

NAME

lessfile, lesspipe – "input preprocessor" for less.

SYNOPSIS

lessfile, lesspipe

DESCRIPTION

This manual page documents briefly the *lessfile*, and *lesspipe* commands. This manual page was written for the Debian GNU/Linux distribution because the input preprocessor scripts are provided by Debian GNU/Linux and are not part of the original program.

lessfile and *lesspipe* are programs that can be used to modify the way the contents of a file are displayed in *less*. What this means is that *less* can automatically open up tar files, uncompress gzipped files, and even display something reasonable for graphics files.

lesspipe will toss the contents/info on STDOUT and *less* will read them as they come across. This means that you do not have to wait for the decoding to finish before less shows you the file. This also means that you will get a 'byte N' instead of an N% as your file position. You can seek to the end and back to get the N% but that means you have to wait for the pipe to finish.

lessfile will toss the contents/info on a file which *less* will then read. After you are done, *lessfile* will then delete the file. This means that the process has to finish before you see it, but you get nice percentages (N%) up front.

USAGE

Just put one of the following two commands in your login script (e.g. ~/.bash_profile):

```
eval "$(lessfile)"
```

or

```
eval "$(lesspipe)"
```

FILE TYPE RECOGNITION

File types are recognized by their extensions. This is a list of currently supported extensions (grouped by the programs that handle them):

```
*.a
*.arj
*.tar.bz2
*.bz
*.bz2
*.deb, *.udeb, *.ddeb
*.doc
*.egg
*.gif, *.jpeg, *.jpg, *.pcd, *.png, *.tga, *.tiff, *.tif
*.iso, *.raw, *.bin
*.lha, *.lzh
*.tar.lz, *.tlz
*.lz
*.7z
*.pdf
*.rar, *.r[0-9][0-9]
*.rpm
*.tar.gz, *.tgz, *.tar.z, *.tar.dz
*.gz, *.z, *.dz
*.tar
*.tar.xz, *.xz
*.whl
*.jar, *.war, *.xpi, *.zip
```

```
*.zoo
*.tar.zst, *.tztst
*.zst
```

USER DEFINED FILTERS

It is possible to extend and overwrite the default *lesspipe* and *lessfile* input processor if you have specialized requirements. Create an executable program with the name *.lessfilter* and put it into your home directory. This can be a shell script or a binary program.

It is important that this program returns the correct exit code: return 0 if your filter handles the input, return 1 if the standard *lesspipe/lessfile* filter should handle the input.

Here is an example script:

```
#!/bin/sh

case "$1" in
  *.extension)
    extension-handler "$1"
    ;;
  *)
    # We don't handle this format.
    exit 1
esac

# No further processing by lesspipe necessary
exit 0
```

FILES

~/lessfilter

Executable file that can do user defined processing. See section USER DEFINED FILTERS for more information.

BUGS

Sometimes, *less* does not display the contents file you want to view but output that is produced by your login scripts (*~/bashrc* or *~/bash_profile*). This happens because *less* uses your current shell to run the *lesspipe* filter. Bash first looks for the variable *\$BASH_ENV* in the environment expands its value and uses the expanded value as the name of a file to read and execute. If this file produces any output *less* will display this. A way to solve this problem is to put the following lines on the top of your login script that produces output:

```
if [ -z "$PS1" ]; then
  exit
fi
```

This tests whether the prompt variable *\$PS1* is set and if it isn't (which is the case for non-interactive shells) it will exit the script.

SEE ALSO

less(1)

AUTHOR

This manual page was written by Thomas Schoepf <schoepf@debian.org>, for the Debian GNU/Linux system (but may be used by others). Most of the text was copied from a description written by Darren Stalder <torin@daft.com>.