

**NAME**

**get\_wstr**, **getn\_wstr**, **wget\_wstr**, **wgetn\_wstr**, **mvget\_wstr**, **mvgetn\_wstr**, **mvwget\_wstr**, **mvwgetn\_wstr** – read a wide-character string from a *curses* terminal keyboard

**SYNOPSIS**

```
#include <curses.h>

int get_wstr(wint_t * wstr);
int wget_wstr(WINDOW * win, wint_t * wstr);
int mvget_wstr(int y, int x, wint_t * wstr);
int mvwget_wstr(WINDOW * win, int y, int x, wint_t * wstr);

int getn_wstr(wint_t * wstr, int n);
int wgetn_wstr(WINDOW * win, wint_t * wstr, int n);
int mvgetn_wstr(int y, int x, wint_t * wstr, int n);
int mvwgetn_wstr(WINDOW * win, int y, int x, wint_t * wstr, int n);
```

**DESCRIPTION**

**wget\_wstr** populates a user-supplied wide-character string buffer *wstr* by repeatedly calling **wget\_wch(3X)** with the *win* argument until a line feed or carriage return character is input. The function

- does not copy the terminating character to *wstr*;
- populates *wstr* with *WEOF* (as defined in *wchar.h*) if an end-of-file condition occurs on the input;
- always terminates the string with a null wide character (after any *WEOF*);
- interprets the screen’s wide erase and wide kill characters (see **eraseswchar(3X)** and **killwchar(3X)**);
- recognizes function keys only if the screen’s keypad option is enabled (see **keypad(3X)**);
- treats the function keys **KEY\_LEFT** and **KEY\_BACKSPACE** the same as the wide erase character; and
- discards function key inputs other than those treated as the wide erase or wide kill characters, calling **beep(3X)**.

The wide erase character replaces the character at the end of the buffer with a null wide character, while the wide kill character does the same for the entire buffer.

If the screen’s echo option is enabled (see **echo(3X)**), **wget\_wstr** updates *win* with **wadd\_wch(3X)**. Further,

- the wide erase character and its function key synonyms move the cursor to the left, and
- the wide kill character returns the cursor to where it was located when **wget\_wstr** was called.

**wgetn\_wstr** is similar, but reads at most *n* wide characters, aiding the application to avoid overrunning the buffer to which *wstr* points. *curses* ignores an attempt to input more than *n* wide characters (other than the terminating line feed or carriage return), calling **beep(3X)**. If *n* is negative, **wgetn\_wstr** reads up to *LINE\_MAX* wide characters (see *sysconf(3)*).

**ncurses(3X)** describes the variants of these functions.

**RETURN VALUE**

These functions return **OK** on success and **ERR** on failure.

In *ncurses*, these functions fail if

- the *curses* screen has not been initialized,
- (for functions taking a *WINDOW* pointer argument) *win* is a null pointer,
- *wstr* is a null pointer, or
- an internal **wget\_wch(3X)** call fails.

Functions prefixed with “mv” first perform cursor movement and fail if the position (*y*, *x*) is outside the window boundaries.

## NOTES

All of these functions except **wgetn\_wstr** may be implemented as macros.

Reading input that overruns the buffer pointed to by *wstr* causes undefined results. Use the **n**-infix functions, and allocate sufficient storage for *wstr* — at least  $n+1$  times **sizeof(wchar\_t)**.

These functions cannot store a **KEY\_** value in *wstr* because there is no way to distinguish it from a valid *wchar\_t* value.

While these functions conceptually implement a series of calls to **wget\_wch**, they also temporarily change properties of the *curses* screen to permit simple editing of the input buffer. Each function saves the screen's state, calls **nl(3X)**, and, if the screen was in canonical ("cooked") mode, **cbreak(3X)**. Before returning, it restores the saved screen state. Other implementations differ in detail, affecting which control characters they can accept in the buffer; see section "PORTABILITY" below.

Unlike **getstr(3X)** and related functions of *ncurses*'s non-wide API, these functions do not return **KEY\_RESIZE** if a *SIGWINCH* event interrupts the function.

## EXTENSIONS

**getn\_wstr**, **wgetn\_wstr**, **mvgetn\_wstr**, and **mvwgetn\_wstr**'s handling of negative *n* values is an *ncurses* extension.

## PORTABILITY

Applications employing *ncurses* extensions should condition their use on the visibility of the **NCURSES\_VERSION** preprocessor macro.

X/Open Curses Issue 4 describes these functions. It specifies no error conditions for them.

Issue 4 documented these functions as passing an array of *wchar\_t*, but that was an error, conflicting with the following language in the standard.

The effect of *get\_wstr()* is as though a series of calls to *get\_wch()* were made, until a newline character, end-of-line character, or end-of-file character is processed.

*get\_wch* can return a negative value (*WEOF*), but *wchar\_t* is a unsigned type. All of the vendors implement these functions using *wint\_t*, following the Issue 7 standard.

X/Open Curses Issue 7 is unclear whether the terminating null wide character counts toward the length parameter *n*. A similar issue affected *wgetnstr* in Issue 4, Version 2; Issue 7 revised that function's description to address the issue, but not that of *wget\_nwstr*, leaving it ambiguous. *ncurses* counts the terminator in the length.

X/Open Curses does not specify what happens if the length *n* is negative.

- For consistency with *wgetnstr*, *ncurses* 6.2 uses a limit based on *LINE\_MAX*.
- Some other implementations (such as Solaris *xcurses*) do the same, while others (*PDCurses*) do not permit a negative *n*.
- NetBSD 7 *curses* imitates *ncurses* 6.1 and earlier, treating a negative *n* as an unbounded count of wide characters.

Implementations vary in their handling of input control characters.

- While they may enable the screen's echo option, some do not take it out of raw mode, and may take *cbreak* mode into account when deciding whether to handle echoing within *wgetn\_wstr* or to rely on it as a side effect of calling *wget\_wch*.

Since 1995, *ncurses* has provided handlers for *SIGINTR* and *SIGQUIT* events, which are typically generated at the keyboard with **^C** and **^\\** respectively. In *cbreak* mode, those handlers catch a signal and stop the program, whereas other implementations write those characters into the buffer.

- Starting with *ncurses* 6.3 (2021), *wgetn\_wstr* preserves raw mode if the screen was already in that state, allowing one to enter the characters the terminal interprets as interrupt and quit events into the buffer, for consistency with SVr4 *curses*'s *wgetnstr*.

**HISTORY**

X/Open Curses Issue 4 (1995) initially specified these functions. The System V Interface Definition Version 4 of the same year specified functions named *wgetwstr* and *wgetnwstr* (and the usual variants). These were later additions to SVr4.x, not appearing in the first SVr4 (1989). Except in name, their declarations did not differ from X/Open's later *wget\_wstr* and *wgetn\_wstr* until X/Open Curses Issue 7 (2009) eventually changed the type of the buffer argument to a pointer to *wint\_t*.

**SEE ALSO**

**curs\_getstr(3X)** describes comparable functions of the *ncurses* library in its non-wide-character configuration.

**curses(3X)**, **curs\_get\_wch(3X)**