

mkdox

0.2

Rob la Lau rob@nerdstock.org
last updated 2009-12-02

Table of Contents

1	Introduction	2
2	Creating Texinfo files	3
2.1	Texinfo for advanced users	3
2.2	Texinfo for new users.....	3
3	Invoking mkdox	4
4	Using gmake directly	6
5	Configuration files	7
6	Dependencies	8
6.1	Required dependencies	8
6.2	Optional dependencies	8
7	Adding formats	9
Appendix A	Troubleshooting	10
A.1	Troubleshooting ps	10
Appendix B	License	11
Index		12

Table of Contents

Copyright © 2008-2009 Nerdstock.org

mkdox is a tool (or actually a set of Makefiles and a shell script) to convert Texinfo documents into many other formats.

The formats supported by mkdox depend on the apps installed.

Please note that this manual is not in sync with the current state of mkdox. On the one hand it is used as kind of a functional design, while on the other hand it is modified as mkdox grows.

The tool and it's documentation will reflect one and other more and more while they mature.

1 Introduction

mkdox generates many formats from one set of Texinfo files.

Well, actually not. . .

mkdox is a wrapper around many commands that can generate other formats from Texinfo files, or that can generate other formats from other formats that were generated from Texinfo files, etcetera.

This wrapper consists of a few Makefiles that tell **gmake** what to do. For ease of use there is the `mkdox` script, which is actually just a wrapper around the wrapper. (And if you'd like to have it even cooler, you could create a config file for your project (See Chapter 5 [Configuration files], page 7.) and then create an alias like `alias myappdocs='/usr/local/bin/mkdox -f ~/.mkdox/myapp.config'` to have a wrapper around the wrapper around the wrapper. . . ;))

mkdox was initiated when we wanted to distribute the Nerdstock.org manuals both as HTML and PDF. And then things just got out of control. . .

So, if you ever feel you need to distribute your documentation (or other writings) in JPEG format, mkdox is the tool for you.

But it may also come in handy at other occasions.

mkdox is distributed under the terms of version 2 of the GPL (See Appendix B [License], page 11.).

The original Texinfo files are copied to a working directory where they are processed, to make sure the originals are never altered.

Have fun!

Nerdstock.org

2 Creating Texinfo files

If you've created Texinfo files before, you probably wouldn't want to read the entire HOWTO.

On the other hand, if you're new to Texinfo, a few quick pointers may not be enough.

So, there's a chapter for the advanced user and a chapter for the beginning user.

2.1 Texinfo for advanced users

At Nerdstock.org, when we write Texinfo files, we create a file for each node, creating directories as needed to reflect the tree structure of the info file to be created. But we can't expect you to completely restructure your Texinfo tree, just because you'd like to try out `mkdtx`.

There are a few changes you should make, though.

If your current Texinfo files are located in `/path/to/MyApp/doc/`, this is the structure `mkdtx` expects to see:

```
/path/to/MyApp/doc/
/path/to/MyApp/doc/vars.mk
/path/to/MyApp/doc/texinfo/
/path/to/MyApp/doc/texinfo/top.texinfo
```

If you're going to distribute other formats (HTML, plaintext, etc.) along with the Texinfo files, you may wish to make that

```
/path/to/MyApp/doc/texinfo/
/path/to/MyApp/doc/texinfo/vars.mk
/path/to/MyApp/doc/texinfo/texinfo/
/path/to/MyApp/doc/texinfo/texinfo/top.texinfo
```

For this manual, however, I will pretend you're using the first structure.

The file `'top.texinfo'` starts with the Top node; everything before that is generated by `mkdtx` from variables in `'vars.mk'`. The generated file, which will be called something like `'/path/to/MyApp/doc/myapp.texinfo'`, will include a line that says

```
@include texinfo/top.texinfo
```

If you don't want to use the file which is generated as default, see the chapter on config files (See Chapter 5 [Configuration files], page 7.) or the `'-f'` switch for the `mkdtx` script (See Chapter 3 [Invoking `mkdtx`], page 4.).

The file `'vars.mk'` is included by the `'Makefile'` that is used to generate the documentation, so it should follow GNU Make (http://www.gnu.org/software/make/manual/html_node/index.html) guidelines. For an example of `'vars.mk'`, see the `'doc'` directory included in this package.

2.2 Texinfo for new users

TODO: Write a tutorial creating a Texinfo documentation project from scratch, fit for `mkdtx`.

3 Invoking mkdox

The `mkdox` script is a wrapper around `gmake`; all it does is build a commandline and execute it.

These are the available options.

- '-a file' Use additional config file; may be used in conjunction with '-e' and '-x'.
- '-c' Clean the working directory, leaving the generated files in place; mutually exclusive with '-m' and '-r'.
(make clean)
- '-d directory' Destination directory; must exist and be writable; mutually exclusive with '-i'.
(make DEST=directory)
- '-e' Ignore '/usr/local/etc/mkdox/config.mk'; may be used in conjunction with '-a' and '-x'.
- '-f "format ..."' Space separated list of desired formats.
(make FORMATS="format ...")
- '-h' Display help.
(make help)
- '-i' Install into target system's default locations; mutually exclusive with '-d'.
(make install)
- '-l' List available formats.
(make help-formats)
- '-m' Clean the working directory, leaving the generated files and the Texinfo files in place; mutually exclusive with '-c' and '-r'.
(make mostlyclean)
- '-n' Assist in the creation of a new project.
- '-p "project ..."' Space separated list of projects; the default is to process all projects in the source directory.
(make PROJECTS="project ...")
- '-q' Be quiet (don't echo any commands).
(make -s)
- '-r' Delete the working directory; mutually exclusive with '-c' and '-m'.
(make realclean)
- '-s directory' Source directory; must exist and be readable.
(make SRC=directory)
- '-u' Uninstall.
(make uninstall)

Invoking mkdox

- '-v' Display version info.
- '-x' Ignore '~/.mkdox/config.mk'; may be used in conjunction with '-a' and '-e'.
- '-y' Add a reference to the mkdox website (optional, but appreciated).
- '-z' Build the mkdox documentation.

4 Using gmake directly

All the `mkdox` script does, is create a `gmake` command.

If you feel confident enough, you can also call `gmake` directly. These are the targets you can use, along with their meanings.

TODO: Say something about the environment vars that are used.

- all** Generate the files in the working directory.
Needs 'SRC' to be set.
This is the default target.
(`mkdox -s directory`)

- install** Install the generated files onto the system.
If 'DEST' is set, files will be installed there (`mkdox -d directory`), otherwise files will be installed in the system-dependant default directories (`mkdox -i`).

- mostlyclean**
Clean the working directory, leaving the generated files and the Texinfo files in place.
(`mkdox -m`)

- clean** Clean the working directory, leaving the generated files in place.
(`mkdox -c`)

- realclean**
Delete the working directory.
(`mkdox -r`)

- uninstall**
Remove the generated files from the system.
(`mkdox -u`)

- help** Display (out of date) help.
(`mkdox -h`)

- help-formats**
Display a list of supported formats (a subset of the target above, but up-to-date).
(`mkdox -l`)

- newproject**
Assist in the creation of a new project.
(`mkdox -n`)

5 Configuration files

Someday, this page will tell you about how to configure mkdcox using files in `‘/usr/local/etc/mkdcox/’`, `‘~/mkdcox/’` and additional config files.

But while this is not yet implemented, this is currently not a priority.

6 Dependencies

Since mkdox doesn't generate anything by itself, it depends on a lot of other applications and commands. These can be divided in required and optional dependencies.

6.1 Required dependencies

Although the formats mkdox can generate, depend on the applications installed, there are some commands that mkdox needs to function.

The first is `gmake` (GNU Make).

The `make` command on Linux usually defaults to `gmake`. To make sure, execute `make -v`. On BSD systems the `make` command usually defaults to `pmake` (BSD Make). To install GNU Make on a BSD system, go to `/usr/ports/devel/gmake/` and execute `make install clean`.

The `mkdox` script will exit with a warning if `gmake` is not found.

The other commands mkdox needs, are installed on any Unix-like system:

- `cat`
- `cp`
- `echo`
- `mkdir`
- `mv`
- `rm`
- `rmdir`
- `sed`

6.2 Optional dependencies

To support all formats mkdox currently supports, these commands should be present:

- `a2ps`¹
- `html2text`
- `makeinfo`
- `pdftoppm`
- `pnmt/jpeg`
- `ps2pdf12`
- `ps2pdf13`
- `ps2pdf14`
- `texi2dvi`
- `texi2html`
- `texi2pdf`

¹ See Appendix A [Troubleshooting], page 10, if your ps or pdf output looks just like the raw Texinfo file

7 Adding formats

In short:

1. Add lines to `'shared/vars.mk'` to find the command that is used to generate the format.
2. Add lines to generate the format to `'shared/Makefile.docs'`.
3. If it works: mail those lines to `mkdox@nerdstock.org`.

In a future version of `mkdox`, `'shared/Makefile.docs'` will be split into separate files for each format.

Appendix A Troubleshooting

This page lists some tips for when things do not look as expected.

A.1 Troubleshooting ps

If the ps file (or any file that depends on it, like pdf) looks just like the raw Texinfo file(s), it may be that a2ps is not configured to delegate Texinfo to texi2dvi. Create a file `~/a2ps/a2psrc` with these lines:

```
Delegation: texi2dvi texinfo:ps \
  texi2dvi -e -o #f0 -l texinfo '$f' && dvips -f #f0 | psnup
  and try again.
```

If `gmake` then exits with an error, it may be because `tex` is not installed. Type `which tex` to check this, and if it's not installed, install it; on a BSD system, this is done by typing `make install clean` in `/usr/ports/print/teTeX-base` (this might take some time, as it will install quite some dependencies).

And if things still don't work after that (`dvips: not found`), install `dvips` (`/usr/ports/print/dvipsk-tetex`).

I guess the above will not be needed on a workstation, since it will probably have all this installed already.

Appendix B License

TODO: Display GPL here.

Index

-	
-a file.....	4
-c.....	4
-d directory.....	4
-e.....	4
-f "format ..".....	4
-h.....	4
-i.....	4
-l.....	4
-m.....	4
-n.....	4
-p "project ..".....	4
-q.....	4
-r.....	4
-s directory.....	4
-u.....	4
-v.....	5
-x.....	5
-y.....	5
-z.....	5
.	
.mkdox.....	7
/	
/etc/mkdox.....	7
/usr/local/etc/mkdox.....	7
~	
~/mkdox.....	7
A	
a2ps.....	8
all, gmake.....	6
all, make.....	6
Assistants.....	8
B	
BSD Make.....	8
C	
cat.....	8
clean, gmake.....	6
clean, make.....	6
Commandline options.....	4
Commands.....	8
Configuration files.....	7
cp.....	8
D	
Dependencies, optional.....	8
Dependencies, required.....	8
Document structure.....	3
E	
echo.....	8
F	
File format.....	3
Files, configuration.....	7
Flags, commandline.....	4
Format, file.....	3
G	
General Public License.....	11
gmake.....	6, 8
gmake all.....	6
gmake clean.....	6
gmake goals.....	6
gmake help.....	6
gmake help-formats.....	6
gmake install.....	6
gmake mostlyclean.....	6
gmake newproject.....	6
gmake realclean.....	6
gmake targets.....	6
gmake uninstall.....	6
GNU General Public License.....	11
GNU Make.....	8
Goals, gmake.....	6
Goals, make.....	6
GPL.....	11
H	
help, gmake.....	6
help, make.....	6
help-formats, gmake.....	6
help-formats, make.....	6
Helpers.....	8
html2text.....	8
I	
install, gmake.....	6
install, make.....	6

Index

L

Licence..... 11

M

make..... 6, 8
make all..... 6
make clean..... 6
make goals..... 6
make help..... 6
make help-formats..... 6
make install..... 6
make mostlyclean..... 6
make newproject..... 6
make realclean..... 6
make targets..... 6
make uninstall..... 6
makeinfo..... 8
mkdir..... 8
mkdix..... 4
mkdix -a..... 4
mkdix -c..... 4
mkdix -d..... 4
mkdix -e..... 4
mkdix -f..... 4
mkdix -h..... 4
mkdix -i..... 4
mkdix -l..... 4
mkdix -m..... 4
mkdix -n..... 4
mkdix -p..... 4
mkdix -q..... 4
mkdix -r..... 4
mkdix -s..... 4
mkdix -u..... 4
mkdix -v..... 4
mkdix -x..... 4
mkdix -y..... 4
mkdix -z..... 4
mostlyclean, gmake..... 6
mostlyclean, make..... 6
mv..... 8

N

New formats..... 9
newproject, gmake..... 6
newproject, make..... 6
Not the expected output..... 10

O

Optional dependencies..... 8
Options, commandline..... 4

P

pdf, troubleshooting..... 10
pdftoppm..... 8
pmake..... 8
pnmtojpeg..... 8
ps, troubleshooting..... 10
ps2pdf12..... 8
ps2pdf13..... 8
ps2pdf14..... 8

R

realclean, gmake..... 6
realclean, make..... 6
Required dependencies..... 8
Restructure Texinfo tree..... 3
rm..... 8
rmdir..... 8

S

Script, mkdix..... 4
sed..... 8
Settings..... 7
Structure, documents..... 3
Supporting format X..... 9
Switches, commandline..... 4

T

Targets, gmake..... 6
Targets, make..... 6
texi2dvi..... 8
texi2html..... 8
texi2pdf..... 8
Texinfo..... 3
Things look strange..... 10
Troubleshooting..... 10
Troubleshooting pdf..... 10
Troubleshooting ps..... 10
Tutorial..... 3

U

uninstall, gmake..... 6
uninstall, make..... 6

V

vars.mk..... 3

W

Warranty..... 11

mkdox 0.2

mkdox documentation generator (<http://nerdstock.org/mkdox>)