

1003.26™

**IEEE Standard for Information
Technology—
Portable Operating System Interface
(POSIX™)—
Part 26: Device Control Application
Program Interface (API) [C Language]**

IEEE Computer Society

Sponsored by the
Portable Applications Standards Committee



3 Park Avenue, New York, NY 10016-5997, USA

9 September 2004

Print: SH95190

PDF: SH95190



Documents

Document Sharing Hub

IEEE Standard for Information Technology— Portable Operating System Interface (POSIX™)— Part 26: Device Control Application Program Interface (API) [C Language]

Sponsor
**Portable Applications Standards Committee
of the
IEEE Computer Society**

Approved 26 April 2004
American National Standards Institute

Approved 10 December 2003
Reaffirmed 17 June 2010

IEEE-SA Standards Board

Abstract: This standard is part of the POSIX series of standards. It defines an application program interface for controlling device drivers. Although it is based on the widely used *ioctl()* system call, the interface is type-safe and has a fixed number of parameters.

Keywords: API, application portability, C (programming language), data processing, information interchange, open systems, operating system, portable application, POSIX, programming language, realtime, device drivers

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA
Copyright © 2004 by the Institute of Electrical and Electronics Engineers, Inc.

Copyright © 1998 IEEE All Rights Reserved

Abstract: This standard is part of the POSIX series of standards. It defines an application program interface for controlling device drivers. Although it is based on the widely used *ioctl()* system call, the interface is type-safe and has a fixed number of parameters.

Keywords: API, application portability, C (programming language), data processing, information interchange, open systems, operating system, portable application, POSIX, programming language, realtime, device drivers

BSD is a trademark of the University of California, Berkeley, CA, USA.

POSIX is a trademark of the Institute of Electrical and Electronics Engineers, Inc.

UNIX is a registered trademark of The Open Group.

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2004 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 09 September 2004. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

Print: ISBN 0-7381-3883-5 SH95190
PDF: ISBN 0-7381-3884-3 SS95190

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS**.”

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Contents

	PAGE
Introduction	vi
Section 1: Overview	1
1.1 Scope	1
1.2 Conformance	2
1.2.1 Implementation Conformance	2
1.2.2 Application Conformance	3
1.2.3 Other Language-Related Specifications	5
Section 2: Normative References	7
2.1 Normative References	7
Section 3: Definitions and General Concepts	9
3.1 General Concepts	9
3.1.1 Special Device	9
3.1.2 Errors	10
3.2 C Language Definitions	10
3.2.1 POSIX.26 Symbols	10
3.2.2 The Name Space	11
3.2.3 Headers	12
Section 4: Conventions and Abbreviations	13
4.1 Conventions	13
4.2 Abbreviations	14
Section 5: Device Control	15
5.1 Functions	15
5.1.1 Control a Device	15
Annex A	19
A.1 Related Open Systems Standards	19
A.2 Other Documents	20
Annex B	21
B.1 Background	21
B.2 Existing Practice	22
B.3 Relationship to <i>ioctl()</i> and the Perceived Needs for Improvement	22
B.4 Which Differences Between <i>posix_devctl()</i> and <i>ioctl()</i> Are Acceptable?	24
B.5 Rationale for the <i>dev_info_ptr</i>	24
B.6 Rationale for No <i>direction</i> Argument	25

Copyright © 2004 IEEE. All rights reserved.

B.7 Rationale for Not Defining the Direction Encoding in the <i>command</i> Word	25
B.8 Recommended Practice for Handling Data Size Errors	26
B.9 Recommended Practice for <i>nbyte</i> == 0	26
B.10 Recommended Practice for Driver-Detected Errors	27
Alphabetic Topical Index.	29

Copyright © 2004 IEEE. All rights reserved.

Introduction

(This introduction is not a normative part of IEEE Std 1003.26™-2003, IEEE Standard for Information Technology—Portable Operating System Interface (POSIX™)—Part 26: Device Control Application Program Interface (API) [C Language].)

This standard defines systems interfaces to support the source portability of applications with realtime requirements and requiring in addition the control of special devices. The system interfaces are all extensions of, or additions to, ISO/IEC 9945:2003 {2} (identical to IEEE Std 1003.1™-2003).

Within this standard, the term “POSIX.1 {2}” refers to ISO/IEC 9945:2003 {2}.

Realtime systems interact with their physical environment using a variety of devices (such as analog-digital converters, digital-analog converters, counters, and video graphic equipment), which provide a set of services that cannot be fully utilized in terms of read and/or write semantics. Traditional practice uses a single function, called *ioctl()*, to encapsulate all the control operations on the different devices connected to the system, both special or common devices. Developers of POSIX.1 {2} decided not to standardize this interface because it was not type safe, it had a variable number of parameters, and it had behaviors that could not be specified by the standard because they were driver-dependent. Instead, POSIX.1 {2} defined a device-specific application program interface (API) for a common class of drivers, Terminals; and it restricted the *ioctl()* function to control of STREAMS devices.

Although the POSIX.1 solution for common classes of devices is the best from the point of view of application portability, there is still a need for a way to interact with special, or even common devices, for which developing a full standard API is not practical. This standard defines a general method for interfacing to the widest possible range of devices, through a new service to pass control information and commands between the application and the device drivers.

A driver for a special device will normally not be portable between system implementations, but an application that uses such a driver can be made portable if all functions calling the driver are well defined and standardized. Users and integrators of realtime systems often add drivers for special devices, and a standardized function format for interfacing with these devices greatly simplifies this process.

This standard has been defined exclusively at the source code level, for the C programming language. Although the interfaces will be portable, some of the parameters used by an implementation may have hardware or configuration dependencies.

Copyright © 2004 IEEE. All rights reserved.

Organization of This Standard

This standard is divided into the following elements:

- (1) Statement of scope and overview (1)
- (2) List of normative references (2)
- (3) Definitions and global concepts (3)
- (4) Conventions and abbreviations (4)
- (5) The various interface facilities (5)
- (6) Informative annexes containing the bibliography and rationale

The C language binding for the service interface is given in the subclause labeled Synopsis (see 5.1.1.1). The Description subclause (see 5.1.1.2) provides a specification of the operation performed by the service interface. Some examples may be provided to illustrate the interfaces described.

In most cases, there are also Returns and Errors subclauses specifying return values and possible error conditions. References are used to direct the reader to other related sections. Additional material to complement sections in this standard may be found in Annex B. This annex provides historical perspectives into the technical choices made by the developers of this standard. It also provides information to emphasize consequences of the interfaces described in the corresponding section.

Informative annexes are not part of the standard and are provided for information only. They are provided for guidance and help in understanding. (There is a type of annex called *normative* that is part of a standard and imposes requirements, but there are no such normative annexes in this standard.)

Background

The developers of POSIX standards represent a cross section of hardware manufacturers, vendors of operating systems and other software development tools, software designers, consultants, academics, authors, applications programmers, and others. In the course of their deliberations, the developers reviewed related American and international standards, both published and in progress.

Conceptually, the POSIX base standards describe a set of fundamental services needed for the efficient construction of application programs. Access to these services has been provided by defining APIs, using the C programming language, that establish standard semantics and syntax. The goal of these interfaces is to enable application writers to write portable applications.

Copyright © 2004 IEEE. All rights reserved.

Although originated to refer to IEEE Std 1003.1, the name POSIX more correctly refers to a family of related standards; namely, the IEEE 1003 series and International Standard ISO/IEC 9945. This standard belongs to the POSIX family.

Audience

The intended audience for this standard is the same as for ISO/IEC 9945: all persons concerned with an industry-wide standard operating system based on the UNIX^{®1)} system. This includes at least four groups of people:

- (1) Persons buying hardware and software systems,
- (2) Persons managing companies that are deciding on future corporate computing directions,
- (3) Persons implementing operating systems, and especially
- (4) Persons developing applications where portability is an objective.

Purpose

Several principles guided the development of the POSIX standards in general and of this standard in particular.

Application-Oriented

The basic goal was to promote portability of application programs across UNIX system environments by developing a clear, consistent, and unambiguous standard for the interface specification of a portable operating system based on the UNIX system documentation.

Interface, Not Implementation

The POSIX standards define an interface, not an implementation. No distinction is made between library functions and system calls: both are referred to as *functions*. No details of the implementation of any function are given (although historical practice is sometimes indicated in Annex B). Symbolic names are given for constants (such as signals and error numbers) rather than numbers.

1) UNIX is a registered trademark of The Open Group.

Copyright © 2004 IEEE. All rights reserved.

Source, Not Object, Portability

The POSIX standards have been written so that a program written and translated for execution on one conforming implementation may also be translated for execution on another conforming implementation. In this standard, source code portability is limited to systems having the same special device drivers; but, if that is not the case, use of this standard will enhance portability by making the unportable parts that access special devices highly visible and uniformly used. This standard does not guarantee that executable (object or binary) code will execute under a conforming implementation other than that for which it was translated, even if the underlying hardware is identical.

The C Language

This standard is written in terms of the standard C language as specified in ISO/IEC 9899:1999 {1}.

No Superuser, No System Administration

There was no intention to specify all aspects of an operating system. System administration facilities and functions are excluded from the POSIX base standards, and functions usable only by the superuser have not been included. Still, an implementation of the standard interface may also implement features not in the POSIX standards. The POSIX standards are also not concerned with hardware constraints or system maintenance.

Minimal Interface, Minimally Defined

In keeping with the historical design principles of the UNIX system, the mandatory core facilities of the POSIX standards have been kept as minimal as possible. Additional capabilities were added as optional extensions or separate standards.

Broadly Implementable

The developers of the POSIX standards endeavored to make all specified functions implementable across a wide range of existing and potential systems, including the following:

- (1) All of the current major systems that are ultimately derived from the original UNIX system code (Version 7 or later).
- (2) Compatible systems that are not derived from the original UNIX system code.

Copyright © 2004 IEEE. All rights reserved.

- (3) Emulations hosted on entirely different operating systems.
- (4) Networked systems.
- (5) Distributed systems.
- (6) Systems running on a broad range of hardware.

Minimal Changes to Historical Implementations

There are no known historical implementations that will not have to change in some area to conform to this standard because, in the device control area, this standard does not exactly match any existing system interface. Nonetheless, POSIX standards specify sets of functions, types, definitions, and concepts that form an interface that is common to most historical implementations.

This standard is specifically not a codification of a particular vendor's product.

It should be noted that implementations will have different kinds of extensions. Some will reflect "historical usage" and will be preserved for execution of pre-existing applications. These functions should be considered "obsolescent," and the standard functions should be used for new applications. Some extensions will represent functions beyond the scope of this standard. These need to be used with careful management to be able to adapt to future extensions of this standard and/or port to implementations that provide these services in a different manner.

Minimal Changes to Existing Application Code

A goal of the POSIX standards was to minimize additional work for the developers of applications. However, because every known historical implementation will have to change at least slightly to conform, some applications will have to change.

Realtime Extensions

POSIX.1 {2} defines optional sets of systems interfaces to support the source portability of applications with realtime requirements. The definition of realtime used in defining the scope of this standard is as follows:

Realtime *in operating systems*: the ability of the operating system to provide a required level of service in a bounded response time.

Specifically within the scope of POSIX.1 {2} is to define interfaces that do not preclude high-performance implementations on traditional uniprocessor realtime systems.

Copyright © 2004 IEEE. All rights reserved.

This standard has the same goals as POSIX.1 {2} regarding the definition of interfaces that support the source portability of applications with real-time requirements.

Related Standards Activities

Activities to extend the POSIX family of standards to address additional requirements are in progress, and similar efforts can be anticipated in the future.

The following areas are under active consideration at this time or are expected to become active in the near future²⁾:

- (1) Additional system APIs in C language
- (2) Ada language bindings to (1)
- (3) Additional realtime facilities
- (4) Profiles describing application- or user-specific combinations of open systems standards

Some extensions may be included in “revisions” to POSIX.1 {2}, following the IEEE and ISO/IEC procedures. Other extensions, such as this standard, are approved as independent standards.

If you have interest in participating in the Portable Application Standards Committee (PASC) working groups addressing these issues, please send your name, address, and telephone number to

*Secretary, IEEE Standards Board
Institute of Electrical and Electronics Engineers, Inc.
445 Hoes Lane
Piscataway, NJ 08854
USA*

When writing, ask to have your letter forwarded to the chair of the appropriate PASC working group.

If you have interest in participating in this work at the international level, contact your ISO/IEC national body.

2) A Standards Status Report that lists all current IEEE Computer Society standards projects is available from the IEEE Computer Society, 1730 Massachusetts Avenue NW, Washington, DC 20036-1903; Telephone: +1 202 371-0101; FAX: +1 202 728-9614. Working drafts of POSIX standards under development are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854 (<http://www.standards.ieee.org/>).

Copyright © 2004 IEEE. All rights reserved.

Notice to Users

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Participants

This standard was prepared by the System Services Working Group—Realtime, sponsored by the Portable Application Standards Committee of the IEEE Computer Society. At the time this standard was approved, the membership of the System Services Working Group—Realtime was as follows:

Portable Application Standards Committee

Chair:	Lowell G. Johnson
Vice Chair:	Joseph M. Gwinn
Functional Vice Chairs:	Jay Ashford
	Andrew Josey
	Curtis Royster, Jr.
Secretary:	Nick Stoughton

Copyright © 2004 IEEE. All rights reserved.

IEEE System Services Working Group—Realtime

Chair:	Joseph M. Gwinn
Secretary:	Karen D. Gordon
Technical Editor:	Michael González
Ballot Coordinator:	Jim Oblinger
Technical Reviewer:	Franklin Prindle

Working Group

At the time this standard was completed, the IEEE System Services Working Group—Realtime had the following members:

Pierre-Jean Arcos	Chris Eck	François Riche
Ted Baker	Dave Emery	Doug Robinson
Bob Barned	Andrew Josey	Greg Shelton
Don Cragun	Jim Litchfield	Peter van der Veen
	Ray Richards	

Balloting Committee

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Pierre Jean Arcos	Charles Hammons	Vikram Punj
Mitchell Bonnett	Barry Hedquist	Francois Riche
Mark Brown	Karl Heubaum	John Riley
Paul Buerger	Daniel Hinz	Curtis Royster
Donald Cragun	Lowell G. Johnson	Diane Schleicher
John Davies	Andrew Josey	Stephen Schwarm
Juan Antonio de la Puente	Joerg Kampmann	Yuriy Sheynin
Sourav Dutta	Bil LaPlant	Gil Shultz
Yaacov Fenster	David Leciston	Keld Simonsen
Michel Gien	Roger J. Martin	Mark-Rene Uchida
Michael González	George Miao	Srinivasa Vemuru
Karen D. Gordon	James Oblinger	Rina Walach
Scott Gudgel	Peter Petrov	Oren Yuen
Joseph M. Gwinn	Franklin Prindle	Janusz Zalewski

Copyright © 2004 IEEE. All rights reserved.