INTERNATIONAL STANDARD

ISO/IEC 20970

First edition
2002-07-01

Information technology — Programming languages, their environments and system software interfaces — JEFF file format

Technologies de l'information — Langages de programmation, leurs environnements et interfaces de logiciel système — Format de fichier JEFF

Reference number
ISO/IEC 20970:2002(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.ch
Web www.iso.ch

Printed 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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of the joint technical committee is to prepare International Standards. 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.

Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 20970 was prepared by J Consortium and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

J Consortium, Inc. (“J Consortium”) has granted permission to ISO and IEC to use the trademark JEFF for products that comply with ISO/IEC 20970. Thus, implementers of ISO/IEC 20970 may use the J Consortium's JEFF trademark in connection with products that fully meet the requirements of ISO/IEC 20970.

0 Introduction

0.1 What is JEFF

This International Standard describes the JEFF File Format. This format is designed to download and store on a platform object oriented programs written in portable code. The distribution of applications is not the target of this specification.

The goal of this International Standard is to provide a ready-for-execution format allowing programs to be executed directly from static memory, thus avoiding the necessity to recopy classes into dynamic runtime memory for execution.

The constraints put on the design of JEFF are the following:

- Any set of class files must be translatable into a single JEFF file.
- JEFF must be a ready-for-execution format. A virtual machine can use it efficiently, directly from static memory (ROM, flash memory…). No copy in dynamic runtime memory or extra data modification shall be needed.
- All the standard behaviors and features of a virtual machine such as Java™ virtual machine must be reproducible using JEFF.
- In particular, JEFF must facilitate “symbolic linking” of classes. The replacement of a class definition by another class definition having a compatible signature (same class name, same fields and same method signatures) must not require any modifications in the other class definitions.

The main consequences of these choices are:

- A JEFF file can contain several classes from several packages. The content can be a complete application, parts of it, or only one class.
- To allow “symbolic linking” of classes, the references between classes must be kept at the symbolic level, even within a single JEFF file.
- The binary content of a JEFF file is adapted to be efficiently read by a wide range of processors (with different byte orders, alignments, etc.).
- JEFF is also a highly efficient format for the dynamic downloading of class definitions to dynamic memory (RAM).

0.1.1 Benefits

JEFF is a file format standard, which allows storing on-platform non pre-linked classes in a form that does not require any modification for efficient execution. JEFF exhibits a large range of benefits:

- The first of these benefits is that classes represented with JEFF can be executed directly from storage memory, without requiring any loading into runtime memory in order to be translated in a format adequate for execution. This results in a dramatic economy of runtime memory: programs with a size of several hundreds of kilobytes may then be executed with only a few kilobytes of dynamic runtime memory thanks to JEFF.
- The second benefit of JEFF is the saving of the processing time usually needed at the start of an execution to load into dynamic memory the stored classes.
- The third benefit is that JEFF does not require the classes to be pre-linked, hence fully preserving the flexibility of portable code technologies. With JEFF, programs can be updated on-platform by the mere replacement of some individual classes without requiring to replace the complete program. This provides a decisive advantage over previously proposed "ready-for-execution" formats providing only pre-linked programs.
- A last benefit of JEFF is that it allows a compact storage of programs, twice smaller than usual class file format, and this without any compression.

Information technology — Programming languages, their environments and system software interfaces — JEFF file format

1 Scope and normative references

1.1 Scope

This International Standard can be used with benefits on all kinds of platform.

This International Standard's most immediate interest is for deploying portable applications on small footprint devices. This International Standard provides dramatic savings of dynamic memory and execution time without sacrificing any of the flexibility usually attached to the use of non-pre-linked portable code.

This International Standard is especially important to provide a complete solution to execute portable programs of which code size is bigger than the available dynamic memory.

This International Standard is also very important when fast reactivity of programs is important. By avoiding the extra-processing related to loading into dynamic memory and formatting classes at runtime, this International Standard provides a complete answer to the problem of class-loading slow-down.

These benefits are particularly interesting for small devices supporting financial applications. Such applications are often complex and relying on code of significant size, while the pressure of the market often imposes to these devices to be of a low price and, consequently, to be very small footprint platforms. In addition, to not impose unacceptable delays to customers, it is important these applications do not waste time in loading classes into dynamic memory when they are launched but, on the contrary, to be immediately actively processing the transaction with no delay. When using smart cards, there are also some loose real-time constraints that are better handled if it can be granted that no temporary freezing of processing can occur due to class loading.

This International Standard can also be of great benefit for devices dealing with real-time applications. In this case, avoiding the delays due to class loading can play an important role to satisfy real-time constraints.

1.2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

[1] IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems*

[2] ISO/IEC 10646-1:2000, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane*

[3] ISO/IEC 10646-2:2001, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 2: Supplementary Planes*

[4] ISO/IEC 10646-1:2000/FDAM 1, *Mathematical symbols and other characters*

NOTE This International Standard is a self-contained specification of the JEFF format standard. However, to ease the understanding of this specification, the reading of the following document is recommended as informative reference:

The Java™ Virtual Machine Specification, Second Edition, by Tim Lindholm and Frank Yellin, 496 pages, Addison Wesley, April 1999, ISBN 0201432943.