



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 'dot.1p' command

\$ man dot.1p

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

dot ? execute commands in the current environment

SYNOPSIS

. file

DESCRIPTION

The shell shall execute commands from the file in the current environment.

If file does not contain a <slash>, the shell shall use the search path specified by PATH to find the directory containing file. Unlike normal command search, however, the file searched for by the dot utility need not be executable. If no readable file is found, a non-interactive shell shall abort; an interactive shell shall write a diagnostic message to standard error, but this condition shall not be considered a syntax error.

OPTIONS

None.

OPERANDS

See the DESCRIPTION.

STDIN

Not used.

INPUT FILES

See the DESCRIPTION.

ENVIRONMENT VARIABLES

See the DESCRIPTION.

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 no readable file was found or if the commands in the file could not be parsed, and the shell is interactive (and therefore does not abort; see Section 2.8.1, Consequences of Shell Errors), the exit status shall be non-zero. Otherwise, return the value of the last command executed, or a zero exit status if no command is executed.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

None.

EXAMPLES

```
cat foobar
```

```
foo=hello bar=world
```

```
./foobar
```

```
echo $foo $bar
```

hello world

RATIONALE

Some older implementations searched the current directory for the file, even if the value of PATH disallowed it. This behavior was omitted from this volume of POSIX.1-2017 due to concerns about introducing the susceptibility to trojan horses that the user might be trying to avoid by leaving dot out of PATH.

The KornShell version of dot takes optional arguments that are set to the positional parameters. This is a valid extension that allows a dot script to behave identically to a function.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.14, Special Built-In Utilities, return

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 .