

NAME

getcchar, **setcchar** – convert between a wide-character string and a *curses* complex character

SYNOPSIS

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

```
int getcchar(const cchar_t * wch, wchar_t * wc,
             attr_t * attrs, short * pair, void * opts);
int setcchar(cchar_t * wch, const wchar_t * wc,
             const attr_t attrs, short pair, const void * opts);
```

DESCRIPTION

The *curses* complex character data type *cchar_t* is a structure type comprising a wide-character string, a set of attributes, and a color pair identifier. The *cchar_t* structure is opaque; do not attempt to access its members directly. The library provides functions to manipulate this type.

getcchar

getcchar deconstructs a *cchar_t* into its components.

If *wc* is not a null pointer, **getcchar**:

- stores the wide-character string in the *curses* complex character *wch* into *wc*;
- stores the attributes in *attrs*; and
- stores the color pair identifier in *pair*.

If *wc* is a null pointer, **getcchar** counts the *wchar_t* wide characters in *wch*, returns that value, and leaves *attrs* and *pair* unchanged.

setcchar

setcchar constructs a *curses* complex character *wch* from the components *wc*, *attrs*, and *pair*. The wide-character string *wch* must be terminated with a null wide character **L'\0'** and must contain at most one spacing character, which, if present, must be the first wide character in the string.

Up to **CCHARW_MAX** – 1 non-spacing characters may follow (see **curs_variables(3X)**). *ncurses* ignores any additional non-spacing characters.

The string may contain a single control character instead. In that case, no non-spacing characters are allowed.

RETURN VALUE

If **getcchar** is passed a null pointer as its *wc* argument, it returns the number of wide characters for a given *wch* that it would store in *wc*, counting a trailing null wide character. If **getcchar** is not passed a null pointer as its *wc* argument, it returns **OK** on success and **ERR** on failure.

In *ncurses*, **getcchar** returns **ERR** if either *attrs* or *pair* is a null pointer and *wc* is not.

setcchar returns **OK** on success and **ERR** on failure.

In *ncurses*, **setcchar** returns **ERR** if

- *wch* is a null pointer,
- *wc* starts with a (wide) control character and contains any other wide characters, or
- *pair* has a negative value.

NOTES

wch may be a value stored by **setcchar** or another *curses* function with a writable *cchar_t* argument. If *wch* is constructed by any other means, the library's behavior is unspecified.

EXTENSIONS

X/Open Curses documents the *opts* argument as reserved for future use, saying that it must be a null pointer. The *ncurses* 6 ABI uses it with functions that have a color pair parameter to support extended color pairs.

- In functions that assign colors, such as **setcchar**, if *opts* is not a null pointer, *ncurses* treats it as a pointer to *int*, and interprets it instead of the *short pair* parameter as a color pair identifier.
- In functions that retrieve colors, such as **getcchar**, if *opts* is not a null pointer, *ncurses* treats it as a pointer to *int*, and stores the retrieved color pair identifier there as well as in the *short pair* parameter (which may therefore undergo a narrowing conversion).

PORTABILITY

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

These functions are described in X/Open Curses Issue 4. It specifies no error conditions for them.

X/Open Curses does not detail the layout of the *cchar_t* structure, describing only its minimal required contents:

- a spacing wide character (*wchar_t*),
- at least five non-spacing wide characters (*wchar_t*; see below),
- attributes (at least 15 bits' worth, inferred from the count of specified *WA_* constants),
- a color pair identifier (at least 16 bits, inferred from the *short* type used to encode it).

Non-spacing characters are optional, in the sense that zero or more may be stored in a *cchar_t*. X/Open/Curses specifies a limit:

Implementations may limit the number of non-spacing characters that can be associated with a spacing character, provided any limit is at least 5.

Then-contemporary Unix implementations adhered to that limit.

- AIX 4 and OSF/1 4 used the same declaration with a single spacing wide character *c* and an array of 5 non-spacing wide characters *z*.
- HP-UX 10 used an opaque structure of 28 bytes, large enough for 6 *wchar_t* values.
- Solaris *xcurses* uses a single array of 6 *wchar_t* values.

ncurses defined its *cchar_t* in 1995 using 5 as the *total* of spacing and non-spacing characters (**CCHARW_MAX**). That was probably due to a misreading of the AIX 4 header files, because the X/Open Curses document was not generally available at that time. Later (in 2002), this detail was overlooked when work began to implement the functions using the structure.

In practice, a mere four non-spacing characters may seem adequate. X/Open Curses documents possible applications of non-spacing characters, including their use as ligatures (a feature apparently not supported by any *curses* implementation). Unicode does not limit the (analogous) number of combining characters in a grapheme cluster; some applications may be affected. *ncurses* can be compiled with a different **CCHARW_MAX** value; doing so alters the library's ABI.

HISTORY

X/Open Curses Issue 4 (1995) initially specified these functions.

SEE ALSO

curses(3X), **curs_attr(3X)**, **curs_color(3X)**, **wcwidth(3)**