```
VERSION          = 1.6.3
RELEASE          = 2
DATE             = $(shell date)
NEWRELEASE       = $(shell echo $$(($(RELEASE) + 1)))
PYTHON           = python
PROJECT_NAME     = profile_projects
TOPDIR = $(shell pwd)
DIRS     = profiles_api profiles_project
PYDIRS   = profiles_api profiles_project
EXAMPLEDIR = examples
MANPAGES =
A2PS2S1C  = /bin/a2ps --sides=2 --medium=Letter --columns=1 --portrait --line-numbers=1 --font-size=8
A2PSTMP    = ./tmp
DOCS       = ./docs

SHELL := /bin/bash


all: .pdf

#https://stackoverflow.com/questions/6273608/how-to-pass-argument-to-makefile-from-command-line
args = `arg="$(filter-out $@,$(MAKECMDGOALS))" && echo $${arg:-${1}}`
%:
        @:

versionfile:
        echo "version:" $(VERSION) > etc/version
        echo "release:" $(RELEASE) >> etc/version
        echo "source build date:" $(DATE) >> etc/version

manpage:
        for manpage in $(MANPAGES); do (pod2man --center=$$manpage --release="" ./docs/$$manpage.pod > ./docs/$$manpage.1); done


build: clean
        $(PYTHON) setup.py build -f

clean: cleantmp
        -rm -f  MANIFEST
        -rm -rf dist/ build/
        -rm -rf *~
        -rm -rf rpm-build/
        -rm -rf deb-build/
        -rm -rf docs/*.1
        -rm -f etc/version
        -find . -type f -name *.pyc -exec rm -f {} \;
        -find . -type f -name *~  -exec rm -f {} \;
        -find profiles_*  -type f -name "__init__.py"  -exec rm -rf {} \;
        -find profiles_*  -type d  -name "__pycache__"  -exec rm -rf {} \;

clean_hard:
        -rm -rf $(shell $(PYTHON) -c "from distutils.sysconfig import get_python_lib; print get_python_lib()")/adagios


clean_hardest: clean_rpms


install: build manpage
        $(PYTHON) setup.py install -f

install_hard: clean_hard install

install_harder: clean_harder install

install_hardest: clean_harder clean_rpms rpms install_rpm

install_rpm:
        -rpm -Uvh rpm-build/adagios-$(VERSION)-$(NEWRELEASE)$(shell rpm -E "%{?dist}").noarch.rpm


recombuild: install_harder

clean_rpms:
        -rpm -e adagios

sdist:
        $(PYTHON) setup.py sdist

pychecker:
        -for d in $(PYDIRS); do ($(MAKE) -C $$d pychecker ); done
pyflakes:
        -for d in $(PYDIRS); do ($(MAKE) -C $$d pyflakes ); done

money: clean
        -sloccount --addlang "makefile" $(TOPDIR) $(PYDIRS) $(EXAMPLEDIR)
```

```

testit: clean
	-cd test; sh test-it.sh

unittest:
	-nosetests -v -w test/unittest

rpms: build sdist
	mkdir -p rpm-build
	cp dist/*.gz rpm-build/
	rpmbuild --define "_topdir %(pwd)/rpm-build" \
	--define "_builddir %{_topdir}" \
	--define "_rpmdir %{_topdir}" \
	--define "_srcrpmdir %{_topdir}" \
	--define '_rpmfilename %%{NAME}-%%{VERSION}-%%{RELEASE}.%%{ARCH}.rpm' \
	--define "_specdir %{_topdir}" \
	--define "_sourcedir  %{_topdir}" \
	-ba adagios.spec
debs: build sdist
	mkdir -p deb-build
	cp dist/*gz deb-build/adagios_${VERSION}.orig.tar.gz
	cp -r debian.upstream deb-build/debian
	cd deb-build/ ; \
	  tar -zxvf adagios_${VERSION}.orig.tar.gz ; \
	  cd adagios-${VERSION} ;\
	  cp -r ../debian debian ;\
	  debuild -i -us -uc -b

coffee:
	cd adagios/media/js/ && coffee -c adagios.coffee

trad: coffee
	cd adagios && \
	django-admin.py makemessages --all -e py,html && \
	django-admin.py makemessages --all -d djangojs && \
	django-admin.py compilemessages

#Ref: https://stackoverflow.com/questions/1490949/how-to-write-loop-in-a-makefile
# MANIFEST
SRC1= Makefile
SRC2=
#SRC2= manage.py profiles_projects-dir-layout.txt

cleantmp:
	rm -f ${A2PSTMP}/*.ps ${A2PSTMP}/*.pdf
.ps: cleantmp
	$(foreach var, $(SRC1), ${A2PS2S1C} $(var) --output=${A2PSTMP}/$(var).ps ;)
	$(foreach var, $(SRC2), ${A2PS2S1C} $(var) --output=${A2PSTMP}/$(var).ps ;)
	touch .ps

allpdf: .pdf
	make -C profiles_api pdf
	make -C profiles_project pdf
	touch .pdf
.pdf: .ps
	$(foreach var, $(SRC1), (cd ${A2PSTMP};ps2pdf $(var).ps $(var).pdf);)
	$(foreach var, $(SRC2), (cd ${A2PSTMP};ps2pdf $(var).ps $(var).pdf);)
	rm -f ${A2PSTMP}/*.ps
	cp ${A2PSTMP}/*.pdf  ${DOCS}/
	touch .pdf

tree: clean
	tree -L 4 > ${PROJECT_NAME}-dir-layout.txt

# https://stackoverflow.com/questions/7507810/how-to-source-a-script-in-a-makefile/7508273
activate:
	@echo "Run command"
	@echo "source ./env/bin/activate"
	@echo "  deactivate to exit"

status:
	git status
commit:
	git commit -am "$(call args, Automated lazy commit message without details, read the code change)"  && git push


```