



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 'exec.1p' command

\$ man exec.1p

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

exec ? execute commands and open, close, or copy file descriptors

SYNOPSIS

exec [command [argument...]]

DESCRIPTION

The exec utility shall open, close, and/or copy file descriptors as specified by any redirections as part of the command.

If exec is specified without command or arguments, and any file descriptors with numbers greater than 2 are opened with associated redirection statements, it is unspecified whether those file descriptors remain open when the shell invokes another utility. Scripts concerned that child shells could misuse open file descriptors can always close them explicitly, as shown in one of the following examples.

If exec is specified with command, it shall replace the shell with command without creating a new process. If arguments are specified, they shall be arguments to command. Redirection affects the current shell execution environment.

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 `command` is specified, `exec` shall not return to the shell; rather, the exit status of the process shall be the exit status of the program implementing `command`, which overlaid the shell. If `command` is not found, the exit status shall be 127. If `command` is found, but it is not an executable utility, the exit status shall be 126. If a redirection error occurs (see Section 2.8.1, Consequences of Shell Errors), the shell shall exit with a value in the range 1-125. Otherwise, `exec` shall return a zero exit status.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

None.

EXAMPLES

Open readfile as file descriptor 3 for reading:

```
exec 3< readfile
```

Open writefile as file descriptor 4 for writing:

```
exec 4> writefile
```

Make file descriptor 5 a copy of file descriptor 0:

```
exec 5<&0
```

Close file descriptor 3:

```
exec 3<&-
```

Cat the file maggie by replacing the current shell with the `cat` utility:

ity:

```
exec cat maggie
```

RATIONALE

Most historical implementations were not conformant in that:

```
foo=bar exec cmd
```

did not pass `foo` to `cmd`.

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.

IEEE/The Open Group

2017

EXEC(1P)