INTERNATIONAL STANDARD

ISO/IEC 9899

First edition
1990-12-15

AMENDMENT 1
1995-04-01

Programming languages — C

AMENDMENT 1: C Integrity

Langages de programmation — C

AMENDEMENT 1: Intégrité de C



Reference number
ISO/IEC 9899:1990/Amd.1:1995(E)

Contents

1 Scope 1
4 Compliance 1
6 Language 1
6.1.5 Operators 1
6.1.6 Punctuators 2
6.8.8 Predefined macro names 2
7 Library 2
7.1.1 Definitions of terms 2
7.1.2 Standard headers 2
7.1.4 Errors <errno.h> 3
7.9 Input/output <stdio.h> 3
7.9.1 Introduction 3
7.9.2 Streams 3
7.9.3 Files 3
7.9.6 Formatted input/output functions 4
7.13 Future library directions 7
7.13.1 Wide-character classification and mapping utilities <wctype.h> 7
7.13.2 Extended multibyte and wide-character utilities <wchar.h> 7
7.14 Alternative spellings <iso646.h> 7
7.15 Wide-character classification and mapping utilities <wctype.h> 7
7.15.1 Introduction 7
7.15.2 Wide-character classification utilities 8
7.15.2.1 Wide-character classification functions 8
7.15.2.2 Extensible wide-character classification functions 10
7.15.3 Wide-character mapping utilities 11
7.15.3.1 Wide-character case-mapping functions 11
7.15.3.2 Extensible wide-character mapping functions 11
7.16 Extended multibyte and wide-character utilities <wchar.h> 12
7.16.1 Introduction 12
7.16.2 Formatted wide-character input/output functions 13
7.16.3 Wide-character input/output functions 21
7.16.4 General wide-string utilities 24
7.16.4.1 Wide-string numeric conversion functions 24
7.16.4.2 Wide-string copying functions 26
7.16.4.3 Wide-string concatenation functions 27
7.16.4.4 Wide-string comparison functions 27
7.16.4.5 Wide-string search functions 29
7.16.4.6 Wide-character array functions 31
7.16.5 The wcsftime function 32
7.16.6 Extended multibyte and wide-character conversion utilities 33
7.16.6.1 Single-byte wide-character conversion functions 33
7.16.6.2 The mbsinit function 34
7.16.6.3 Restartable multibyte/wide-character conversion functions 34
7.16.6.4 Restartable multibyte/wide-string conversion functions 35
Annex D: (informative) Library summary 37
Annex H: (informative) Rationale 39
Index 51

© ISO/IEC 1995
All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

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. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 1 to International Standard ISO/IEC 9899:1990 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee 22, *Programming languages, their environments and system software interfaces*.

Annexes A and B of this amendment are for information only.

Introduction

This first amendment to ISO/IEC 9899:1990 primarily consists of a set of library extensions that provide a complete and consistent set of utilities for application programming using multibyte and wide characters. It also contains extensions that provide alternate spellings for certain tokens.

The base standard deliberately chose not to include a complete multibyte and wide-character library. Instead, it defined just enough support to provide a firm foundation, both in the library and language proper, on which implementations and programming expertise could grow. Vendors did implement such extensions; this first amendment reflects the studied and careful inclusion of the best of today's existing art in this area.

The base standard also chose to provide only minimal support for writing C source code in character sets that redefine some of the punctuation characters, such as national variants of ISO 646. The alternate spellings provided here can be used to write many (but not all) tokens that are less readable when expressed in terms of trigraphs.

This first amendment to ISO/IEC 9899:1990 is divided into three major subdivisions:

— those additions and changes that affect the preliminary subdivision of ISO/IEC 9899:1990 (clauses 1 through 4);

— those additions and changes that affect the language syntax, constraints, and semantics (ISO/IEC 9899:1990 clause 6);

— those additions and changes that affect library facilities (ISO/IEC 9899:1990 clause 7).

Examples are provided to illustrate possible forms of the constructions described. Footnotes are provided to emphasize consequences of the rules described in that subclause or elsewhere in this first amendment. References are used to refer both to the base standard and to related subclauses within this document. These two can be distinguished either by context or are labeled as referring to the base standard (as above). Annex A summarizes the contents of this first amendment. Annex B provides a rationale.

Programming languages — C

AMENDMENT 1: C Integrity

1 Scope

This amendment defines extensions to ISO/IEC 9899:1990 that provide a more complete set of multibyte and wide-character utilities, as well as alternative spellings for certain tokens. Use of these features can help promote international portability of C programs.

This amendment specifies extensions that affect various clauses of ISO/IEC 9899:1990:

— To the compliance clause (clause 4), the additional header **<iso646.h>** is provided by both freestanding and hosted implementations.

— To the language clause (clause 6), six additional tokens are accepted.

— To the library clause (clause 7), new capabilities are specified for the existing formatted input/output functions (7.9.6).

— To the library clause (clause 7), the additional header **<wctype.h>** is provided, which defines a macro, several types, and many functions, including:

- wide-character testing functions, **iswalnum** for example;
- extensible wide-character classification functions, **wctype** and **iswctype**;
- wide-character case-mapping functions, **towlower** and **towupper**;
- extensible wide-character case-mapping functions, **wctrans** and **towctrans**.

— To the library clause (clause 7), the additional header **<wchar.h>** is provided, which defines several macros, several types, and many functions, including:

- formatted wide-character input/output functions, **fwprintf** for example;
- wide-character input/output functions, **fgetwc** for example;
- wide-string numeric conversion functions, **wcstod** for example;
- wide-string general utility functions, **wcscpy** for example;
- a wide-string time conversion function, **wcsftime**;
- restartable multibyte/wide-character conversion functions, **mbrtowc** for example;
- restartable multibyte/wide-string conversion functions, **mbsrtowcs** and **wcsrtombs**.

4 Compliance

The description is adjusted so that the standard header **<iso646.h>** is included in the list of headers that must be provided by both freestanding and hosted implementations.

Forward References: alternate spellings **<iso646.h>** (7.14).

6 Language

Subclauses 6.1.5 and 6.1.6 of ISO/IEC 9899:1990 are adjusted to include the following six additional tokens. In all aspects of the language, these six tokens

```
<:   :>   <%   %>   %:   %:%:
```

behave, respectively, the same as these existing six tokens

```
[    ]    {    }    #    ##
```

except for their spelling.[1)]

6.1.5 Operators

Syntax

operator: also one of

1) Thus **[** and **<:** behave differently when "stringized" (see ISO/IEC 9899:1990 subclause 6.8.3.2), but can otherwise be freely interchanged.