Technical Specification

ISO/IEC TS 6010

Programming languages — C — A provenance-aware memory object model for C

Langages de programmation — C — Modèle d'objet mémoire sensible à la provenance pour C

First edition
2025-05

Reference number
ISO/IEC TS 6010:2025(en)

© ISO/IEC 2025



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

1 Scope **1**

2 Normative references **1**

3 Terms and definitions **1**

4 Environment **2**
4.1 Execution environments . . . 2
4.2 Sizes of integer types `<limits.h>` . . . 2

5 Language **3**
5.1 Concepts . . . 3
5.1.1 Storage durations and object lifetimes . . . 3
5.1.2 Types . . . 3
5.1.3 Representation of types . . . 4
5.2 Conversion . . . 6
5.2.1 Lvalues, arrays and function designators . . . 6
5.2.2 Pointers . . . 7
5.2.3 String literals . . . 9
5.3 Expressions . . . 9
5.3.1 General . . . 9
5.3.2 Postfix operators . . . 9
5.3.3 Address and indirection operators . . . 9
5.3.4 Additive operators . . . 10
5.3.5 Relational operators . . . 10
5.3.6 Equality operators . . . 11
5.3.7 Assignment operators . . . 11
5.3.8 Declarations . . . 11
5.3.9 Structure and union specifiers . . . 11
5.3.10 Array declarations . . . 11
5.3.11 Initialization . . . 11
5.4 Statements and blocks . . . 12
5.4.1 General . . . 12
5.4.2 The `switch` statement . . . 12
5.5 External definitions . . . 12
5.5.1 General . . . 12
5.5.2 Function definitions . . . 12

6 Library **12**

© ISO/IEC 2025 – All rights reserved

6.1 Use of library functions . . . 12
6.2 Errors <errno.h> . . . 13
6.3 The longjmp function . . . 14
6.4 The signal function . . . 14
6.5 Variable arguments <stdarg.h> . . . 14
6.6 Atomics <stdatomic.h> . . . 14
6.6.1 The ATOMIC_VAR_INIT macro . . . 14
6.6.2 Atomic flag type and operations . . . 14
6.7 Integer types <stdint.h> . . . 14
6.7.1 Integer types . . . 14
6.7.2 Macros for integer constants . . . 15
6.8 Input/output <stdio.h> . . . 15
6.8.1 Streams . . . 15
6.8.2 Files . . . 15
6.8.3 File access functions . . . 16
6.8.4 Direction input/output functions . . . 16
6.9 General utilities <stdlib.h> . . . 17
6.9.1 Storage management functions . . . 17
6.9.2 Multibyte/wide character conversion functions . . . 18
6.10 String handling <string.h> . . . 18
6.10.1 Copying functions . . . 18
6.10.2 The strxfrm function . . . 18
6.11 Threads <threads.h> . . . 18
6.11.1 The tss_create function . . . 18
6.11.2 The tss_set function . . . 19
6.12 The strftime function, Date and time <time.h> . . . 19
6.13 Extended multibyte and wide character utilities <wchar.h> . . . 19
6.13.1 The fwprintf function . . . 19
6.13.2 The fwscanf function . . . 19
6.13.3 The fgetws function . . . 19
6.13.4 The wcsxfrm function . . . 20
6.13.5 The wcsftime function . . . 20

Annex A (informative) Portability issues **21**

Annex B (informative) Bounds checking interfaces **22**

Annex C (informative) Analyzability **23**

Index **24**

© ISO/IEC 2025 – All rights reserved

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.

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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

© ISO/IEC 2025 – All rights reserved

Introduction

The resolution of DR260 confirmed the concept of provenance of pointers, introduced as means to track and distinguish pointer values that represent storage instances with the same address. Implementations started to use that concept in optimisations relying on provenance-based alias analysis, without it ever being clearly or formally defined, and without it being integrated consistently with the rest of the C standard. This document provides a solution for this: a provenance-aware memory object model for C to put C programmers and implementers on a solid footing in this regard.

In addition to this document, https://cerberus.cl.cam.ac.uk/cerberus provides an executable version of the semantics, with a web interface that allows one to explore and visualise the behaviour of small test programs.

This document does not address subobject provenance.

© ISO/IEC 2025 – All rights reserved

1 Scope

This document specifies the form and establishes the interpretation of programs written in the C programming language. It is not a complete specification of that language but builds upon ISO/IEC 9899:2018 by constraining and clarifying the Memory Object Model.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 9899:2018, Programming languages – C

ISO 80000–2, Quantities and units — Part 2: Mathematical signs and symbols to be used in the natural sciences and technology.

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 9899:2018 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp/ui
- IEC Electropedia: available at https://www.electropedia.org/

3.1

pointer provenance

provenance

entity that is associated to a pointer value in the abstract machine, which is either empty, or the identity of a storage instance

3.2

storage instance

storage instance

inclusion-maximal region of data storage in the execution environment that is created when either an object definition or an allocation is encountered

Note 1 to entry: Storage instances are created and destroyed when specific language constructs (ISO/IEC 9899:2018, 6.2.4) are met during program execution, including program startup, or when specific library functions (ISO/IEC 9899:2018, 7.22.3) are called.

© ISO/IEC 2025 – All rights reserved