

The New C Standard (Excerpted material)

An Economic and Cultural Commentary

Derek M. Jones

derek@knosof.co.uk

3.4.1

implementation-defined behavior

unspecified behavior where each implementation documents how the choice is made

Commentary

In choosing between making the behavior of a construct implementation-defined or unspecified, the Committee had to look at the ability of translator vendors to be able to meaningfully document their implementations behavior. For instance, for assignment the order in which the operands are evaluated may depend on a range of different conditions decided on during code optimization. The only way of documenting this behavior being to supply documentation showing the data structures and algorithms used, an impractical proposition.

C90

Behavior, for a correct program construct and correct data, that depends on the characteristics of the implementation and that each implementation shall document.

The C99 wording has explicitly made the association between the terms implementation-defined and unspecified that was only implicit within the wording of the C90 Standard. It is possible to interpret the C90 definition as placing few restrictions on what an implementation-defined behavior might be. For instance, raising a signal or terminating program execution appear to be permitted. The C99 definition limits the possible behaviors to one of the possible behaviors permitted by the standard.

C++

The C++ Standard uses the same wording, 1.3.5, as C90.

Other Languages

The C Standard is up front, and in general explicitly specifies the behavior that may vary between implementations. Other languages are not always so explicit. A major design aim for Java was to make it free of any implementation-defined behaviors. The intent is that Java programs exhibit the same behavior on all hosts. It will take time to see the extent to which this design aim can be achieved in practice.

Common Implementations

Many vendors list all of the implementation-defined behavior in an appendix of the user documentation.

Coding Guidelines

Some implementation-defined behaviors have no effect on the behavior of the abstract machine. For instance, the handling of the **register** storage-class specifier can affect execution time performance, but the program semantics remain unchanged. Some implementation-defined behaviors affect the execution time characteristics of a conforming program without affecting its output; for instance, the layout of structure members having a bit-field type.

The C Standard's definition of a strictly conforming program is based on the output produced by that program, not what occurs internally in the executing program image. Many coding guideline documents contain a blanket recommendation against the use of any implementation-defined behavior. This is a simplistic approach to guideline recommendations that is overly restrictive.

Rev 42.1

Use of implementation-defined behavior shall be decided on a case-by-case basis. The extent to which the parameters of these cost/benefit decisions occur on a usage-by-usage based, a project-by-project basis, or a companywide-basis, is a management decision.

Coding guideline documents commonly recommend that a program's dependency on any implementation-defined behavior be documented. What purpose does such a guideline serve? Any well-written documentation

unspecified behavior implementation document assignment operand evaluation order

signed integer conversion implementation-defined

bit-field addressable storage unit strictly conforming program output shall not program image

would include information on implementation dependencies. The coding guidelines in this book do not aim to teach people how to write documentation. That said, the list of implementation-defined behaviors in annex J provides a good starting point.

Usage

Annex J.3 lists 97 implementation-defined behaviors.

43 EXAMPLE An example of implementation-defined behavior is the propagation of the high-order bit when a signed integer is shifted right.

EXAMPLE
high-order bit
propagation

References