Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'echo.1p' command

$ man echo.1p

ECHO(1P) POSIX Programmer's Manual ECHO(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

echo ? write arguments to standard output

SYNOPSIS

echo [string...]

DESCRIPTION

The echo utility writes its arguments to standard output, followed by a <newline>. If there are no arguments, only the <newline> is written.

OPTIONS

The echo utility shall not recognize the "--" argument in the manner specified by Guideline 10 of the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines; "--" shall be recognized as a string operand.

Implementations shall not support any options.

OPERANDS

The following operands shall be supported:

string A string to be written to standard output. If the first oper?

and is -n, or if any of the operands contain a <backslash>

character, the results are implementation-defined.

On XSI-conformant systems, if the first operand is -n, it shall be treated as a string, not an option. The following character sequences shall be recognized on XSI-conformant systems within any of the arguments:

\a Write an <alert>.

\b Write a <backspace>.

\c Suppress the <newline> that otherwise follows the final argument in the output. All characters following the '\c' in the arguments shall be ignored.

\f Write a <form-feed>.

\n Write a <newline>.

\r Write a <carriage-return>.

\t Write a <tab>.

\v Write a <vertical-tab>.

\\ Write a <backslash> character.

\0num Write an 8-bit value that is the zero, one, two, or three-digit octal number num.

STDIN

Not used.

INPUT FILES

None.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of echo:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of POSIX.1?2017, Section 8.2, Internationalization Variables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as

opposed to multi-byte characters in arguments).

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard er? ror.

NLSPATH Determine the location of message catalogs for the processing of LC_MESSAGES.

ASYNCHRONOUS EVENTS

Default.

STDOUT

The echo utility arguments shall be separated by single <space> charac? ters and a <newline> character shall follow the last argument. Output transformations shall occur based on the escape sequences in the input. See the OPERANDS section.

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

It is not possible to use echo portably across all POSIX systems unless both -n (as the first argument) and escape sequences are omitted.

The printf utility can be used portably to emulate any of the tradi? tional behaviors of the echo utility as follows (assuming that IFS has its standard value or is unset):

* The historic System V echo and the requirements on XSI implementa?
tions in this volume of POSIX.1?2017 are equivalent to:

```
printf "%b\n$*"
```

* The BSD echo is equivalent to:

```
if [ "X$1" = "X-n" ]
then
    shift
    printf "%s$*"
else
    printf "%s\n$*"
fi
```

New applications are encouraged to use printf instead of echo.

EXAMPLES

None.

RATIONALE

The echo utility has not been made obsolescent because of its extremely widespread use in historical applications. Conforming applications that wish to do prompting without <newline> characters or that could possi? bly be expecting to echo a -n, should use the printf utility derived from the Ninth Edition system.

As specified, echo writes its arguments in the simplest of ways. The two different historical versions of echo vary in fatally incompatible ways.

The BSD echo checks the first argument for the string -n which causes it to suppress the <newline> that would otherwise follow the final ar? gument in the output.

The System V echo does not support any options, but allows escape se? quences within its operands, as described for XSI implementations in the OPERANDS section.

The echo utility does not support Utility Syntax Guideline 10 because historical applications depend on echo to echo all of its arguments, except for the -n option in the BSD version.

FUTURE DIRECTIONS

None.

SEE ALSO

printf

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines