NAME

termios.h - define values for termios

SYNOPSIS

```
#include <termios.h>
```

DESCRIPTION

The *<termios.h>* header contains the definitions used by the terminal I/O interfaces (see the **XBD** specification, **General Terminal Interface** for the structures and names defined).

The termios Structure

The following data types are defined through **typedef**:

cc_t
     Used for terminal special characters.
speed_t
     Used for terminal baud rates.
tcflag_t
     Used for terminal modes.

The above types are all unsigned integral types.

The **termios** structure is defined, and includes at least the following members:

```
tcflag_t  c_iflag     input modes
tcflag_t  c_oflag     output modes
tcflag_t  c_cflag     control modes
tcflag_t  c_lflag     local modes
cc_t      c_cc[NCCS]  control chars
```

A definition is given for:

NCCS
     Size of the array **c_cc** for control characters.

The following subscript names for the array **c_cc** are defined:

Subscript Usage		
Canonical Mode	**Non-canonical Mode**	**Description**
VEOF		EOF character
VEOL		EOL character
VERASE		ERASE character
VINTR	VINTR	INTR character

VKILL		KILL character
	VMIN	MIN value
VQUIT	VQUIT	QUIT character
VSTART	VSTART	START character
VSTOP	VSTOP	STOP character
VSUSP	VSUSP	SUSP character
	VTIME	TIME value

The subscript values are unique, except that the VMIN and VTIME subscripts may have the same values as the VEOF and VEOL subscripts, respectively.

Input Modes

The **c_iflag** field describes the basic terminal input control:

BRKINT
    Signal interrupt on break.
ICRNL
    Map CR to NL on input.
IGNBRK
    Ignore break condition.
IGNCR
    Ignore CR
IGNPAR
    Ignore characters with parity errors.
INLCR
    Map NL to CR on input.
INPCK
    Enable input parity check.
ISTRIP
    Strip character
IUCLC
    Map upper-case to lower-case on input (**LEGACY**).
IXANY
    Enable any character to restart output.
IXOFF
    Enable start/stop input control.
IXON
    Enable start/stop output control.
PARMRK
    Mark parity errors.

Output Modes

The **c_oflag** field specifies the system treatment of output:

OPOST
    Post-process output
OLCUC
    Map lower-case to upper-case on output (**LEGACY**).
ONLCR
    Map NL to CR-NL on output.
OCRNL
    Map CR to NL on output.
ONOCR
    No CR output at column 0.

ONLRET
> NL performs CR function.

OFILL
> Use fill characters for delay.

NLDLY
> Select newline delays:
>
> **NL0**
> > Newline character type 0.
>
> **NL1**
> > Newline character type 1.

CRDLY
> Select carriage-return delays:
>
> **CR0**
> > Carriage-return delay type 0.
>
> **CR1**
> > Carriage-return delay type 1.
>
> **CR2**
> > Carriage-return delay type 2.
>
> **CR3**
> > Carriage-return delay type 3.

TABDLY
> Select horizontal-tab delays:
>
> **TAB0**
> > Horizontal-tab delay type 0.
>
> **TAB1**
> > Horizontal-tab delay type 1.
>
> **TAB2**
> > Horizontal-tab delay type 2.
>
> **TAB3**
> > Expand tabs to spaces.

BSDLY
> Select backspace delays:
>
> **BS0**
> > Backspace-delay type 0.
>
> **BS1**
> > Backspace-delay type 1.

VTDLY
> Select vertical-tab delays:
>
> **VT0**
> > Vertical-tab delay type 0.
>
> **VT1**
> > Vertical-tab delay type 1.

FFDLY
> Select form-feed delays:
>
> **FF0**
> > Form-feed delay type 0.
>
> **FF1**
> > Form-feed delay type 1.

The input and output baud rates are stored in the **termios** structure. These are the valid values for objects of type **speed_t**. The following values are defined, but not all baud rates need be supported by the underlying hardware.

B0
> Hang up
B50
> 50 baud
B75
> 75 baud
B110
> 110 baud
B134
> 134.5 baud
B150
> 150 baud
B200
> 200 baud
B300
> 300 baud
B600
> 600 baud
B1200
> 1200 baud
B1800
> 1800 baud
B2400
> 2400 baud
B4800
> 4800 baud
B9600
> 9600 baud
B19200
> 19200 baud
B38400
> 38400 baud

Control Modes

The **c_cflag** field describes the hardware control of the terminal; not all values specified are required to be supported by the underlying hardware:

CSIZE
> Character size:
>
> CS5
> > 5 bits.
> CS6
> > 6 bits.
> CS7
> > 7 bits.
> CS8
> > 8 bits.

CSTOPB
> Send two stop bits, else one.

CREAD
        Enable receiver.

PARENB
        Parity enable.

PARODD
        Odd parity, else even.

HUPCL
        Hang up on last close.

CLOCAL
        Ignore modem status lines.

Local Modes

The **c_lflag** field of the argument structure is used to control various terminal functions:

ECHO
        Enable echo.
ECHOE
        Echo erase character as error-correcting backspace.
ECHOK
        Echo KILL.
ECHONL
        Echo NL.
ICANON
        Canonical input (erase and kill processing).
IEXTEN
        Enable extended input character processing.
ISIG
        Enable signals.
NOFLSH
        Disable flush after interrupt or quit.
TOSTOP
        Send SIGTTOU for background output.
XCASE
        Canonical upper/lower presentation (**LEGACY**).

Attribute Selection

The following symbolic constants for use with *tcsetattr()* are defined:

TCSANOW
        Change attributes immediately.
TCSADRAIN
        Change attributes when output has drained.
TCSAFLUSH
        Change attributes when output has drained; also flush pending input.

Line Control

The following symbolic constants for use with *tcflush()* are defined:

TCIFLUSH
        Flush pending input. Flush untransmitted output.

TCIOFLUSH
    Flush both pending input and untransmitted output.
TCOFLUSH
    Flush untransmitted output.

The following symbolic constants for use with *tcflow()* are defined:

TCIOFF
    Transmit a STOP character, intended to suspend input data.
TCION
    Transmit a START character, intended to restart input data.
TCOOFF
    Suspend output.
TCOON
    Restart output.

The following are declared as functions and may also be defined as macros. Function prototypes must be provided for use with an ISO C compiler.

```
speed_t cfgetispeed(const struct termios *);
speed_t cfgetospeed(const struct termios *);
int     cfsetispeed(struct termios *, speed_t);
int     cfsetospeed(struct termios *, speed_t);
int     tcdrain(int);
int     tcflow(int, int);
int     tcflush(int, int);
int     tcgetattr(int, struct termios *);
pid_t   tcgetsid(int);
int     tcsendbreak(int, int);
int     tcsetattr(int, int, struct termios *);
```

APPLICATION USAGE

The following names are commonly used as extensions to the above, therefore portable applications must not use them:

CBAUD	EXTB	VDSUSP
DEFECHO	FLUSHO	VLNEXT
ECHOCTL	LOBLK	VREPRINT
ECHOKE	PENDIN	VSTATUS
ECHOPRT	SWTCH	VWERASE
EXTA	VDISCARD	

FUTURE DIRECTIONS

None.

SEE ALSO

cfgetispeed(), *cfgetospeed()*, *cfsetispeed()*, *cfsetospeed()*, *tcdrain()*, *tcflow()*, *tcflush()*, *tcgetattr()*, *tcgetsid()*, *tcsendbreak()*, *tcsetattr()*, the **XBD** specification, **General Terminal Interface** .