Usage of dash (-) in place of a filename

Asked 11 years, 2 months ago Modified 2 years, 5 months ago Viewed 346k times

For a command, if using `-` as an argument in place of a file name will mean STDIN or STDOUT.

1. But in this example, it creates a file with the name `-`:

```
echo hello > -
```

   How can I make `-` in this example mean STDOUT?

2. Conversely, how can I make `-` mean a file named `-` in examples such as:

```
cat -
```

shell command-line utilities

edited Jul 22, 2011 at 8:42

Caleb
66.7k 16 193 220

asked Jul 9, 2011 at 22:14

Tim
92.3k 163 510 893

4 Since I apparently have to have 50 reputation to comment directly... On the /dev/stdin /dev/stdout comment, AIX, which is a legitimate UNIX derivative does not have these pseudodevices. And, as a further comment, LINUX is not a UNIX derivative in any case. It is a POSIX-compliant workalike, and the most popular of the UNIX-ish OS's at this point, but make no mistake, this is no UNIX. But, the gist of the replies here are correct. The "-" notation is not interpreted as special by the shell, and is thus passed directly to each individual application as an ARG. If the application does not recogni – user95873 Dec 23, 2014 at 20:04

There are also `2>&-` construction, which means "close descriptor 2". – user3132194 Dec 2, 2016 at 5:58

@user95873, what I suppose you wanted to say is: *while Linux is Unix-like, not every Unix-like (or true UNIX) is Linux*. The matter whether Linux is or isn't true UNIX (i.e. complies the Single UNIX Specification) doesn't have relation to /dev/std{in,out,err} issue. As /dev/std{in,out,err} is *added* feature, not *missing*. – sasha Dec 9, 2016 at 12:32

6 Answers

Sorted by: Highest score (default)

208

Using `-` as a filename to mean stdin/stdout is a convention that a lot of programs use. It is not a special property of the filename. The kernel does not recognise `-` as special so any system calls referring to `-` as a filename will use `-` literally as the filename.

With bash redirection, `-` is not recognised as a special filename, so bash will use that as the literal filename.

When `cat` sees the string `-` as a filename, it treats it as a synonym for stdin. To get around this, you need to alter the string that `cat` sees in such a way that it still refers to a file called `-`. The usual way of doing this is to prefix the filename with a path - `./-`, or `/home/Tim/-`. This technique is also used to get around similar issues where command line options clash with filenames, so a file referred to as `./-e` does not appear as the `-e` command line option to a program, for example.

Share Improve this answer Follow

edited Jul 10, 2011 at 5:41

answered Jul 9, 2011 at 23:25

camh

36.4k 8 72 62

50 It's worth adding that `/dev/stdin` and `/dev/stdout` are universally available and can be used in place of the `-` convention. – jmtd Jul 22, 2011 at 9:00

12 @jmtd: /dev/std{in,out} are not universally available. Not all unixes have it. – camh Jul 22, 2011 at 10:37

2 Interesting, I assumed they were part of POSIX (but can't confirm). They're present at least on Linux, the BSDs and Solaris. Can you give an example of a modern UNIX that lacks them? – jmtd Aug 2, 2011 at 8:51

1 @camh Not sure if you got the notification, and this is a very, very old topic; I'm not sure if you can, but I'm very curious if you know the answer to `jmtd`'s question? :) – Swivel Nov 23, 2016 at 6:29

2 @jmtd I seem to have found a tentative answer to the question: unix.stackexchange.com/a/278368/31669 – Swivel Nov 23, 2016 at 6:35

1. Instead of `echo hello > -`, you can use `echo hello > /dev/stdout`.

21

While '-' is a convention that has to be implemented by each program wanting to support it, `/dev/stdin`, `/dev/stdout` and `/dev/stderr` are, when supported by the OS (at least Solaris, Linux and BSDs do), independent of the application and then will work as you intend.

Share Improve this answer Follow

edited Apr 29, 2013 at 2:41

answered Jul 10, 2011 at 2:26

jlliagre

58.5k 10 111 154

13

As camh mentioned, `-` is just a naming convention used by some programs. If you want to refer to these streams with a file descriptor the shell will recognize, jlliagre was correct in having you use the name `/dev/stdin` or `/dev/stdout` instead. Those file names should work any place a normal file name would work.

1. That being said, your first example is kind of silly. Any output that would be caught by the redirect operator to write to a file is already ON standard-output, so redirecting it and writing it back to where it came from is useless. The behavior you use there is the pipe, not a redirect:

```
echo hello |
```

2. In your second example you simply need to give can some indication that you want a litteral file of that name, not the internal alias it has. You can do this easiest by specifying a path to the file like this:

```
cat ./-
```

Share Improve this answer Follow

edited Apr 13, 2017 at 12:36

Community Bot

1

answered Jul 10, 2011 at 6:18

Caleb

66.7k 16 193 220

6

As for 1, the program has to support it. You can't just arbitrarily use it. As for 2, redirect input from (e.g., `cat < -`).

Share Improve this answer Follow

answered Jul 9, 2011 at 22:22

bahamat

36.6k 3 68 103

The '-' approach has a lot of problems. First of all it requires an interpretation of the '-' character and many programs don't perform such interpretation. And furthermore, there are some programs that interpret an hyphen as a delimiter marking the end of command line options. The programs are written to work with filename arguments, the '-' approach is an hack, nice but weak.

The best way is:

```
$ echo hello > /dev/fd/1
```

`/dev/stdout` is a symbolik link of `/dev/fd/1`

Share Improve this answer Follow

answered Oct 3, 2016 at 18:26

b3h3m0th

322 2 7

The redirection is interpreted by the shell, not the program being invoked. – sherrellbc Jan 3, 2017 at 13:46

Special characters have mostly two meanings:

> ASCII numeric chart.
>
> Scripting or symbolic.

It's possible that a single character represents a string, or act as a string. as my understanding.

in C language `fopen()` function takes two arguments first file stream and the second mode in which file will be open. the mode is a string. even if it's single character.

cat > "-" works.

Share Improve this answer Follow

edited Jun 11, 2020 at 14:16

Community Bot

1

answered Apr 4, 2020 at 9:31

Owais Qureshi

1 3