The New C Standard (Excerpted material)

An Economic and Cultural Commentary

Derek M. Jones
derek@knosof.co.uk

2. Normative references

Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. 18

Commentary

A normative reference is the one that carries the same weight as wording in the standard itself. An informative reference is just that, informative.

Rationale

Just as the Standard proper excludes all examples, footnotes, references, and informative annexes, this Rationale is not part of the Standard. The C language is defined by the Standard alone. If any part of this Rationale is not in accord with that definition, the Committee would very much like to be so informed.

The ISO Directives do not permit the *Normative references* clause to contain

- documents that are not publicly available,
- documents to which only informative reference is made, or
- documents that have merely served as references in the preparation of the standard.

These kinds of documents may be listed in a bibliography. Other normative documents, not listed here, can be created through the mechanism of Defect Reports (DRs) raised against wording in the existing standards document. A DR can be raised by a National Body (a country who is a P, Participating, member of SC22), the Project Editor, or the convener of WG14.

defect report

The C committee tries to deal with DRs during the meeting immediately following their submission. The response might be to agree that there is a problem with existing wording in the standard and to provide amended wording, or to say that the issue described is not considered a problem with the standard. The committee can choose to add additional material to the standard by issuing an Amendment (such an Amendment requires a new work item, which needs the support of five P member countries before it can go ahead). There was one Amendment for C90, dealing with wide character issues.

Embedded C TR

An ISO committee can also chose to issue Technical Reports (TRs). Work has started on a TR relating to C99 (the *Embedded C* Technical Report[3]). It deals with embedded systems issues (fixed-point types, differentiating different kinds of memory, saturation arithmetic, etc.). Up-to-date information on the C Standard can be found at the official Web site, http://www.open-std.org/jtc1/sc22/wg14/.

Coding Guidelines

Referencing other documents, from within coding guidelines, could mean the reader has to spend significant time obtaining a copy of that reference. In many cases readers are unlikely to invest the effort needed to locate the referenced document.

Prior to C99 the cost of obtaining a copy of the C Standard was relatively high. The introduction of electronic distribution, and support from ANSI, has made a much lower-cost copy of the standard available. It is now realistic for guidelines to assume that their readers have access to a copy of the C Standard.

Rev 18.1

A coding guidelines document shall try to minimize references to other documents; if necessary, by including the relevant information in the guidelines document, perhaps in an annex.

Understanding the issues behind most DRs requires a close reading of the wording in the standard and usually involve situations that do not occur very often. In theory most DRs will not be of interest because use of constructs should not be relying on close readings of the standard. In practice usage of particular wording in the standard may be incidental, or the developer may not have read the wording closely at all.

The DRs raised against the C90 Standard are unlikely to have had any significant impact on programs (a few implementations had to change the way they handled constructs). It is hoped that the DRs raised against

C99 follow this pattern. Authors of coding guidelines documents might like to periodically check the official log of DRs on the C Standard Web site, http://www.open-std.org/jtc1/sc22/wg14/.

A coding guidelines document is likely to be referred to by higher-level documents. The extent to which such documents follow the cost/benefit aims of these coding guidelines, or are even known to be effective, is unknown (although one study[5] that experimentally looked at the consistency of ISO/IEC 15504 Software Process Improvement found some interesting results).

---

19 For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. Dated references

Commentary

Who knows what changes a future revision of a normative document might have. This sentence prevents revisions of some normative documents having an unintended impact on the interpretation of wording in the current version of the standard. This sentence is also explicitly stating a rule that is specified in the ISO directives.

C90

This sentence did not appear in the C90 Standard.

C++

At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. 1.2p1

The C++ Standard does not explicitly state whether later versions of standards do or do not apply. In the case of the C++ library, clause 17.3.1.4 refers to "the ISO C standard,", which could be read to imply that agreements based on the C++ Standard may reference either the C90 library or the C99 library. The C++ ISO Technical Report TR 19768 (C++ Library Extensions) includes support for the wide character library functionality that is new in C99, but does not include support for some of the type generic maths functions (some of these are the subject of work on a separate TR) or extended integer types. However, the current C++ Standard document effective references the C90 library.

---

20 However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below.

Commentary

ISO rules require that every five years the Committee responsible for a standard investigate: should the standard be reconfirmed, should the standard be withdrawn, should the standard be revised. It typically seems to take around five years to produce and revise a language standard, creating a 10-year cycle; most other kinds of standards are either confirmed or are on a shorter cycle. Different standards can also be on different parts of their cycle. It usually takes at least 12 months for some kind of formal update to a standard to be adopted.

While parties may investigate the possibility of applying the most recent editions of standards, the timescales involved in formally adopting them are such that, unless there is a very important issue, the Committee may well decide to wait for the next revision of the standard to update the references.

C90

This sentence did not appear in the C90 Standard.

Coding Guidelines

It is unlikely that a revision of one of these standards will materially affect the C Standard. It is more likely that a revision of one of these standards will affect a developer's application domain. It is this wider issue which is outside the scope of these guidelines, that is likely to be of more importance to an application.

For undated references, the latest edition of the normative document referred to applies. 21

Commentary

The referenced standards will contain a date of publication. The C Standard requires that the most up-to-date version be applied.

C90

This sentence did not appear in the C90 Standard.

Coding Guidelines

Rev 21.1 A coding guidelines document shall specify which edition of the C Standard they refer to. They shall also state if any Addendums, Defect Reports, or Technical Reports are to be taken into account.

Members of ISO and IEC maintain registers of currently valid International Standards. 22

Commentary

The ISO Web site, http://www.iso.ch, is a good starting point for information.

Coding Guidelines

Putting coding guidelines documents under the same change control system as that used for source code is a good starting point for tracking revisions. However, the first priority should always be to make sure that the guideline recommendations are followed, not inventing new procedures to handle their change control.

ISO 31-11 ISO 31-11:1992, *Quantities and units— Part 11: Mathematical signs and symbols for use in the physical sciences and technology*. 23

Commentary

This is Part 11 of a 13-part standard. Even though it is 27 pages long it is a nonexhaustive list of mathematical symbols. Part 1 of ISO 31, "Space and time", is a useful companion to ISO 8601.

ISO 8601 27

ISO 646 ISO/IEC 646, *Information technology— ISO 7-bit coded character set for information interchange*. 24

Commentary

This standard assigns meanings to values in the range 0x0 to 0x7f. There are national variants of this standard (e.g., Ascii).

ISO 8859 There are also 8-bit coded character set standards (i.e., ISO 8859–1 through ISO 8859–16) which assign meanings to the values 0x80 to 0xff (0x80–0x9f specify control codes and 0xa0 to 0xff additional graphics characters). ISO 8859–1 is commonly called *Latin-1* and covers all characters that occur in Danish, Dutch, English, Faeroese, Finnish, French, German, Icelandic, Italian, Norwegian, Portuguese, Spanish, and Swedish. This standard was recently replaced by ISO 8859–15, Latin-9, which added support for the Euro symbol and some forgotten French and Finnish letters. Values in the range 0x20 to 0x7e are always used to represent the invariant portion of ISO 646 (the so-called *International Reference Version*).

C++

1.2p1 *ISO/IEC 10646–1:1993 Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

25 ISO/IEC 2382–1:1993, *Information technology— Vocabulary — Part 1: Fundamental terms.* ISO 2382

Commentary

This is a reference to part 1 of a 27-part standard. The other 26 parts define the vocabulary for a wide range of computer-related areas.

C++

ISO/IEC 2382 (all parts), Information technology — Vocabulary 1.2p1

26 ISO 4217, *Codes for the representation of currencies and funds.*

Commentary

Quoting from its scope: "This International Standard provides the structure for a three letter alphabetic code and an equivalent three-digit numeric code for the representation of currencies and funds." Apart from an example in the library section, the contents of this standard are not used within the C Standard. However, a translator vendor may need to use this standard to implement a locale. The document is shorter than its 31 pages suggest. Half of it is written in French.

C++

There is no mention of this document in the C++ Standard.

27 ISO 8601, *Data elements and interchange formats— Information interchange— Representation of dates and times.* ISO 8601

Commentary

This standard specifies the presentation format for dates and times. It also covers issues such as whether weeks start on Sunday or Monday, and which is week-1 of a year. The definition of terms is provided by ISO 31–1, "Space and time". A translator vendor may need to use this standard to implement a locale. 23 ISO 31-11

C++

There is no mention of this document in the C++ Standard.

28 ISO/IEC 10646 (all parts), *Information technology— Universal Multiple-Octet Coded Character Set (UCS).* ISO 10646

Commentary

The ISO/IEC 10646 Standard uses a 32-bit representation, with the code positions divided into 128 groups of 256 planes with each plane containing 256 rows of 256 cells. An industrial consortium, known as Unicode http://www.unicode.org, developed a 16-bit encoding that corresponded exactly to plane zero (known as the Basic Multilingual Plane) of the 32-bit encoding used in ISO/IEC 10646. The two groups eventually merged their efforts and at the time of this writing the Unicode encoding uses the range 0x000000 to 0x10FFFF.

The supported characters do not just include letters, numbers, and symbols denoting words or parts of words, they also include symbols for non-words. For instance, *BLACK SPADE SUIT* (U+2660) ♠, *MUSIC NATURAL SIGN* (U+266E) ♮, *BLACK TELEPHONE* (U+260E) ☎, and *WHITE SMILING FACE* (U+263A) ☺.

The conditionally defined macro __STDC_ISO_10646__ may contain information on the version of ISO/IEC 10646 supported by an implementation. __STDC_ISO_106 macro

Common Implementations

Support for Unicode is more commonly seen than that for the full (or subset that is larger than Unicode) ISO/IEC 10646 specification. Both specify three encoding forms, UTF (in Unicode this is an acronym for *Unicode Transformation Format*, while in ISO/IEC 10646 it is *UCS Transformation Format*), of encoding characters:

universal character name syntax

1. As a 32-bit value directly holding the numeric value, *UTF-32* (this is the value used in UCNs).

UTF-16

2. As one, or two, 16-bit values (values in the range 0x0000–0xD7FF and 0xE000–0xFFFF goes in one, 0x010000–0x10FFFF goes in two; values in the range 0xD800–0xDFFF in the first 16 bits indicate that another value, in the range 0xDC00–0xDFFF, follows), *UTF-16*.

UTF-8

3. As a sequence of one or more 8-bit values, *UTF-8*. The following list shows the encoding used for various ranges of characters. For multibyte sequences, the number of leading 1's in the first octet equals the number of octets in the sequence:

```
U+00000000–U+0000007F: 0xxxxxxx
U+00000080–U+000007FF: 110xxxxx 10xxxxxx
U+00000800–U+0000FFFF: 1110xxxx 10xxxxxx 10xxxxxx
U+00010000–U+001FFFFF: 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
U+00200000–U+03FFFFFF: 111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
U+04000000–U+7FFFFFFF: 1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
```

C++

1.2p1 *ISO/IEC 10646–1:1993 Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*

Dated references [19]

ISO/IEC 10646:2003 is not divided into parts and the C++ Standard encourages the possibility of applying the most recent editions of standards.

IEC 60559

IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems* (previously designated IEC 559:1989) [29]

Commentary

The term *IEEE floating point* is often heard. This usage came about because the original standards on this topic were published by the IEEE. This standard for binary floating-point arithmetic is what many host processors have been providing for over a decade. However, its use is not mandated by C99. See annex F.1 for a discussion of the relationship between this standard and other related floating-point standards from which it was derived.

C90

This standard did not specify a particular floating-point format, although the values given as an example for **<float.h>** were IEEE-754 specific (which is now an International Standard, IEC 60559).

C++

There is no mention of this document in the C++ Standard.

Common Implementations

Figueroa del Cid[1] provides an extensive discussion on what the phrase *support the IEEE floating-point standard* might be interpreted to mean.

The representation for binary floating-point specified in this standard is used by the Intel x86 processor family, Sun SPARC, HP PA-RISC, IBM PowerPC, HP–was DEC– Alpha, and the majority of modern processors (some DSP processors support a subset, or make small changes, for cost/performance reasons;

while others have more substantial differences e.g., TMS320C3x[6] uses two's complement). There is also a publicly available software implementation of this standard.[2]

Other representations are still supported by processors (IBM 390 and HP–was DEC– VAX) having an existing customer base that predates the publication the documents on which this standard is based. These representations will probably continue to be supported for some time because of the existing code that relies on it (the IBM 390 and HP–was DEC– Alpha support both their companies respective older representations and the IEC 60559 requirements).

Coding Guidelines

There is a common belief that once the IEC 60559 Standard has been specified all of its required functionality will be provided by conforming implementations. It is possible that a C program's dependencies on IEC 60559 constructs, which can vary between implementations, will not be documented because of this common, incorrect belief (the person writing documentation is not always the person who is familiar with this standard).

Like the C Standard the IEC 60559 Standard does not fully specify the behavior of every construct.[4] It also provides optional behavior for some constructs, such as when underflow is raised, and has optional constructs that an implementation may or may not make use of, such as double standard. C99 does not always provide a method for finding out an implementation's behavior in these optional areas. For instance, there are no standard macros describing the various options for handling underflow.

References

1. S. A. Figueroa del Cid. *A Rigorous Framework for Fully Supporting the IEEE Standard for Floating-Point Arithmetic in High-Level Programming Languages*. PhD thesis, New York University, Jan. 2000.
2. J. Hauser. Softfloat-2b. www.jhauser.us/arithmetic/SoftFloat.html, 2002.
3. ISO. *Programming languages, their environments and system software interfaces —Extensions for the programming language C to support embedded processors*. ISO, 2004.
4. W. Kahan. IEEE standard 754 for binary floating-point arithmetic. Lecture Notes in progress, May 1995.
5. J.-M. Simon, K. E. Emam, S. Rousseau, E. Jacquet, and F. Babey. The reliability of ISO/IEC PDTR 15504 assessments. Technical Report Technical Report ISERN-97-28, Fraunhofer Institute for Experimental Software Engineering, 1997.
6. Texas Instruments. *TMS320C3x User's Guide*. Texas Instruments, Inc, spru031e, revision l edition, July 1997.