



## ***Red Hat Enterprise Linux Release 9.2 Manual Pages on 'cfgetispeed.3p' command***

***\$ man cfgetispeed.3p***

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

cfgetispeed ? get input baud rate

### SYNOPSIS

```
#include <termios.h>

speed_t cfgetispeed(const struct termios *termios_p);
```

### DESCRIPTION

The cfgetispeed() function shall extract the input baud rate from the termios structure to which the termios\_p argument points.

This function shall return exactly the value in the termios data structure, without interpretation.

### RETURN VALUE

Upon successful completion, cfgetispeed() shall return a value of type speed\_t representing the input baud rate.

### ERRORS

No errors are defined.

The following sections are informative.

### EXAMPLES

None.

## APPLICATION USAGE

None.

## RATIONALE

The term "baud" is used historically here, but is not technically correct. This is properly "bits per second", which may not be the same as baud. However, the term is used because of the historical usage and understanding.

The `cfgetospeed()`, `cfgetispeed()`, `cfsetospeed()`, and `cfsetispeed()` functions do not take arguments as numbers, but rather as symbolic names. There are two reasons for this:

1. Historically, numbers were not used because of the way the rate was stored in the data structure. This is retained even though a function is now used.
2. More importantly, only a limited set of possible rates is at all portable, and this constrains the application to that set.

There is nothing to prevent an implementation accepting as an extension a number (such as 126), and since the encoding of the `Bxxx` symbols is not specified, this can be done to avoid introducing ambiguity.

Setting the input baud rate to zero was a mechanism to allow for split baud rates. Clarifications in this volume of POSIX.1-2017 have made it possible to determine whether split rates are supported and to support them without having to treat zero as a special case. Since this functionality is also confusing, it has been declared obsolescent. The 0 argument referred to is the literal constant 0, not the symbolic constant `B0`. This volume of POSIX.1-2017 does not preclude `B0` from being defined as the value 0; in fact, implementations would likely benefit from the two being equivalent. This volume of POSIX.1-2017 does not fully specify whether the previous `cfsetispeed()` value is retained after a `tcgetattr()` as the actual value or as zero. Therefore, conforming applications should always set both the input speed and output speed when setting either.

In historical implementations, the baud rate information is tradition?

ally kept in `c_cflag`. Applications should be written to presume that this might be the case (and thus not blindly copy `c_cflag`), but not to rely on it in case it is in some other field of the structure. Setting the `c_cflag` field absolutely after setting a baud rate is a non-portable action because of this. In general, the unused parts of the flag fields might be used by the implementation and should not be blindly copied from the descriptions of one terminal device to another.

## FUTURE DIRECTIONS

None.

## SEE ALSO

`cfgetospeed()`, `cfsetispeed()`, `cfsetospeed()`, `tcgetattr()`

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](https://www.kernel.org/doc/man-pages/reporting_bugs.html).