

## Table Of Contents

Table Of Contents .....	1
xargs is more safer and easy to use .....	2
xargs examples .....	2
{ } as the argument list marker .....	2
Dealing file names with blank spaces and newline .....	2
Avoiding errors and resource hungry problems with xargs and find combo .....	3

[Home](#) > [Faq](#) > [BASH Shell](#)

## xargs: How To Control and Use Command Line Arguments

Posted by [Vivek Gite](#) <[vivek@nixcraft.com](mailto:vivek@nixcraft.com)>

I am trying to use xargs command using shell pipes and not able to understand how to control and use command line arguments. For example I'd like to find out all \*.c file located in 100s of sub-directories and move them to another directory called ~/old.src. How do I use command line args with xargs to achieve the same?



[1]

xargs command is designed to construct argument lists and invoke other utility. xargs reads items from the standard input or pipes, delimited by blanks or newlines, and executes the command one or more times with any initial-arguments followed by items read from standard input. Blank lines on the standard input are ignored.

### xargs is more safer and easy to use

xargs functionality can be achieved using the backquote feature of shell. But, it offers more options. It can deal with blanks or special characters in file names easily. It is often used with find, grep and other commands.

### xargs examples

For example following example will print 1 2 3 4 using xargs (echo command is default)

```
$ echo 1 2 3 4 | xargs echo
```

OR

```
$ echo 1 2 3 4 | xargs
```

You can force xargs to use at most max-args arguments per command line. For example following will use first two argument per command:

```
$ echo 1 2 3 4 | xargs -n 2
```

Find all .bak files in or below the current directory and delete them.

```
$ find . -name "*.bak" -type f -print | xargs /bin/rm -f
```

### { } as the argument list marker

{ } is the default argument list marker. You need to use { } this with various command which take more than two arguments at a time. For example mv command need to know the file name. The following will find all .bak files in or below the current directory and move them to ~/old.files directory:

```
$ find . -name "*.bak" -print0 | xargs -0 -I mv { } ~/old.files
```

You can rename { } to something else. In the following example { } is renamed as file. This is more readable as compare to previous example:

```
$ find . -name "*.bak" -print0 | xargs -0 -I file mv file ~/old.files
```

Where,

1. **-0** If there are blank spaces or characters (including newlines) many commands will not work. This option takes care of file names with blank space.
2. **-I** Replace occurrences of replace-str in the initial-arguments with names read from standard input. Also, unquoted blanks do not terminate input items; instead the separator is the newline character.

### Dealing file names with blank spaces and newline

The following will work incorrectly if there are any filenames containing newlines or spaces (it will find out all .mp3 file located in current directory and play them using mplayer):

```
$ find . -iname "*.mp3" -print | xargs mplayer
```

To get rid of this problem use -0 option:

```
$ find . -iname "*.mp3" -print0 | xargs -0 -I mp3file mplayer mp3file
```

To find out all \*.c file located in 100s of subdirectories and move them to another directory called ~/old.src, use:

```
$ find /path/to/dir -iname "*.c" -print0 | xargs -0 -I file mv file ~/old.src
```

## Avoiding errors and resource hungry problems with xargs and find combo

To copy all media files to another location called /bakup/iscsi, you can use cp as follows:

```
$ cp -r -v -p /share/media/mp3/ /backup/iscsi/mp3
```

However, cp command may fail if an error occurs such as if the number of files is too large for the cp command to handle. xargs in combination with find can handle such operation nicely. xargs is more resource efficient and will not halt with an error:

```
$ find /share/media/mp3/ -type f -name "*.mp3" -print0 | xargs -0 -r -I file cp -v -p file
```

Please note that all of the above commands are tested with GNU/xargs version. BSD and UNIX xargs command may not have options such as -r. Please refer to your local xargs man page for further info:

```
man xargs
```

4000+ howtos and counting! Want to read more Linux / UNIX howtos, tips and tricks? Subscribe to our [daily email](#) newsletter or [weekly newsletter](#) to make sure you don't miss a single tip/tricks. Alternatively, subscribe via [RSS/XML](#) feed.

Article printed from Frequently Asked Questions About Linux / UNIX: <http://www.cyberciti.biz/faq/>

URL to article: <http://www.cyberciti.biz/faq/linux-unix-bsd-xargs-construct-argument-lists-utility/>

URLs in this post:

[1] Image: <http://www.cyberciti.biz/faq/category/bash-shell/>

---

Copyright © 2006-2010 [nixCraft](#). All rights reserved. This print / pdf version is for personal non-commercial use only. More details <http://www.cyberciti.biz/tips/copyright>.