



*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 'eval.1p' command***

### ***\$ man eval.1p***

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

eval ? construct command by concatenating arguments

#### SYNOPSIS

eval [argument...]

#### DESCRIPTION

The eval utility shall construct a command by concatenating arguments together, separating each with a <space> character. The constructed command shall be read and executed by the shell.

#### OPTIONS

None.

#### OPERANDS

See the DESCRIPTION.

#### STDIN

Not used.

#### INPUT FILES

None.

#### ENVIRONMENT VARIABLES

None.

## ASYNCHRONOUS EVENTS

Default.

## STDOUT

Not used.

## STDERR

The standard error shall be used only for diagnostic messages.

## OUTPUT FILES

None.

## EXTENDED DESCRIPTION

None.

## EXIT STATUS

If there are no arguments, or only null arguments, `eval` shall return a zero exit status; otherwise, it shall return the exit status of the command defined by the string of concatenated arguments separated by `<space>` characters, or a non-zero exit status if the concatenation could not be parsed as a command and the shell is interactive (and therefore did not abort).

## CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

## APPLICATION USAGE

Since `eval` is not required to recognize the `"--"` end of options delimiter, in cases where the argument(s) to `eval` might begin with `'-'` it is recommended that the first argument is prefixed by a string that will not alter the commands to be executed, such as a `<space>` character:

```
eval " $commands"
```

or:

```
eval "$(some_command)"
```

## EXAMPLES

```
foo=10 x=foo
```

```
y='$$x
```

```
echo $y
```

```
$foo
eval y='$$x
echo $y
10
```

## RATIONALE

This standard allows, but does not require, `eval` to recognize `--`. Although this means applications cannot use `--` to protect against operations supported as an extension (or errors reported for unsupported operations), the nature of the `eval` utility is such that other means can be used to provide this protection (see APPLICATION USAGE above).

## FUTURE DIRECTIONS

None.

## SEE ALSO

Section 2.14, Special Built-In Utilities

## COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see [https://www.kernel.org/doc/man-pages/reporting\\_bugs.html](https://www.kernel.org/doc/man-pages/reporting_bugs.html) .