Technical Specification

ISO/IEC TS 24718

First edition
2025-01

Information technology — Programming languages — Guidance for the use of the Ada Ravenscar Profile in high integrity systems

Technologies de l'information — Langages de programmation — Guide pour l'usage du profil "Ada Ravenscar" dans les systèmes de haute intégrité

Reference number
ISO/IEC TS 24718:2025(en)

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

Page

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*.

This first edition cancels and replaces the first edition (ISO/IEC TR 24718:2005).

The main changes are as follows:

— a relatively minor change to the use of a newer syntactic form for specifying aspects of entities, such as the relative priority of a task, rather than the prior use of pragmas;

— a more important change resulting from updates to the definition of the Ravenscar profile, in which support for multiple cores is now included. The primary change is to specify that all assignments of tasks to CPUs are static. In addition, some language-defined facilities are specified as not required or included in the profile for the sake of ensuring a relatively simple run-time library implementation.

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.

Introduction

0.1 General

There is an increasing recognition that the software components of critical real-time applications can be demonstrated as predictable. This is particularly the case for a hard real-time system, in which the failure of a component of the system to meet its timing deadline can result in an unacceptable degradation of the whole system. The choice of a suitable design and development method, in conjunction with supporting tools that enable the real-time performance of a system to be analysed and simulated, can lead to a high level of confidence that the final system meets its real-time constraints.

Traditional methods used for the design and development of complex applications, which concentrate primarily on functionality, are increasingly inadequate for hard real-time systems. This is because non-functional requirements such as dependability (e.g. safety and reliability), timeliness, memory usage and dynamic change management are left until too late in the development cycle.

The traditional approach to formal verification and certification of critical real-time systems has been to dispense entirely with separate processes, each with their own independent thread of control, and to use a cyclic executive that calls a series of procedures in a fully deterministic manner. Such a system becomes easy to analyse but is difficult to design for systems of more than moderate complexity, inflexible to change, and not well suited to applications where sporadic activity can occur and where error recovery is important. Moreover, it can lead to poor software engineering if small procedures must be artificially constructed to fit the cyclic schedule.

The use of the Ada programming language has proven to be of great value within high-integrity and real-time applications, albeit via language subsets of deterministic constructs, to ensure full analysability of the code. Such subsets have been defined for ISO/IEC 8652:1987 (conventionally known as “Ada 83” by language users), but these have excluded tasking on the grounds of its non-determinism and inefficiency. Subsequent advances in the area of schedulability analysis have allowed hard deadlines to be checked, even in the presence of a run-time system that enforces pre-emptive task scheduling based on multiple priorities. This valuable research work has been mapped onto a number of new Ada constructs and rules that have been incorporated into the Real-Time Annex of the Ada language Standard (ISO/IEC 8652:2023, Annex D). This evolution has opened the way for these tasking constructs to be used in high integrity subsets while retaining the core elements of predictability and reliability.

The Ravenscar profile is a subset of the tasking model as defined in ISO/IEC 8652:2023. It is restricted to meet the real-time community requirements for determinism, schedulability analysis and memory-boundedness, and is also suitable for mapping to a small and efficient run-time system that supports task synchronization and communication, and which can be certifiable to the highest integrity levels. The concurrency model promoted by the Ravenscar profile is consistent with the use of tools that allow the static properties of programs to be verified. Applicable verification techniques include information flow analysis, schedulability analysis, execution-order analysis and model checking. These techniques allow analyses of a system to be performed throughout its development life cycle, thus avoiding the common problem of discovering only during system integration and testing that the design fails to meet its non-functional requirements.

It is important to note that the Ravenscar profile is silent on the non-tasking (i.e. sequential) aspects of the language. For example, it does not dictate how exceptions should, or should not, be used. For any application in the intended domain, it is likely that constraints on the sequential part of the language will be required. These can be due to other forms of static analysis to be applied to the code, or to enable worst-case execution time information to be derived for the sequential code. See ISO/IEC TR 15942 for a detailed discussion on all aspects of static analysis of sequential Ada.

The Ravenscar profile has been designed such that the restricted form of tasking that it defines can be used even for software that should be verified to the very highest integrity levels. The Ravenscar profile has already been included in ISO/IEC TR 15942.

0.2 Structure

The document is organized as follows. The motivation for the development of the Ravenscar profile is given in Clause 4. Clause 4 also includes the definition of the profile as specified by ISO/IEC 8652:2023 (the Ada

Standard); the definition is included here for convenience, but this document is not the definitive statement of the profile. In Clause 6, the rationale for each aspect of the profile is described. Examples of usage are then provided in Clause 7. The need for verification is an important design goal for the Ravenscar profile: Clause 8 reviews the verification approach appropriate to Ravenscar programs. Finally, in Clause 9 an extended example is given.

0.3 Conventions

For all Ada-related terms, this document follows the style of the ISO/IEC 8652:2023 (the Ada standard): it uses a distinct font where there is a reference to defined syntax entities (e.g. `delay_relative_statement`).

Information technology — Programming languages — Guidance for the use of the Ada Ravenscar Profile in high integrity systems

1 Scope

This document provides guidance on the use of the Ravenscar profile for concurrent Ada software intended for verification up to, and including, the very highest levels of integrity.

To this end, this document provides a complete description of the motivations behind the Ravenscar profile, to show how conformant programs can be analysed, and to give examples of usage.

This document is aimed at a broad audience, including application programmers, implementers of run-time systems, those responsible for defining company or project guidelines, and academics. Familiarity with the Ada language is assumed.

2 Normative references

There are no normative references in this document.