

Mar 07, 21 0:21

Makefile

Page 1/2

```

1  VERSION      = 1.6.3
2  RELEASE      = 2
3  DATE         = $(shell date)
4  NEWRELEASE   = $(shell echo $$(( $(RELEASE) + 1 )) )
5  TOPDIR       = $(shell pwd)
6  DIRS         = profiles_api profiles_project
7  A2PS2S1C     = /bin/a2ps --sides=2 --medium=Letter --columns=1 --portrait --line-numbers=1 --font-size=8
8  A2PSTMP      = /tmp
9  DOCS         = ./docs
10
11  SHELL        := /bin/bash
12  #TRANS       = ~/.local/bin/translate
13  #TRANS2      = /home/x220/github/translate-shell/translate
14  TRANS        = /usr/bin/trans
15  TRANS2       = /usr/bin/trans
16
17  DIA=/usr/bin/dia --nosplash --export=/tmp/t.png inplace-workflow.dia
18
19
20  # enable makefile to accept argument after command
21  #https://stackoverflow.com/questions/6273608/how-to-pass-argument-to-makefile-from-command-line
22  args = `arg="$(filter-out $@,$(MAKECMDGOALS))" && echo ${arg:-${1}}`
23  %:
24      @:
25
26  all: help
27
28  versionfile:
29      echo "version:" $(VERSION) > etc/version
30      echo "release:" $(RELEASE) >> etc/version
31      echo "source build date:" $(DATE) >> etc/version
32
33  clean_hard:
34      -rm -rf $(shell $(PYTHON) -c "from distutils.sysconfig import get_python_lib; print get_python_lib()")/adagios
35
36  clean_hardest: clean_rpms
37
38  #Ref: https://stackoverflow.com/questions/1490949/how-to-write-loop-in-a-makefile
39  # MANIFEST
40  SRC1= Makefile README.md
41  SRC2= README.cat.md
42  #SRC2= manage.py profiles_projects-dir-layout.txt
43
44  cleantmp:
45      rm -f ${A2PSTMP}/*.ps ${A2PSTMP}/*.pdf
46  ps: cleantmp
47      $(foreach var, $(SRC1), ${A2PS2S1C} $(var) --output=${A2PSTMP}/${var}.ps ; )
48      $(foreach var, $(SRC2), ${A2PS2S1C} $(var) --output=${A2PSTMP}/${var}.ps ; )
49
50  pdf: ps
51      $(foreach var, $(SRC1), (cd ${A2PSTMP};ps2pdf $(var).ps $(var).pdf); )
52      $(foreach var, $(SRC2), (cd ${A2PSTMP};ps2pdf $(var).ps $(var).pdf); )
53      rm -f ${A2PSTMP}/*.ps
54      cp ${A2PSTMP}/*.pdf ${DOCS}/
55
56  tree: clean
57      tree -L 4 > ${PROJECT_NAME}-dir-layout.txt
58
59  status:
60      git status
61
62  commit:
63      git commit -am "$(call args, Automated commit message without details, Please read the code difference)" && git push
64
65  sc2tc: sc2tc01
66
67  sc2tc01:
68      ${TRANS} "æM-^M-^MâM-^J;âM-^Y`âM-^Zç«M-^YçM-^B'è@¼ç½@" -d zh-tw -s zh-cn
69
70  tc2sc:
71      @echo "Simplified Chinese to Traditonal Chinese"
72
73  sc2tc:
74      @echo "Traditonal Chinese to Simplified Chinese"
75
76  demo: sc2tc01 file01 guide-1-3-chinese guide-1-3-japanese
77      @echo
78
79  file01:
80      ${TRANS2} file:///stage/greet.txt :zh-tw
81      ${TRANS2} file:///stage/greet.txt :zh-cn
82
83  guide-1-3-chinese:
84      ${TRANS2} file:///docs-style-guide/sample-documentation/advanced-docs.md :zh-tw
85      ${TRANS2} file:///docs-style-guide/sample-documentation/beginner-docs.md :zh-cn
86      ${TRANS2} file:///docs-style-guide/sample-documentation/simplified-docs.md :zh-cn
87
88  guide-1-3-japanese:
89      ${TRANS2} file:///docs-style-guide/sample-documentation/advanced-docs.md :jp
90      # ${TRANS2} file:///docs-style-guide/sample-documentation/beginner-docs.md :jp

```

Mar 07, 21 0:21

Makefile

Page 2/2

```

87 #      ${TRANS2} file:///./docs-style-guide/sample-documentation/simplified-docs.md :jp
88 trans-ver:
89      ${TRANS2} -V
90 testcmds:
91      @which openc
92      @which translate
93
94 with-dia:
95      @which dia
96
97 help:
98      @echo "Usage: make <target>"
99      @echo
100      @echo "Available targets are:"
101      @echo "  help      show this text"
102      @echo "  clean     clean the mess"
103      @echo "  sc2tc01   demonstrate S.C. to T.C."
104      @echo "  print_release  print release how it should look like with"
105      @echo "               with the given parameters"
106      @echo "  source    create the source tarball suitable for"
107      @echo "            packaging"
108      @echo "  srpm      create the SRPM"
109      @echo "  copr_build  create the COPR build using the COPR TOKEN"
110      @echo "            - default path is: $(_COPR_CONFIG)"
111      @echo "            - can be changed by the COPR_CONFIG env"
112      @echo "  install-deps  create python virtualenv and install there"
113      @echo "               leapp-repository with dependencies"
114      @echo "  install-deps-fedora  create python virtualenv and install there"
115      @echo "               leapp-repository with dependencies for Fedora OS"
116      @echo "  lint      lint source code"
117      @echo "  test      lint source code and run tests"
118      @echo "  test_no_lint  run tests without linting the source code"
119      @echo ""
120      @echo "Targets test, lint and test_no_lint support environment variables ACTOR and"
121      @echo "TEST_LIBS."
122      @echo "If ACTOR=<actor> is specified, targets are run against the specified actor."
123      @echo "If TEST_LIBS=y is specified, targets are run against shared libraries."
124      @echo ""
125      @echo ""
126      @echo "Possible use:"
127      @echo "  make <target>"
128      @echo "  PR=5 make <target>"
129      @echo "  MR=6 make <target>"
130      @echo "  PR=7 SUFFIX='my_additional_suffix' make <target>"
131      @echo "  MR=6 COPR_CONFIG='path/to/the/config/copr/file' make <target>"
132      @echo "  ACTOR=<actor> TEST_LIBS=y make test"
133      @echo ""
134
135 clean:
136      @echo "---- Clean repo ----"
137      @rm -rf packaging/{sources,SRPMS,tmp}/
138      @rm -rf build/ dist/ *.egg-info .pytest_cache/
139      @find . -name 'leapp.db' | grep "\leapp/leapp.db" | xargs rm -f
140      @find . -name '__pycache__' -exec rm -fr {} +
141      @find . -name '*.pyc' -exec rm -f {} +
142      @find . -name '*.pyo' -exec rm -f {} +
143      -rm -rf dist/ build/
144      -rm -rf *~
145      -find . -type f -name *.pyc -exec rm -f {} \;
146      -find . -type f -name *~ -exec rm -f {} \;
147

```