



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 'tcgetattr.3p' command

\$ man tcgetattr.3p

TCGETATTR(3P) POSIX Programmer's Manual TCGETATTR(3P)

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

tcgetattr ? get the parameters associated with the terminal

SYNOPSIS

```
#include <termios.h>

int tcgetattr(int fildes, struct termios *termios_p);
```

DESCRIPTION

The tcgetattr() function shall get the parameters associated with the terminal referred to by fildes and store them in the termios structure referenced by termios_p. The fildes argument is an open file descriptor associated with a terminal.

The termios_p argument is a pointer to a termios structure.

The tcgetattr() operation is allowed from any process.

If the terminal device supports different input and output baud rates, the baud rates stored in the termios structure returned by tcgetattr() shall reflect the actual baud rates, even if they are equal. If differing baud rates are not supported, the rate returned as the output baud rate shall be the actual baud rate. If the terminal device does not

support split baud rates, the input baud rate stored in the `termios` structure shall be the output rate (as one of the symbolic values).

RETURN VALUE

Upon successful completion, 0 shall be returned. Otherwise, -1 shall be returned and `errno` set to indicate the error.

ERRORS

The `tcgetattr()` function shall fail if:

EBADF The `fildev` argument is not a valid file descriptor.

ENOTTY The file associated with `fildev` is not a terminal.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

Care must be taken when changing the terminal attributes. Applications should always do a `tcgetattr()`, save the `termios` structure values returned, and then do a `tcsetattr()`, changing only the necessary fields. The application should use the values saved from the `tcgetattr()` to reset the terminal state whenever it is done with the terminal. This is necessary because terminal attributes apply to the underlying port and not to each individual open instance; that is, all processes that have used the terminal see the latest attribute changes.

A program that uses these functions should be written to catch all signals and take other appropriate actions to ensure that when the program terminates, whether planned or not, the terminal device's state is restored to its original state.

Existing practice dealing with error returns when only part of a request can be honored is based on calls to the `ioctl()` function. In historical BSD and System V implementations, the corresponding `ioctl()` returns zero if the requested actions were semantically correct, even if some of the requested changes could not be made. Many existing applications assume this behavior and would no longer work correctly if the

return value were changed from zero to -1 in this case.

Note that either specification has a problem. When zero is returned, it implies everything succeeded even if some of the changes were not made.

When -1 is returned, it implies everything failed even though some of the changes were made.

Applications that need all of the requested changes made to work properly should follow `tcsetattr()` with a call to `tcgetattr()` and compare the appropriate field values.

FUTURE DIRECTIONS

None.

SEE ALSO

`tcsetattr()`

The Base Definitions volume of POSIX.1-2017, Chapter 11, General Terminal Interface, `<termios.h>`

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 .