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** 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 VALUES

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 documents these functions to pass an array of **wchar_t**, but all of the vendors implement this using **wint_t**.

SEE ALSO

Functions: **ncurses**(3NCURSES), **get_wch**(3NCURSES), **getstr**(3NCURSES).