

**NAME**

**inch**, **winch**, **mvinch**, **mvwinch** - get a *curses* character from a window

**SYNOPSIS**

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

```
chtype inch(void);
```

```
chtype winch(WINDOW *win);
```

```
chtype mvinch(int y, int x);
```

```
chtype mvwinch(WINDOW *win, int y, int x);
```

**DESCRIPTION**

These routines return the character, of type **chtype**, at the current position in the named window. If any attributes are set for that position, their values are OR'ed into the value returned. Constants defined in **<curses.h>** can be used with the **&** (logical AND) operator to extract the character or attributes alone.

**Attributes**

The following bit masks may be AND-ed with characters returned by **winch**.

Name	Description
<b>A_CHARTEXT</b>	Extract character
<b>A_ATTRIBUTES</b>	Extract attributes
<b>A_COLOR</b>	Extract color pair information

**RETURN VALUE**

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

The **winch** function does not return an error if the window contains characters larger than 8-bits (255). Only the low-order 8 bits of the character are used by **winch**.

**NOTES**

Note that all of these routines may be macros.

**PORTABILITY**

These functions are described in X/Open Curses, Issue 4.

Very old systems (before standardization) provide a different function with the same name:

- ⊕ The **winch** function was part of the original BSD curses library, which stored a 7-bit character combined with the *standout* attribute.

In BSD curses, **winch** returned only the character (as an integer) with the *standout* attribute removed.

- ⊕ System V curses added support for several video attributes which could be combined with characters in the window.

Reflecting this improvement, the function was altered to return the character combined with all video attributes in a **chtype** value.

X/Open Curses does not specify the size and layout of attributes, color and character values in **chtype**; it is implementation-dependent. This implementation uses 8 bits for character values. An application using more bits, e.g., a Unicode value, should use the wide-character equivalents to these functions.

## SEE ALSO

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

**curses**(3X), **curs\_instr**(3X)