

Chapter 1

Documenting a makefile for this manual

This chapter will show you how to document (in \LaTeX) a makefile for this manual. While \LaTeX may look intimidating if you haven't used it before, it is fairly straightforward to use. We have created a number of tools to help you insert your makefile into this manual.

Makefile to \LaTeX

There are a number of things that need to be changed in a bare makefile in order to import it to the \LaTeX structure used for this manual. To do the basic stuff, call the script `makeToTex`¹ on your makefile(s). It can accept any number makefiles and outputs `<file>_texed.txt`. It has two flags, `h` and `v`, which call the help menu and verbose mode, respectively.

There are two main things that need to happen: certain symbols need to be escaped in \LaTeX , and recipes need to be formatted in a macro to make them pretty. Symbols like “\$” and “#” have special functions in \LaTeX , and will result in catastrophe if you don't fix them.

The functionality of the script is documented in [Appendix A](#). Please refer to it if you have any troubles.

Now to bring your new makefile into a \LaTeX document!

¹Please report bugs to Trevor at tkmday@uw.edu.

Editing a .tex File

Any script compiled with the main document will include the necessary environment `makefileread`. In your \LaTeX document, create a newline, ensuring the indentation is consistent and insert the command `\begin{makefileread}`. If you are using a \LaTeX editor, it may autocomplete with the last line, `\end{makefileread}`. If not, you'll have to type it yourself.

Paste your code between those two commands. It should be one level more indented.

To insert text comments between sections of code, you may terminate the environment, add your text and begin a new environment with the rest of the code. Refer to the style guide ([Appendix B](#)) for stylistic choices we use to keep our text consistent. There are many idiosyncrasies to \LaTeX , too many to go into here, but a few of note:

- Quotes are written with two backticks (opening) or two apostrophes (closing).
- Please leave a blank line between paragraphs.
- Special characters should always be escaped (see [Appendix A](#) or Internet resources).

Feel free to scroll through chapter files to look for examples of usage if you are in doubt. The answers to further questions can probably be found online (I especially like <http://tex.stackexchange.com>).

How Makefile Documentation Should Look

Notice how documentation will begin on a new page and a new page will be created at the end of the documentations. This is necessary to allow for wider margins that make the code easier to read.

```
# act-plus freesurfer makefile

# This is where the subject directories live
PROJHOME=/projects2/act-plus/subjects/session1

cwd=$(shell pwd)

SUBJECTS=$(shell cat /projects2/act-plus/uds/good_subjects.txt)
#SUBJECT=$(notdir $(cwd))

# Set open MP number of threads to be 1, so that we can parallize using make.
export OMP_NUM_THREADS=1

# for Freesurfer (running version 5.3)
export SUBJECTS_DIR=/projects2/act-plus/freesurfer
export QA_TOOLS=/usr/local/freesurfer/QAtools_v1.1

# be really careful with paths and variables - two versions of freesurfer
# installed
export FREESURFER_SETUP = /usr/local/freesurfer/stable5_3/SetUpFreeSurfer.sh
export RECON_ALL = /usr/local/freesurfer/stable5_3/bin/recon-all $(RECON_FLAGS)
export TKMEDIT = /usr/local/freesurfer/stable5_3/bin/tkmedit

define usage
@echo Usage:
@echo "make, or make interactive          Makes interactive targets"
@echo "make noninteractive                Makes noninteractive targets"
@echo
@echo Noninteractive targets:
@echo "make setup                        Copies source files to this directory"
@echo "make freesurfer                    Runs freesurfer"
@echo
@echo Other useful targets:
@echo "make clean                        Remove everything! Be careful!"
@echo "make mostlyclean                  Remove everything but the good bits."
@echo "make help                          Print this message."
@echo
@echo Variables:
@echo "RECON_FLAGS                        Set to flags to recon-all, by default"
@echo "WAVE                              1, 2 or 3 to select subjects, for setup"
endef

export SHELL=/bin/bash

.PHONY: qa clean mostlyclean output noninteractive

noninteractive: setup freesurfer

all: noninteractive
```

```

output=$(SUBJECTS:%=%/mri/aparc+aseg.mgz) $(SUBJECTS:%=%/mri/brainmask.nii.gz)
freesurfer: $(output)

#recon-all $(RECON_FLAGS) -subjid ${subj} -FLAIR ${subj}/flair.nii.gz -FLAIRpial;\

#####

qfiles=$(SUBJECTS:%=QA/%)

qa: $(qfiles)

QA/%: %
source $$FREESURFER_SETUP ;\
$(QA_TOOLS)/recon_checker -s $*

#####

%/mri/aparc+aseg.mgz: $(PROJHOME)/%/memprage/T1.nii.gz
rm -rf `dirname $@`/IsRunning.*
source /usr/local/freesurfer/stable5_3/SetUpFreeSurfer.sh ;\
export SUBJECTS_DIR=$(SUBJECTS_DIR) ;\
/usr/local/freesurfer/stable5_3/bin/recon-all -i $< -subjid $* -all ;\
/usr/local/freesurfer/stable5_3/bin/recon-all -s $* -T2
    $(PROJHOME)/$*/flair/Flair.nii.gz -T2pial

%/mri/brainmask.nii.gz: $(SUBJECTS_DIR)/%/mri/aparc+aseg.mgz
source /usr/local/freesurfer/stable5_3/SetUpFreeSurfer.sh ;\
mri_convert $(SUBJECTS_DIR)/$*/mri/brainmask.mgz $@

clean:
echo rm -rf $(inputdirs)

mostlyclean:
@echo "Here I would delete things that are not necessary after all is said and done."

output:
@echo "Nothing yet"

setup: $(SUBJECTS)

%: $(PROJHOME)/%/memprage/T1.nii.gz
mkdir -p $@/mri/orig; \
cp $^ $@/mri/orig; \
cd $@/mri/orig; \
mri_convert T1.nii.gz 001.mgz

help:
$(usage)

```

Appendix A

makeToTeX

In case your script gives L^AT_EX problems, the steps makeToTeX undertakes are listed here.

1. ;\ is replaced with ;\textbackslash
 2. Covert any recipes to a special command to turn the target blue.
 3. Strip comments (lines beginning with #).
 - (a) At present, it doesn't strip trailing comments. Please remove those yourself.
 4. Escape the characters \$, %, and #.
 5. Add the terminal newline command (\n) to non-blank lines.
 6. Replace ^ with \textasciicircum.
 7. Replace double dashes (--) with a macro that stops it from turning them into hyphens (-).
 8. Replace single TAB characters with a macro.
 9. Replace multiple TAB characters with \hfill, a L^AT_EX command that spaces nicely.
 10. Leading and trailing quotes are specified differently in L^AT_EX. Leading quotes are coded with a backtick and trailing with an apostrophe.
 11. Extra spaces and multiple blank lines are filtered out.
-

Please ensure all comments are removed! They are very troublesome to display.

Appendix B

Style Guide

Please refer to this guide when in doubt. It covers typographic and phrasing consistencies.

1. Monospace (use the `\texttt{}` macro) the following things:

- (a) **make**, when referring to the program.
 - i. Use the macro `\maken{}`. To remove the trailing space (for example, to use a comma after **make**), call the macro without the braces, `\maken`.
 - ii. Never uppercase.
- (b) **bash**
 - i. Use the macro `\bashn{}`.
- (c) Any **bash** or **make** command. This is done automatically in the one-liner and multi-line environments. Be sure to monospace commands written in-line.
- (d) Explicit references to a file: e.g., “...open `subjects.txt`.”
- (e) Commands run from the command line: e.g., `ls`, `bet`, `flirt`, `qmake`, etc ...
- (f) References to directories: e.g., `/bin/`, `oasis-multisubject-sample/`.
 - i. Use the trailing slash for clarity in all cases.
- (g) References to named variables: e.g., `$(SUBJECTS)`, `$<`, `%`, `$PWD`.
- (h) Command-line flags written in-line: e.g., `-n`, `-lart`.

2. Don't monospace:

- (a) **Don't double up on quotes and monospacing.** If both would be used, prefer quotes. This is to avoid a complete monospacing overload.
- (b) Abstractions or filetypes: e.g., “Makefiles are usually located ...” or “The nifti files ...”
- (c) Programs typically used through a GUI: e.g. OpenOffice, gimp.
- (d) Tool suites, like FSL or AFNI.
- (e) The names of operating systems.
- (f) References to the grid engine in any form.
 - i. Use the article with the abbreviation: “the SGE.”
 - ii. “Grid engine,” not “gridengine.” This is Oracle/SG's usage, although they capitalize it, which we won't do.
- (g) Subject identifiers.

3. Oxford comma = yes: e.g. “X, Y, and Z,” not “X, Y and Z.”

-
4. Don't use em dashes (–) for emphasis. They should appear in pairs the majority of the time.
 5. Margin notes and footnotes:
 - (a) Use margin notes to call out things a new reader would find useful:
 - i. Define important terms with a [?].
 - ii. Call out important notes with a [!].
 - (b) Use footnotes for things that could be completely ignored: advanced features, humorous commentary, etc. . . .
 6. \LaTeX swallows up two dashes (--). Use `\dd` to get a double dash. You can use a space after this command and \LaTeX will ignore it.
 7. **Skull-strip:** Two words, always hyphenated.
 8. Headers:
 - (a) Chapters and Sections Are in Title Case
 - (b) Subsections in sentence case
 - (c) Figure captions in sentence case
 9. Numerals:
 - (a) Spell out numbers less than 10, except the ones \LaTeX generates.
 - (b) Always spell out a number at the beginning of a sentence. However, it is usually better to recast the sentence to avoid this.
 - (c) If you must refer to multiple number things, try:
 - i. Using A, B, C . . . instead: "Process A will overwrite foo.out just in time for process B to use it."
 10. Plurals
 - (a) Since we may have to pluralize some weird things: No apostrophe between the stem and the "s," no matter what. Use the same formatting (e.g. monospacing). If possible, recast the sentence.
 11. M/makefile
 - (a) "a makefile" is an abstract noun referring to any make script (cf. "a shell script"). This should be capitalized when appropriate.
 - (b) "the Makefile" is the top-level makefile in a directory, usually named `Makefile` by IBIC conventions. This reference to the file with the same name does not need to be monospaced.
 12. Backticks and single quotes in monospaced font:
 - (a) Use `\`{ }` to get the correct backtick symbol.
 - (b) There is currently no support for a straight single quote in the default `\ttfamily` font or Inconsolata at present.
 13. Ellipses
 - (a) Spaces before and after: "this . . . and that."
 - (b) Use `\ldots`
 14. FreeSurfer