Subject: Makefile's install depends on all
Posted by bonami on Thu, 19 Mar 2015 05:03:07 GMT
View Forum Message <> Reply to Message

I use command line umk to build.
I wonder from where I got this Makefile.
first I run make, which will build "all".
Then I run make install. It relies on "all". "all" will complain ${program_name} already exists, which IS the former result.
"install" should depends on "all" but use its result if it has been run.

And, I wonder why "install" needs gawk, which is not needed when I compile anything in theide.

File Attachments
1) Makefile, downloaded 459 times

Subject: Re: Makefile's install depends on all
Posted by dolik.rce on Thu, 19 Mar 2015 06:05:29 GMT
View Forum Message <> Reply to Message

Hi bonami,

bonami wrote on Thu, 19 March 2015 06:03I use command line umk to build.
first I run make, which will build "all".
Then I run make install. It relies on "all". "all" will complain ${program_name} already exists, which IS the former result.
"install" should depends on "all" but use its result if it has been run.

Did you write the Makefile? It seems quite overcomplicated, I couldn't really figure out what should all the shell command in the targets actually do. But what I see at first sight is that there are no dependencies at all. So make have no idea that your application was already built, when running install and tries to build it again. You have to list the files that are created by each target as its dependencies. For example:all: ${program_name}
This should fix your problem. It would also probably allow you to simplify the makefile a lot. GNU Make is a great tool with lots of possibilities, if you haven't read any of it's documentation you probably should

bonami wrote on Thu, 19 March 2015 06:03And, I wonder why "install" needs gawk, which is not needed when I compile anything in theide.Because you're using it on line 73: $(AWK) 'BEGIN { files["."] = ""; dirs["."] = 1 } \
  { d=$$3; if (dirs[d] != 1) { print "d", d; dirs[d] = 1 } \
    if ($$2 == $$4) files[d] = files[d] " " $$1; \
    else { print "f", $$3 "/" $$4, $$1; } } \
  END { for (d in files) print "f", d, files[d] }'

Best regards,
Honza

Subject: Re: Makefile's install depends on all
Posted by bonami on Thu, 19 Mar 2015 06:56:17 GMT
View Forum Message <> Reply to Message

It is a strange makefile.
Isn't there a universal Makefile for umk?
I added your suggestion as dependency and changed it as the following. It is better.
all: ${program_name}

${program_name}: ${code_files}
	@echo building...; \

Subject: Re: Makefile's install depends on all
Posted by dolik.rce on Thu, 19 Mar 2015 07:48:52 GMT
View Forum Message <> Reply to Message

bonami wrote on Thu, 19 March 2015 07:56It is a strange makefile.
Then why are you using it? You can write better one

bonami wrote on Thu, 19 March 2015 07:56Isn'
I added your suggestion as dependency and changed it as the following. It is better.
all: ${program_name}

${program_name}: ${code_files}
	@echo building...; \
t there a universal Makefile for umk?
There is a universal makefile for U++ that doesn't need umk at all Search the forum. I am using it to build U++ packages.

Honza

Subject: Re: Makefile's install depends on all
Posted by bonami on Mon, 23 Mar 2015 01:43:15 GMT
View Forum Message <> Reply to Message

I think you meant https:// code.google.com/p/upp-mirror/source/browse/trunk/uppbox/lpbu ild2/mkfile
I did not mention it because I cannot understand it. Don't know how to migrate it, since its parameters include package, etc. Plus, it has no "dist" target?

Subject: Re: Makefile's install depends on all
Posted by dolik.rce on Mon, 23 Mar 2015 05:21:47 GMT
View Forum Message <> Reply to Message

bonami wrote on Mon, 23 March 2015 02:43I think you meant https://
code.google.com/p/upp-mirror/source/browse/trunk/uppbox/lpbu ild2/mkfile
I did not mention it because I cannot understand it. Don't know how to migrate it, since its parameters include package, etc. Plus, it has no "dist" target?Yes, that's it. I'm not sure what you mean by dist target, but it definitely doesn't have it. Also it has no install target, it only handles building. You can have a look on how it can be used at the other makefile in the same directory, which includes the general one to build theide and umk:
https://code.google.com/p/upp-mirror/source/browse/trunk/upp box/lpbuild2/Makefile

Simpler example might be something like this (I haven't got time to actually test it, so sorry if there are some minor issues):PKG=MyPackage
NESTS=mysrc uppsrc
FLAGS=GUI GCC DEBUG DEBUG_FULL STACKTRACE

build: MyPackage

this is only needed if you want support parallel builds with dpkg-buildpackage -j
ifneq (,$(filter parallel=%,$(DEB_BUILD_OPTIONS)))
  JOBS:=$(patsubst parallel=%,%,$(filter parallel=%,$(DEB_BUILD_OPTIONS)))
else
  JOBS:=1
endif

OPTS:=CC="cc -g" CXX="c++ -g" CFLAGS="-g" CXXFLAGS="-g" LDFLAGS="-Wl,--gc-sections -Wl,-O,2" TIME= COLOR=0 USEMAINCFG=0
y
$(PKG):
	$(MAKE) -f mkfile JOBS=$(JOBS) PKG=$@ "NESTS=$(NESTS)" $(OPTS) "FLAGS=$(FLAGS)" TARGET=`pwd`/$@

install:
	install -D $(PKG) $(DESTDIR)/usr/bin/

clean:
	rm -f $(PKG)

Just edit the first three lines to suit your needs and place it to the same directory where the mkfile and you nests (mysrc and uppsrc in this exaple) are.

Honza