How does "make" app know default target to build if no target is specified?

Asked 12 years ago Active 5 months ago Viewed 160k times

234

Most Linux apps are compiled with:

```
make
make install clean
```

40

As I understand it, the `make` command takes names of build targets as arguments. So for example `install` is usually a target that copies some files to standard locations, and `clean` is a target that removes temporary files.

But what target will `make` build if no arguments are specified (e.g. the first command in my example)?

makefile

Share Improve this question Follow

edited Aug 11 '21 at 14:13

waldyrious
3,253 ● 4 ● 31 ● 37

asked Jan 13 '10 at 15:15

grigoryvp
36.2k ● 58 ● 161 ● 267

4 Answers

Active | Oldest | Votes

279

By default, it begins by processing the first target that does not begin with a `.` aka the default goal; to do that, it may have to process other targets - specifically, ones the first target depends on.

The GNU Make Manual covers all this stuff, and is a surprisingly easy and informative read.

Share Improve this answer Follow

edited Aug 29 '19 at 1:39

Marcel Gosselin
4,510 ● 2 ● 27 ● 51

answered Jan 13 '10 at 15:19
anon

3 If the GNU make manual covers this stuff, why was this question asked? The manual is great at documenting how to write rules, but makes no mention on how to write a Makefile. – Nick Dec 31 '14 at 16:35

1 I'm looking at a make file that `include`s another makefile, whose first target is `%.o : %.cpp`. So the default target is... to build all .cpp files? – Qwertie Nov 9 '18 at 19:55

237

To save others a few seconds, and to save them from having to read the manual, here's the short answer. Add this to the top of your make file:

```
.DEFAULT_GOAL := mytarget
```

mytarget will now be the target that is run if "make" is executed and no target is specified.

If you have an older version of make (<= 3.80), this won't work. If this is the case, then you can do what anon mentions, simply add this to the top of your make file:

```
.PHONY: default
default: mytarget ;
```

References: https://www.gnu.org/software/make/manual/html_node/How-Make-Works.html

Share Improve this answer Follow

edited Aug 1 '17 at 18:02

Alison R.
4,084 ● 26 ● 32

answered May 11 '15 at 19:52

Samuel
6,626 ● 8 ● 39 ● 39

1 @nathan It makes mytarget the default target if someone runs "make" without any parameters. – Samuel Jan 15 '16 at 14:55

1 Note that .DEFAULT_GOAL doesn't appear to be supported in GNU make v3.80, and I would assume prior versions as well (v3.81 does support it though). If you are running an older version of make you will have to make sure your default target is the first/topmost target in your make file. – Samuel Mar 25 '16 at 19:15

1 is `default:` a special target name, or this works because it's the first target in the file? – Anentropic Sep 23 '16 at 9:19

9 "...to save them from having to read the manual" is one of the points of SO: quick answers to questions that would otherwise take awhile to dig for in ancient documentation. – WattsInABox Dec 14 '17 at 16:22

3 This should be the correct answer. Explicit over implicit wins in this case. – Sion Sep 14 '18 at 9:05

Share Improve this answer Follow

edited Aug 29 '19 at 4:58

jottr

3,111 ● 3 ● 23 ● 33

answered May 6 '11 at 6:04

Sandip Bhattacharya

964 ● 9 ● 11

1

bmake's equivalent of GNU Make's `.DEFAULT_GOAL` is `.MAIN`:

```
$ cat Makefile
.MAIN: foo

all:
    @echo all

foo:
    @echo foo
$ bmake
foo
```

See the bmake(1) manual page.

Share Improve this answer Follow

answered Jul 5 '21 at 19:36

Mateusz Piotrowski

6,570 ● 9 ● 46 ● 74
