

The New C Standard (Excerpted material)

An Economic and Cultural Commentary

Derek M. Jones

derek@knosof.co.uk

3.4.2

locale-specific behavior

locale-specific behavior

behavior that depends on local conventions of nationality, culture, and language that each implementation documents

Commentary

The concept of locale was introduced by the C Standard. It has been picked up and extended by POSIX.^[1]

Some of the execution time behavior that was implementation-defined in C90 has become locale-specific in C99. This allows programs that need to adapt to a locale to maintain a greater degree of standards conformance. Use of locale-specific behavior does not affect the conformance status of a program, while use of implementation-defined behavior means it cannot be strictly conforming (assuming that the program’s output depends on it).

Rationale

While responding to a Defect Report filed against C89, the Committee came to realize that the term, “implementation-defined,” was sometimes being used in the sense of “implementation must document” when dealing with locales. The term, “locale-specific behavior,” already in C89, was then used extensively in C95 to distinguish those properties of locales which can appear in a strictly conforming program. Because the presence or absence of a specific locale is, with two exceptions, implementation-defined, some users of the Standard were confused as to whether locales could be used at all in strictly conforming programs.

A successful call to `setlocale` has side effects, known informally as “setting the contents of the current locale,” which can alter the subsequent output of the program. A program whose output is altered only by such side effects— for example, because the decimal point character has changed— is still strictly conforming.

A program whose output is affected by the value returned by a call to `setlocale` might not be strictly conforming. If the only way in which the result affects the final output is by determining, directly or indirectly, whether to make another call to `setlocale`, then the program remains strictly conforming; but if the result affects the output in some other way, then it does not.

Common Implementations

A locale, other than "C", for which there are many common implementations is Japanese.

The POSIX locale registry is slowly beginning to accumulate information on the locales of planet Earth. Both the Unix and Microsoft Windows host environments provide some form of interface to locale databases. It is these locales that C implementations usually provide a means of accessing.

Coding Guidelines

There are two locale issues that relate to these coding guidelines:

1. *The locale in which source code is translated.* Many coding guidelines relate to how developers extract information from source code. Having the source code written in the locale of the reader is likely to make this process easier. The issues involved in using identifiers that contain characters other than those in the basic execution character set are discussed elsewhere.
2. *The locale in which translated programs are executed.* Users want programs that adapt to their locales. The priority given to ensuring that software satisfies user’s requirements is invariably much higher than that given to satisfying coding guidelines.

Writing programs that depend on locale-specific behavior obviously reduces their portability to other locales. There is also the possibility that expected behaviors will not be available if the locale is changed. However, the issues involved in deciding when to use locale-specific behavior are outside the scope of these coding guideline subsections.

universal character name syntax identifier UCN

Implementations do not always fully document their handling of locale-specific behavior. Locales are a concept that is still evolving. To gain confidence in the behavior of an implementation, test programs need to be written to verify the behavior is as documented. Dealing with partial documentation is outside the scope of these coding guidelines.

45 EXAMPLE An example of locale-specific behavior is whether the `islower` function returns true for characters other than the 26 lowercase Latin letters.

EXAMPLE
locale-specific
behavior

Example

The character e-acute is a lowercase letter in a Latin-1 locale, but not the "C" locale.

References

1. ISO. *ISO/IEC 9945-1:1990 Information technology —Portable Oper-*

ating System Interface (POSIX). ISO, 1990.