

Document: N1561

Submitter: Martin Sebor

Submission Date: 2011-03-10

Source:

Reference Document:

Version:

Date: 2011-03-10

Subject: lifetime of locale string

The C99 standard requires that the lifetime of the string referenced by the pointer returned from a successful call to `setlocale()` extend through the end of the program.

This requirement effectively prevents implementations from dynamically allocating the string (and freeing it the next time `setlocale()` is called).

The POSIX standard is considering an enhancement whereby implementations would be allowed to allocate the string dynamically and free it each time `setlocale()` is called. This is useful when the string is excessively long, typically when using "combined" locales or user defined locale databases.

<http://austingroupbugs.net/view.php?id=288>

### Suggested Technical Corrigendum

We propose that the C standard be similarly relaxed so as to permit such implementations. Specifically, change the text in 7.11.1.1 The `setlocale` function, paragraph 8 as follows:

The pointer to string returned by the `setlocale` function is such that a subsequent call with that string value and its associated category will restore that part of the program's locale. The string pointed to shall not be modified by the program. The value of the returned pointer becomes indeterminate after the subsequent call to the `setlocale` function.

Add footnote:

The behavior of a program that invokes the `setlocale` function with the value of a pointer returned by a previous invocation of the function is undefined.

Add a new bullet to Annex J, Undefined Behavior

- The `setlocale` function is invoked with the value of a pointer returned by a previous invocation of function (7.11.1).