**Template for comments and secretariat observations**

| Date: | Document: | Project: |
|---|---|---|

| MB/ NC[1] | Line number (e.g. 17) | Clause/ Subclause (e.g. 3.1) | Paragraph/ Figure/ Table/ (e.g. Table 1) | Type of comment[2] | Comments | Proposed change | Observations of the secretariat |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | 7.23.6.1 | 8 | TE | AG Reference Bug 1647 (https://austingroupbugs.net/view.php?id=1647)<br>=======<br>Currently printf("%lc", L'\0') is required to output no bytes. This is because 7.23.6.1 para 8 (c conversion) says "If an l length modifier is present, the wint_t argument is converted as if by an ls conversion specification with no precision and an argument that points to storage suitably sized for at least two wchar_t elements, the first element containing the wint_t argument to the lc conversion specification and the second a null wide character." This behavior is inconsistent with the related calls<br>printf("%c", '\0'), wprintf("%lc", L'\0'), and wprintf("%c", '\0') which all output a NUL. It appears that most libc implementations have chosen to make all four consistent by having printf("%lc", L'\0') output a NUL; we are only aware of one (musl) that obeys a strict reading of the C standard. The Austin Group would like to require printf("%lc", L'\0') to output a NUL in the next POSIX revision (which will necessitate that it does not defer to C17 on this) but only if the C committee indicates that it will at least allow this behavior in C23. | Option 1 (require a NUL) - change the text to:<br>If an l length modifier is present, the wint_t argument is converted as if by a call to the wcrtomb function with a pointer to storage of at least MB_CUR_MAX bytes, the wint_t argument converted to wchar_t, and an initial shift state.<br><br>Option 2 (allow either behavior) - replace the period at the end of the current text with:<br>, except that if the wint_t argument is a null wide character it is unspecified whether it is converted to no bytes or to a null byte.<br><br>The Austin Group would prefer option 1. | |
| | | 7.29.2.3 | 3 | TE | AG Reference Bug 1614 (https://austingroupbugs.net/view.php?id=1614)<br><br>========<br><br>Further discussion in the Austin Group identified two additional problems with the mktime function. | Under "Returns" in 7.29.2.3 para 3, change:<br><br>If the calendar time cannot be represented in the time_t encoding used for the return value, the function returns the value (time_t)(-1).<br><br>to: | |

1 **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

| MB/ NC[1] | Line number (e.g. 17) | Clause/ Subclause (e.g. 3.1) | Paragraph/ Figure/ Table/ (e.g. Table 1) | Type of comment[2] | Comments | Proposed change | Observations of the secretariat |
|---|---|---|---|---|---|---|---|
| | | | | | 1. When the tm components are range-corrected this can result in a tm_year value that is not representable as an int. The standard is silent about what happens in this case and therefore this is undefined behavior. All implementations we have tested either handle this in the same way as the return value not being representable, i.e. return (time_t)(-1), or don't detect the problem and return the calculated calendar time but with a bogus tm_year value. It would be good if returning (time_t)(-1) could be required, instead of the onus being on applications to avoid undefined behavior by ensuring tm_year will not overflow before they call mktime.<br><br>2. It is possible for the calendar time to be returned to have the value (time_t)(-1) and the standard does not provide any way for applications to distinguish a successful return of this value from the return-value-is-not-representable case. There is some existing practice where application code sets tm_wday to an out-of-range sentinel value and checks whether it was changed by mktime, and we are not aware of any implementation where this does not work. It would be good for this method to be required to work. | If the calendar time cannot be represented in the time_t encoding used for the return value or the value to be returned in the tm_year component of the structure pointed to by timeptr cannot be represented as an int, the function returns the value (time_t)(-1) and does not change the value of the tm_wday component of the structure.<br><br>and in the example code change:<br><br>if (mktime(&time_str) == (time_t)(-1))<br>time_str.tm_wday = 7;<br><br>to:<br><br>time_str.tm_wday = 7;<br>mktime(&time_str); | |
| | | 7.31.3.3 | 3 | TE | AG Reference Bug 1022 (https://austingroupbugs.net/view.php?id=1022)<br>=========<br>The fgetwc() function was changed in C17 to require the error indicator for the stream to be set in the case of an encoding error. However, the | Change:<br><br>"If an encoding error occurs, the value of the macro EILSEQ is stored in errno and fputwc returns WEOF." | |

1 **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

**Template for comments and secretariat observations**

| Date: | Document: | Project: |
|---|---|---|

| MB/ NC[1] | Line number (e.g. 17) | Clause/ Subclause (e.g. 3.1) | Paragraph/ Figure/ Table/ (e.g. Table 1) | Type of comment[2] | Comments | Proposed change | Observations of the secretariat |
|---|---|---|---|---|---|---|---|
| | | | | | equivalent requirement for fputwc() is not present, though POSIX requires it and does not mark it as an extension to C99. Since POSIX defers to C, and we are not aware of any implementations that do not set the error indicator, this should be aligned. | to:<br><br>"If an encoding error occurs, the error indicator for the stream is set and the value of the macro EILSEQ is stored in errno and fputwc returns WEOF." | |

1 **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by ******)
2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial