

**NAME**

**get\_wstr**, **getn\_wstr**, **wget\_wstr**, **wgetn\_wstr**, **mvget\_wstr**, **mvgetn\_wstr**, **mvwget\_wstr**, **mvwgetn\_wstr**  
 - get an array of wide characters from a curses terminal keyboard

**SYNOPSIS**

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

```
int get_wstr(wint_t *wstr);
int getn_wstr(wint_t *wstr, int n);
int wget_wstr(WINDOW *win, wint_t *wstr);
int wgetn_wstr(WINDOW *win, wint_t *wstr, int n);

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

**DESCRIPTION**

The effect of **get\_wstr** is as though a series of calls to **get\_wch**(3X) were made, until a newline, other end-of-line, or end-of-file condition is processed. An end-of-file condition is represented by **WEOF**, as defined in **<wchar.h>**. The newline and end-of-line conditions are represented by the **\n wchar\_t** value. In all instances, the end of the string is terminated by a null **wchar\_t**. The routine places resulting values in the area pointed to by *wstr*.

The user's erase and kill characters are interpreted. If keypad mode is on for the window, **KEY\_LEFT** and **KEY\_BACKSPACE** are both considered equivalent to the user's kill character.

Characters input are echoed only if **echo** is currently on. In that case, backspace is echoed as deletion of the previous character (typically a left motion).

The effect of **wget\_wstr** is as though a series of calls to **wget\_wch** were made.

The effect of **mvget\_wstr** is as though a call to **move** and then a series of calls to **get\_wch** were made.

The effect of **mvwget\_wstr** is as though a call to **wmove** and then a series of calls to **wget\_wch** were made.

The **getn\_wstr**, **mvgetn\_wstr**, **mvwgetn\_wstr**, and **wgetn\_wstr** functions are identical to the **get\_wstr**, **mvget\_wstr**, **mvwget\_wstr**, and **wget\_wstr** functions, respectively, except that the **\*n\*** versions read at most *n* characters, letting the application prevent overflow of the input buffer.

## NOTES

Using **get\_wstr**, **mvget\_wstr**, **mvwget\_wstr**, or **wget\_wstr** to read a line that overflows the array pointed to by **wstr** causes undefined results. The use of **getn\_wstr**, **mvgetn\_wstr**, **mvwgetn\_wstr**, or **wgetn\_wstr**, respectively, is recommended.

These functions cannot return **KEY\_** values because there is no way to distinguish a **KEY\_** value from a valid **wchar\_t** value.

All of these routines except **wgetn\_wstr** may be macros.

## RETURN VALUE

All of these functions return **OK** upon successful completion. Otherwise, they return **ERR**.

Functions using a window parameter return an error if it is null.

### **wgetn\_wstr**

returns an error if the associated call to **wget\_wch** failed.

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.

## PORTABILITY

These functions are described in The Single Unix Specification, Version 2. No error conditions are defined. This implementation returns **ERR** if the window pointer is null, or if the lower-level **wget\_wch** call returns an **ERR**. In the latter case, an **ERR** return without other data is treated as an end-of-file condition, and the returned array contains a **WEOF** followed by a null **wchar\_t**.

X/Open curses documented these functions to pass an array of **wchar\_t** in 1997, but that was an error because of this part of the description:

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.

The latter function *get\_wch()* can return a negative value, while **wchar\_t** is a unsigned type. All of the vendors implement this using **wint\_t**, following the standard.

X/Open Curses, Issue 7 (2009) is unclear regarding whether the terminating *null* **wchar\_t** value is counted in the length parameter *n*. X/Open Curses, Issue 7 revised the corresponding description of **wgetnstr** to address this issue. The unrevised description of **wget\_nwstr** can be interpreted either way. This implementation counts the terminator in the length.

`curs_get_wstr(3X)`

`curs_get_wstr(3X)`

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

- ⊕ For analogy 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 allow this.
- ⊕ NetBSD 7 curses imitates ncurses 6.1 in this regard, treating a **-1** as an indefinite number of characters.

## SEE ALSO

Functions: **curses(3X)**, **curs\_get\_wch(3X)**, **curs\_getstr(3X)**.

`curs_get_wstr(3X)`