

**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 *color_pair,  
    void *opts );
```

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

**DESCRIPTION****getcchar**

The **getcchar** function gets a wide-character string and rendition from a **cchar\_t** argument. When *wc* is not a null pointer, the **getcchar** function does the following:

- ⊕ Extracts information from a **cchar\_t** value *wch*
- ⊕ Stores the character attributes in the location pointed to by *attrs*
- ⊕ Stores the color pair in the location pointed to by *color\_pair*
- ⊕ Stores the wide-character string, characters referenced by *wch*, into the array pointed to by *wc*.

When *wc* is a null pointer, the **getcchar** function does the following:

- ⊕ Obtains the number of wide characters pointed to by *wch*
- ⊕ Does not change the data referenced by *attrs* or *color\_pair*

**setcchar**

The **setcchar** function initializes the location pointed to by *wch* by using:

- ⊕ The character attributes in *attrs*
- ⊕ The color pair in *color\_pair*
- ⊕ The wide-character string pointed to by *wc*. The string must be L'\0' terminated, contain at most one spacing character, which must be the first.

Up to **CCHARW\_MAX**-1 non-spacing characters may follow. Additional non-spacing characters are ignored.

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

## RETURN VALUE

When *wc* is a null pointer, **getcchar** returns the number of wide characters referenced by *wch*, including one for a trailing null.

When *wc* is not a null pointer, **getcchar** returns **OK** upon successful completion, and **ERR** otherwise.

Upon successful completion, **setcchar** returns **OK**. Otherwise, it returns **ERR**.

## NOTES

The *wch* argument may be a value generated by a call to **setcchar** or by a function that has a **cchar\_t** output argument. If *wch* is constructed by any other means, the effect is unspecified.

## EXTENSIONS

X/Open Curses documents the *opts* argument as reserved for future use, saying that it must be null. This implementation uses that parameter in ABI 6 for the functions which have a color pair parameter to support extended color pairs:

- ⊕ For functions which modify the color, e.g., **setcchar**, if *opts* is set it is treated as a pointer to **int**, and used to set the color pair instead of the **short** pair parameter.
- ⊕ For functions which retrieve the color, e.g., **getcchar**, if *opts* is set it is treated as a pointer to **int**, and used to retrieve the color pair as an **int** value, in addition retrieving it via the standard pointer to **short** parameter.

## PORTABILITY

The **CCHARW\_MAX** symbol is specific to *ncurses*. X/Open Curses does not provide details for the layout of the **cchar\_t** structure. It tells what data are stored in it:

- ⊕ a spacing character (**wchar\_t**, i.e., 32-bits).
- ⊕ non-spacing characters (again, **wchar\_t**'s).
- ⊕ attributes (at least 16 bits, inferred from the various ACS- and WACS-flags).
- ⊕ color pair (at least 16 bits, inferred from the **unsigned short** type).

The non-spacing characters are optional, in the sense that zero or more may be stored in a **cchar\_t**. XOpen/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.

The Unix implementations at the time follow that limit:

- ⊕ AIX 4 and OSF1 4 use the same declaration with an array of 5 non-spacing characters *z* and a single spacing character *c*.
- ⊕ HP-UX 10 uses an opaque structure with 28 bytes, which is large enough for the 6 **wchar\_t** values.
- ⊕ Solaris *xpg4* curses uses a single array of 6 **wchar\_t** values.

This implementation's **cchar\_t** was defined in 1995 using **5** for 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 beginning to implement the functions using the structure.

In practice, even four non-spacing characters may seem enough. X/Open Curses documents possible uses for non-spacing characters, including using them for ligatures between characters (a feature apparently not supported by any curses implementation). Unicode does not limit the (analogous) number of combining characters, so some applications may be affected.

## SEE ALSO

**curses(3X)**, **curs\_attr(3X)**, **curs\_color(3X)**, **wcwidth(3)**