesided input mk print printer rc verbose view viewer writeto  
globals used: rc  
returns: 1 on error, 0 otherwise

---

## find\_pdf

parameters: -  
description: Find the input and provide a pdf-copy; If **vpp** had no file argument, standard input is used. If the argument has one of the extensions .pdf, .ps or .eps, or any uppercase variant, that file is used. Any other argument is used as such, if the file exists or, if not, a .pdf, PDF, PS, .ps, .eps or .EPS extension is added and the first existing file is used.  
globals set: log tempdir  
globals used: input  
returns: 1 if no input is found, 0 otherwise.

---

## pdfproperties

parameters: -  
description: Find page width, page height and the number of pages in the input file  
globals set: height pagecount width  
globals used: height pagecount width  
returns: 0

---

## ask

parameters: -  
description: Prompt for a command, return the command in com  
globals set: com  
globals used: com prompt  
returns: 0

---

## printhelp

parameters: -  
description: Print help for vpp-commands and show which viewer and printer are active.  
globals set: -  
globals used: Com Nor doublesided mk viewer viewers  
returns: 0

---

## ask\_selection

parameters: zero to many user commands  
description: Interact with user, specifying pages to be printed or exported as pdf, or to re-view the pdf or (if called from mk) re-edit the tex-source. If called with arguments (caused by vpp's --batch option) executes those.  
globals set: booklet com doublesided lpropt output output printer saved selection twosided viewer  
globals used: Err Nor VPPCHECKSAVED War booklet com compileexit editexit output pagecount printers saved selection twosided viewers  
returns: 0

---

## wait\_for\_printer

parameters: -  
description: Wait for user typing `enter`, signalling that the printer is ready for next job. `^D` instead skips further output.  
globals set: -  
globals used: -  
returns: 0

---

## printout

parameters: -  
description: Print selected pages or output them to pdf. Calls `doselection` for the actual output.  
globals set: selection  
globals used: writeto batch booklet doublesided output selection twosided  
returns: 0

---

## read\_rc

parameters: -  
description: If `--rc` was used, source its argument; otherwise, execute `~/.{myname}rc` if it exists  
globals set: -  
globals used: HOME myname rc  
returns: 0

---

## doselection

parameters: 1: (empty) if all pages of the selection are to be printed, "odd" if only the odd pages, "even" if only the even pages to be printed  
description: Make a selection of pdf pages and print it or output it to pdf file.  
globals set: selection  
globals used: Err Nor War booklet height lpr lpropt output printer selection verbose width  
returns: 0

---

## check\_printers

parameters: -

`description:` Check if any printers are installed. If not, warn the user. If there are system printers, and the user has defined printers in his rc file, verify that those are known to the system. If he did not, use the system-known printers, single-sided. If the user defined a default printer in his rc file, check it similarly. If OK, set it with `lptions`, silently. If the rc file sets no default printer, use the system default; if there is no sytem default set it to the first available printer, and warn the user.

`globals set:` `printers printer doublesided`

`globals used:` `printers printer`

`returns:` -