**NAME**

xargs – construct argument lists and invoke utility

**SYNOPSIS**

**xargs [–t] [–p] [–e[***eofstr***]] [–E** *eofstr***] [–I** *replstr***] [–i[***replstr***]] [–L** *number***] [–l[***number***]] [–n** *number* **[–x]] [–s** *size***] [***utility* **[***argument*. . .**]]**

**AVAILABILITY**

SUNWcsu

**DESCRIPTION**

The **xargs** utility constructs a command line consisting of the *utility* and *argument* operands specified followed by as many arguments read in sequence from standard input as will fit in length and number constraints specified by the options. The **xargs** utility then invokes the constructed command line and waits for its completion. This sequence is repeated until an end-of-file condition is detected on standard input or an invocation of a constructed command line returns an exit status of **255**.

Arguments in the standard input must be separated by unquoted blank characters, or unescaped blank characters or newline characters. A string of zero or more non-double-quote (**"**) and non-newline characters can be quoted by enclosing them in double-quotes. A string of zero or more non-apostrophe (′) and non-newline characters can be quoted by enclosing them in apostrophes. Any unquoted character can be escaped by preceding it with a backslash (\). The *utility* will be executed one or more times until the end-of-file is reached. The results are unspecified if the utility named by *utility* attempts to read from its standard input.

The generated command line length will be the sum of the size in bytes of the utility name and each argument treated as strings, including a null byte terminator for each of these strings. The **xargs** utility will limit the command line length such that when the command line is invoked, the combined argument and environment lists will not exceed **{ARG_MAX}–2048** bytes. Within this constraint, if neither the **–n** nor the **–s** option is specified, the default command line length will be at least **{LINE_MAX}**.

**OPTIONS**

The following options are supported:

**–e[***eofstr***]** Use *eofstr* as the logical end-of-file string. Underscore (_) is assumed for the logical EOF string if neither **–e** nor **–E** is used. When the **–eofstr** option-argument is omitted, the logical EOF string capability is disabled and underscores are taken literally. The **xargs** utility reads standard input until either end-of-file or the logical EOF string is encountered.

**–E** *eofstr* Specify a logical end-of-file string to replace the default underscore. The **xargs** utility reads standard input until either end-of-file or the logical EOF string is encountered.

**–I** *replstr* Insert mode. *utility* will be executed for each line from standard input, taking the entire line as a single argument, inserting it in *argument s* for each occurrence of *replstr*. A maximum of five arguments in *argument*s can each contain one or more instances of *replstr*. Any blank characters at the beginning of each line are ignored. Constructed arguments cannot grow larger than 255 bytes. Option **–x** is forced on. The **–I** and **–i**

options are mutually exclusive; the last one specified takes effect.

**–i[***replstr***]** This option is equivalent to **–I** *replstr*. The string **{ }** is assumed for *replstr* if the option-argument is omitted.

**–L** *number* The *utility* will be executed for each non-empty *number* lines of arguments from standard input. The last invocation of *utility* will be with fewer lines of arguments if fewer than *number* remain. A line is considered to end with the first newline character unless the last character of the line is a blank character; a trailing blank character signals continuation to the next non-empty line, inclusive. The **–L**, **–l**, and **–n** options are mutually exclusive; the last one specified takes effect.

**-l[***number***]** (The letter ell.) This option is equivalent to **–L** *number*. If *number* is omitted, **1** is assumed. Option **–x** is forced on.

**–n** *number* Invoke *utility* using as many standard input arguments as possible, up to *number* (a positive decimal integer) arguments maximum. Fewer arguments will be used if:

- The command line length accumulated exceeds the size specified by the **–s** option (or **{LINE_MAX}** if there is no **–s** option), or
- The last iteration has fewer than *number*, but not zero, operands remaining.

**–p** Prompt mode. The user is asked whether to execute *utility* at each invocation. Trace mode (**–t**) is turned on to write the command instance to be executed, followed by a prompt to standard error. An affirmative response (specific to the user's locale) read from **/dev/tty** will execute the command; otherwise, that particular invocation of *utility* is skipped.

**–s** *size* Invoke *utility* using as many standard input arguments as possible yielding a command line length less than *size* (a positive decimal integer) bytes. Fewer arguments will be used if:

- The total number of arguments exceeds that specified by the **–n** option, or
- The total number of lines exceeds that specified by the **–L** option, or
- End of file is encountered on standard input before *size* bytes are accumulated.

Values of *size* up to at least **{LINE_MAX}** bytes are supported, provided that the constraints specified in **DESCRIPTION** are met. It is not considered an error if a value larger than that supported by the implementation or exceeding the constraints specified in **DESCRIPTION** is given; **xargs** will use the largest value it supports within the constraints.

**–t** Enable trace mode. Each generated command line will be written to standard error just prior to invocation.

**–x** Terminate if a command line containing *number* arguments (see the **–n**

option above) or *number* lines (see the −**L** option above) will not fit in the implied or specified size (see the −**s** option above).

## OPERANDS

The following operands are supported:

*utility* The name of the utility to be invoked, found by search path using the **PATH** environment variable; see **environ**(5). If *utility* is omitted, the default is the **echo**(1) utility. If the *utility* operand names any of the special built-in utilities in **shell_builtins**(1), the results are undefined.

*argument* An initial option or operand for the invocation of *utility*.

## USAGE

The **255** exit status allows a utility being used by **xargs** to tell **xargs** to terminate if it knows no further invocations using the current data stream will succeed. Thus, *utility* should explicitly **exit** with an appropriate value to avoid accidentally returning with **255.**

Note that input is parsed as lines; blank characters separate arguments. If **xargs** is used to bundle output of commands like **find** *dir* −**print** or **ls** into commands to be executed, unexpected results are likely if any filenames contain any blank characters or newline characters. This can be fixed by using **find** to call a script that converts each file found into a quoted string that is then piped to **xargs**. Note that the quoting rules used by **xargs** are not the same as in the shell. They were not made consistent here because existing applications depend on the current rules and the shell syntax is not fully compatible with it. An easy rule that can be used to transform any string into a quoted form that **xargs** will interpret correctly is to precede each character in the string with a backslash (\).

On implementations with a large value for **{ARG_MAX}**, **xargs** may produce command lines longer than **{LINE_MAX}**. For invocation of utilities, this is not a problem. If **xargs** is being used to create a text file, users should explicitly set the maximum command line length with the −**s** option.

The **xargs** utility returns exit status **127** if an error occurs so that applications can distinguish “failure to find a utility” from “invoked utility exited with an error indication.” The value **127** was chosen because it is not commonly used for other meanings; most utilities use small values for “normal error conditions” and the values above **128** can be confused with termination due to receipt of a signal. The value **126** was chosen in a similar manner to indicate that the utility could be found, but not invoked.

## EXAMPLES

1. The following will move all files from directory **$1** to directory **$2**, and echo each move command just before doing it:
   **ls $1 | xargs -I {} -t mv $1/{} $2/{}**
2. The following command will combine the output of the parenthesised commands onto one line, which is then written to the end of file **log**:
   **(logname; date; printf "%s\n" "$0 $*") | xargs >>log**
3. The following command will invoke **diff** with successive pairs of arguments originally typed as command line arguments (assuming there are no embedded blank characters in the elements of the original argument list):
   **printf "%s\n" "$*" | xargs -n 2 -x diff**

4. The user is asked which files in the current directory are to be archived. The files are archived into **arch ;** a, one at a time, or b, many at a time.
   a. **ls | xargs -p -L 1 ar -r arch**

   b. **ls | xargs -p -L 1 | xargs ar -r arch**
5. The following will execute with successive pairs of arguments originally typed as command line arguments:
   **echo $* | xargs -n 2 diff**

## ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **xargs**: **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, **NLSPATH**, and **PATH**.

## EXIT STATUS

The following exit values are returned:

**0** All invocations of *utility* returned exit status **0**.

**1–125** A command line meeting the specified requirements could not be assembled, one or more of the invocations of *utility* returned a non-zero exit status, or some other error occurred.

**126** The utility specified by *utility* was found but could not be invoked.

**127** The utility specified by *utility* could not be found.

If a command line meeting the specified requirements cannot be assembled, the utility cannot be invoked, an invocation of the utility is terminated by a signal, or an invocation of the utility exits with exit status **255**, the **xargs** utility will write a diagnostic message and exit without processing any remaining input.

## SEE ALSO

**echo**(1), **shell_builtins**(1), **environ**(5)