TECHNICAL SPECIFICATION

ISO/IEC TS 18661-2

First edition
2015-02-15

Information Technology — Programming languages, their environments, and system software interfaces — Floating-point extensions for C —

Part 2: Decimal floating-point arithmetic

Technologies de l'information — Langages de programmation, leurs environnements et interfaces du logiciel système — Extensions à virgule flottante pour C —

Partie 2: Arithmétique décimal en virgule flottante

Reference number
ISO/IEC TS 18661-2:2015(E)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee, SC 22, *Programming languages, their environments, and system software interfaces*.

ISO/IEC/TS 18661 consists of the following parts, under the general title *Information technology— Programming languages, their environments, and system software interfaces — Floating-point extensions for C*:

— *Part 1: Binary floating-point arithmetic*

— *Part 2: Decimal floating-point arithmetic*

The following parts are under preparation:

— *Part 3: Interchange and extended types*

— *Part 4: Supplementary functions*

— *Part 5: Supplementary attributes*

ISO/IEC/TS 18661-1 updates ISO/IEC 9899:2011, *Information technology — Programming Language C*, Annex F in particular to support all required features of ISO/IEC/IEEE 60559:2011, *Information technology — Microprocessor Systems — Floating-point arithmetic*.

ISO/IEC/TS 18661-2 supersedes ISO/IEC/TR 24732:2009, *Information technology — Programming languages, their environments and system software interfaces — Extension for the programming language C to support decimal floating-point arithmetic*.

ISO/IEC/TS 18661-3, ISO/IEC TS 18661-4, and ISO/IEC TS 18661-5 specify extensions to ISO/IEC 9899:2011 for features recommended in ISO/IEC/IEEE 60559:2011.

Introduction

Background

IEC 60559 floating-point standard

The IEEE 754:1985 standard for binary floating-point arithmetic was motivated by an expanding diversity in floating-point data representation and arithmetic which made writing robust programs, debugging, and moving programs between systems exceedingly difficult. Now, the great majority of systems provide data formats and arithmetic operations according to this International Standard. The IEC 60559:1989 international standard is equivalent to IEEE 754-1985 standard. Its stated goals are the following:

a) facilitate movement of existing programs from diverse computers to those that adhere to this International Standard;

b) enhance the capabilities and safety available to programmers who, though not expert in numerical methods, can well be attempting to produce numerically sophisticated programs. However, we recognize that utility and safety are sometimes antagonists;

c) encourage experts to develop and distribute robust and efficient numerical programs that are portable, by way of minor editing and recompilation, onto any computer that conforms to this International Standard and possesses adequate capacity. When restricted to a declared subset of the standard, these programs should produce identical results on all conforming systems;

d) provide direct support for

   1) execution-time diagnosis of anomalies,

   2) smoother handling of exceptions, and

   3) interval arithmetic at a reasonable cost;

e) provide for the development of

   1) standard elementary functions such as exp and cos,

   2) very high precision (multiword) arithmetic, and

   3) coupling of numerical and symbolic algebraic computation;

f) enable rather than preclude further refinements and extensions.

To these ends, the standard specified a floating-point model comprised of the following:

— *formats* – for binary floating-point data, including representations for Not-a-Number (NaN) and signed infinities and zeros;

— *operations* – basic arithmetic operations (addition, multiplication, etc.) on the format data to compose a well-defined, closed arithmetic system; also specified conversions between floating-point formats and decimal character sequences, and a few auxiliary operations;

— *context* – status flags for detecting exceptional conditions (invalid operation, division by zero, overflow, underflow, and inexact) and controls for choosing different rounding methods.

The ISO/IEC/IEEE 60559:2011 international standard is equivalent to the IEEE 754-2008 standard for floating-point arithmetic which is a major revision to IEEE 754-1985.

The revised standard specifies more formats including decimal as well as binary. It adds a 128-bit binary format to its basic formats. It also defines extended formats for all of its basic formats. It then specifies data interchange formats (which may or may not be arithmetic), including a 16-bit binary format and an unbounded tower of wider formats. To conform to the floating-point standard, an implementation must provide at least one of the basic formats, along with the required operations.

The revised standard specifies more operations. New requirements include, among others, arithmetic operations that round their result to a narrower format than the operands (with just one rounding), more conversions with integer types, more classifications and comparisons, and more operations for managing flags and modes. New recommendations include an extensive set of mathematical functions and seven reduction functions for sums and scaled products.

The revised standard places more emphasis on the reproducible results which is reflected in its standardization of more operations. For most parts, behaviors are completely specified. The standard requires conversions between floating-point formats and decimal character sequences to be correctly rounded for at least three more decimal digits than what is required to distinguish all numbers in the widest supported binary format. It also fully specifies conversions involving any number of decimal digits. It then recommends that transcendental functions be correctly rounded.

The revised standard requires a way to specify a constant rounding direction for a static portion of code with details left to programming language standards. This feature potentially allows rounding control without incurring the overhead of runtime access to a global (or thread) rounding mode.

Other features recommended by the revised standard include alternate methods for exception handling, controls for expression evaluation (allowing or disallowing various optimizations), support for fully reproducible results, and support for program debugging.

The revised standard, like its predecessor, defines its model of floating-point arithmetic in the abstract. It neither defines the way in which operations are expressed (which might vary depending on the computer language or other interface being used), nor does it define the concrete representation (specific layout in storage or in a processor's register, for example) of data or context, except that it does define specific encodings that are to be used for data that can be exchanged between different implementations that conform to the specification.

IEC 60559 does not include bindings of its floating-point model for particular programming languages. However, the revised standard does include guidance for programming language standards in recognition of the fact that features of the floating-point standard, even if well supported in the hardware, are not available to users unless the programming language provides a commensurate level of support. The implementation's combination of both hardware and software determines conformance to the floating-point standard.

C support for IEC 60559

The C standard specifies floating-point arithmetic using an abstract model. The representation of a floating-point number is specified in an abstract form where the constituent components (sign, exponent, significand) of the representation are defined, but not the internals of these components. In particular, the exponent range, significand size, and the base (or radix) are implementation-defined. This allows flexibility for an implementation to take advantage of its underlying hardware architecture. Furthermore, certain behaviors of operations are also implementation-defined, for example in the area of handling of special numbers and in exceptions.

The reason for this approach is historical. At the time when C was first standardized, before the floating-point standard was established, there were various hardware implementations of floating-point arithmetic in common use. Specifying the exact details of a representation would have made most of the existing implementations at the time not conforming.

Beginning with ISO/IEC 9899:1999, (C99), C has included an optional second level of specification for implementations supporting the floating-point standard. C99, in conditionally normative Annex F, introduced nearly complete support for the IEC 60559:1989 standard for binary floating-point arithmetic. Also, C99's informative Annex G offered a specification of complex arithmetic that is compatible with IEC 60559:1989.

ISO/IEC 9899:2011, (C11) includes refinements to the C99 floating-point specification, though it is still based on IEC 60559. C11:1989 upgraded Annex G from "informative" to "conditionally normative".

ISO/IEC/TR 24732:2009 introduced partial C support for the decimal floating-point arithmetic in ISO/IEC/IEEE 60559:2011. ISO/IEC/TR 24732, for which technical content was completed while

IEEE 754-2008 was still in the later stages of development, specifies decimal types based on ISO/IEC/IEEE 60559:2011 decimal formats, though it does not include all of the operations required by ISO/IEC/IEEE 60559:2011.

Purpose

The purpose of this International Standard is to provide a C language binding for ISO/IEC/IEEE 60559:2011 based on the C11 standard that delivers the goals of ISO/IEC/IEEE 60559 to users and is feasible to be implemented. It is then organized into five parts.

ISO/IEC/TS 18661-1 provides changes to C11 that cover all the requirements plus some basic recommendations of ISO/IEC/IEEE 60559:2011 for binary floating-point arithmetic. C implementations intending to support ISO/IEC/IEEE 60559:2011 are expected to conform to conditionally normative Annex F as enhanced by the changes in ISO/IEC TS 18661-1.

ISO/IEC/TS 18661-2 enhances ISO/IEC/TR 24732 to cover all the requirements plus some basic recommendations of ISO/IEC/IEEE 60559:2011 for decimal floating-point arithmetic. C implementations intending to provide an extension for decimal floating-point arithmetic supporting ISO/IEC/IEEE 60559:2011 are expected to conform to ISO/IEC TS 18661-2.

ISO/IEC/TS 18661-3 (Interchange and extended types), ISO/IEC/TS 18661-4 (Supplementary functions), and ISO/IEC/TS 18661-5 (Supplementary attributes) cover recommended features of ISO/IEC/IEEE 60559:2011. C implementations intending to provide extensions for these features are expected to conform to the corresponding parts.

Additional background on decimal floating-point arithmetic

Most of today's general-purpose computing architectures provide binary floating-point arithmetic in hardware. Binary floating point is an efficient representation that minimizes memory use and is simpler to implement than floating-point arithmetic using other bases. It has therefore become the norm for scientific computations with almost all implementations following the IEEE 754 standard for binary floating-point arithmetic (and the equivalent international ISO/IEC/IEEE 60559 standard).

However, human computation and communication of numeric values almost always uses decimal arithmetic and decimal notations. Laboratory notes, scientific papers, legal documents, business reports, and financial statements all record numeric values in decimal form. When numeric data are given to a program or are displayed to a user, conversion between binary and decimal is required. There are inherent rounding errors involved in such conversions. Decimal fractions cannot, in general, be represented exactly by binary floating-point values. These errors often cause usability and efficiency problems depending on the application.

These problems are minor when the application domain accepts or requires results to have associated error estimates (as is the case with scientific applications). However, in business and financial applications, computations are either required to be exact (with no rounding errors), unless explicitly rounded or supported by detailed analyses that are auditable to be correct. Such applications therefore have to take special care in handling any rounding errors introduced by the computations.

The most efficient way to avoid conversion error is to use decimal arithmetic. Currently, the IBM z/Architecture (and its predecessors since System/360) is a widely used system that supports built-in decimal arithmetic. Prior to the IBM System z10 processor, however, this provided integer arithmetic only, meaning that every number and computation has to have separate scale information preserved and computed in order to maintain the required precision and value range. Such scaling is difficult to code and is error-prone. It also affects execution time significantly and the resulting program is often difficult to maintain and enhance.

Eventhough the hardware might not provide decimal arithmetic operations, the support can still be emulated by software. Programming languages used for business applications either have native decimal types (such as PL/I, COBOL, REXX, C#, or Visual Basic) or provide decimal arithmetic libraries (such as the BigDecimal class in Java). The arithmetic used in business applications, nowadays, is almost invariably decimal floating-point. The COBOL 2002 ISO standard, for example, requires that all standard decimal arithmetic calculations use 32-digit decimal floating-point.

The IEEE has recognized this importance. Decimal floating-point formats and arithmetic are major new features in the IEEE 754-2008 standard and its international equivalent ISO/IEC/IEEE 60559:2011.

Information Technology — Programming languages, their environments, and system software interfaces — Floating-point extensions for C —

Part 2: Decimal floating-point arithmetic

1 Scope

This part of ISO/IEC/TS 18661 extends programming language C as specified in ISO/IEC 9899:2011, (C11) with changes specified in ISO/IEC/TS 18661-1, to support decimal floating-point arithmetic conforming to ISO/IEC/IEEE 60559:2011. It covers all requirements of IEC 60559 as they pertain to C decimal floating types.

This part of ISO/IEC/TS 18661 supersedes ISO/IEC/TR 24732:2009.

This part of ISO/IEC/TS 18661 does not cover binary floating-point arithmetic (which is covered in ISO/IEC/TS 18661-1), nor does it cover most optional features of IEC 60559.

2 Conformance

An implementation conforms to this part of ISO/IEC/TS 18661 if

a) it meets the requirements for a conforming implementation of C11 with all the changes to C11 specified in ISO/IEC/TS 18661-1 and in this part of ISO/IEC/TS 18661, and

b) it defines `__STDC_IEC_60559_DFP__` to `201ymmL`.

NOTE Conformance to this part of ISO/IEC/TS 18661 does not include all the requirements of ISO/IEC/TS 18661-1. An implementation can conform to either or both of ISO/IEC/TS 18661-1 and this part of ISO/IEC/TS 18661.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9899:2011, *Information technology — Programming languages — C*

ISO/IEC/IEEE 60559:2011, *Information technology — Microprocessor Systems — Floating-Point arithmetic*

ISO/IEC/TS 18661-1, *Information technology — Programming languages, their environments, and system software interfaces — Floating-point extensions for C — Part 1: Binary floating-point arithmetic*