

addch(3NCURSES)

addch(3NCURSES)

## NAME

**addch**, **waddch**, **mvaddch**, **mvwaddch**, **echochar**, **wechochar** — add a character (with attributes) to a **curses** window, then advance the cursor

## SYNOPSIS

```
#include <curses.h>
```

```
int addch(const chtype ch);
```

```
int waddch(WINDOW *win, const chtype ch);
```

```
int mvaddch(int y, int x, const chtype ch);
```

```
int mvwaddch(WINDOW *win, int y, int x, const chtype ch);
```

```
int echochar(const chtype ch);
```

```
int wechochar(WINDOW *win, const chtype ch);
```

## DESCRIPTION

### Adding characters

The **addch**, **waddch**, **mvaddch** and **mvwaddch** routines put the character *ch* into the given window at its current window position, which is then advanced. They are analogous to **putchar** in **stdio(3)**. If the advance is at the right margin:

- The cursor automatically wraps to the beginning of the next line.
- At the bottom of the current scrolling region, and if **scrollok** is enabled, the scrolling region is scrolled up one line.
- If **scrollok** is not enabled, writing a character at the lower right margin succeeds. However, an error is returned because it is not possible to wrap to a new line

If *ch* is a tab, newline, carriage return or backspace, the cursor is moved appropriately within the window:

- Backspace moves the cursor one character left; at the left edge of a window it does nothing.
- Carriage return moves the cursor to the window left margin on the current line.
- Newline does a **clrtoeol**, then moves the cursor to the window left margin on the next line, scrolling the window if on the last line.
- Tabs are considered to be at every eighth column. The tab interval may be altered by setting the **TABSIZE** variable.

If *ch* is any other control character, it is drawn in **^X** notation. Calling **winch** after adding a control character does not return the character itself, but instead returns the **^**-representation of the control character.

Video attributes can be combined with a character argument passed to **addch** or related functions by logical-ORing them into the character. (Thus, text, including attributes, can be copied from one place to another using **inch** and **addch**.) See the **attr(3NCURSES)** page for values of predefined video attribute constants that can be usefully OR'ed into characters.

### Echoing characters

The **echochar** and **wechochar** routines are equivalent to a call to **addch** followed by a call to **refresh**, or a call to **waddch** followed by a call to **wrefresh**. The knowledge that only a single character is being output is used and, for non-control characters, a considerable performance gain may be seen by using these routines instead of their equivalents.

### Line Graphics

The following variables may be used to add line drawing characters to the screen with routines of the **addch** family. The default character listed below is used if the **acsc** capability does not define a terminal-specific replacement for it, or if the terminal and locale configuration requires Unicode but the library is unable to use Unicode.

The names are taken from VT100 nomenclature.

<i>Name</i>	<i>Default</i>	<i>Description</i>
ACS_BLOCK	#	solid square block
ACS_BOARD	#	board of squares
ACS_BTEE	+	bottom tee



addch(3NCURSES)

addch(3NCURSES)

ACS_BULLET	o	bullet
ACS_CKBOARD	:	checker board (stipple)
ACS_DARROW	v	arrow pointing down
ACS_DEGREE	'	degree symbol
ACS_DIAMOND	+	diamond
ACS_GEQUAL	>	greater-than-or-equal-to
ACS_HLINE	—	horizontal line
ACS_LANTERN	#	lantern symbol
ACS_LARROW	<	arrow pointing left
ACS_LEQUAL	<	less-than-or-equal-to
ACS_LLCORNER	+	lower left-hand corner
ACS_LRCORNER	+	lower right-hand corner
ACS_LTEE	+	left tee
ACS_NEQUAL	!	not-equal
ACS_PI	*	greek pi
ACS_PLMINUS	#	plus/minus
ACS_PLUS	+	plus
ACS_RARROW	>	arrow pointing right
ACS_RTEE	+	right tee
ACS_S1	—	scan line 1
ACS_S3	—	scan line 3
ACS_S7	—	scan line 7
ACS_S9	—	scan line 9
ACS_STERLING	f	pound-sterling symbol
ACS_TTEE	+	top tee
ACS_UARROW	^	arrow pointing up
ACS_ULCORNER	+	upper left-hand corner
ACS_URCORNER	+	upper right-hand corner
ACS_VLINE		vertical line

## RETURN VALUE

All routines return the integer **ERR** upon failure and **OK** on success (the SVr4 manuals specify only "an integer value other than **ERR**") upon successful completion, unless otherwise noted in the preceding routine descriptions.

Functions with a "mv" prefix first perform a cursor movement using **wmove**, and return an error if the position is outside the window, or if the window pointer is null.

## NOTES

Note that **addch**, **mvaddch**, **mvwaddch**, and **echochar** may be macros.

## PORTABILITY

All these functions are described in the XSI Curses standard, Issue 4. The defaults specified for forms-drawing characters apply in the POSIX locale.

X/Open Curses states that the **ACS\_** definitions are **char** constants. For the wide-character implementation (see  **curs\_add\_wch**), there are analogous **WACS\_** definitions which are **cchar\_t** constants.

Some ACS symbols (**ACS\_S3**, **ACS\_S7**, **ACS\_LEQUAL**, **ACS\_GEQUAL**, **ACS\_PI**, **ACS\_NEQUAL**, **ACS\_STERLING**) were not documented in any publicly released System V. However, many publicly available terminfos include **acsc** strings in which their key characters (**pryz{||}**) are embedded, and a second-hand list of their character descriptions has come to light. The ACS-prefixed names for them were invented for **ncurses(3NCURSES)**.

The *displayed* values for the **ACS\_** and **WACS\_** constants depend on

- the library configuration, i.e., **ncurses** versus **ncursesw**, where the latter is capable of displaying Unicode while the former is not, and
- whether the *locale* uses UTF-8 encoding.

In certain cases, the terminal is unable to display line-drawing characters except by using UTF-8 (see the discussion of **NCURSES\_NO\_UTF8\_ACS** in **ncurses(3X)**).

The **TABSIZE** variable is implemented in some versions of curses, but is not part of X/Open curses.



`addch(3NCURSES)``addch(3NCURSES)`

If *ch* is a carriage return, the cursor is moved to the beginning of the current row of the window. This is true of other implementations, but is not documented.

**SEE ALSO**

**`ncurses(3NCURSES)`**, **`attr(3NCURSES)`**, **`clear(3NCURSES)`**, **`inch(3NCURSES)`**, **`outopts(3NCURSES)`**, **`refresh(3NCURSES)`**, **`curses_variables(3NCURSES)`**, **`putc(3)`**.

Comparable functions in the wide-character (`ncursesw`) library are described in **`add_wch(3NCURSES)`**.

