



Suppress echoing of commands in a makefile

Content

- 1 Objective
- 2 Background
- 3 Scenario
- 4 Method
- 5 Further reading

Objective

To prevent commands in a makefile from being echoed before they are executed

Tested on

Debian (Etch, Lenny, Squeeze)
Fedora (14)
Ubuntu (Hardy, Intrepid, Jaunty, Karmic, Lucid, Maverick, Natty, Trusty)

Background

The default behaviour of `make` is to echo each command to `stdout` before it is executed. This is usually helpful because it places error messages in context and gives an indication of progress. However there are some situations where echoing is less desirable, notably when using the `echo` command to print messages. In that case, allowing the command to be both echoed and executed causes the message to be printed twice.

Scenario

Suppose that a makefile contains the following rule:

```
foo: foo.c
    echo Building foo module
    gcc -o $@ $<
```

The default behaviour is to echo both commands. You wish to suppress this for the `echo` command but not for the `mkfoo` command.

Method

Prepend an at sign (`'@'`) to any command that you do not want to be echoed:

```
foo: foo.c
    @echo Building foo module
    gcc -o $@ $<
```

This should have the desired effect with any POSIX-compatible implementation of `make`.

Further reading

The Open Group Base Specifications Issue 6, IEEE Std 1003.1, The Open Group, 2004

Tags: [make](#)

© 2010–2014 [Graham Shaw](#), [some rights](#) reserved.