

This is a preview of "ISO/IEC 20970:2002". [Click here to purchase the full version from the ANSI store.](#)

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)



© ISO/IEC 2002

This is a preview of "ISO/IEC 20970:2002". [Click here to purchase the full version from the ANSI store.](#)

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/IEC 2002

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 either ISO at the address below or ISO's member body in the country of the requester.

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

This is a preview of "ISO/IEC 20970:2002". Click here to purchase the full version from the ANSI store.

Contents

Page

Foreword	v
0 Introduction	vi
0.1 What is JEFF	vi
0.1.1 Benefits.....	vi
1 Scope and normative references	1
1.1 Scope	1
1.2 Normative references	1
1.3 Definitions.....	2
2 Data Types	3
2.1 Basic Types	3
2.2 Language Types.....	3
2.3 Strings	3
2.3.1 Definition	3
2.3.2 Comparison.....	3
2.3.3 Representation.....	3
2.4 Specific Types	4
2.4.1 Access Flags.....	4
2.4.2 Type Descriptor.....	5
2.4.3 Offsets.....	7
3 File Structure	8
3.1 Definitions.....	8
3.1.1 Fully Qualified Names	8
3.1.2 Internal Classes and External Classes	8
3.1.3 Fields and Methods.....	8
3.1.4 Field Position.....	9
3.2 Conventions.....	10
3.2.1 Notations	10
3.2.2 Byte Order	10
3.2.3 Alignment and Padding	10
3.3 Definition of the File Structures	11
3.3.1 File Header.....	11
3.3.2 Class Section	14
3.3.2.1 Class Header	14
3.3.2.2 Interface Table.....	16
3.3.2.3 Referenced Class Table	16
3.3.2.4 Internal Field Table	17
3.3.2.5 Internal Method Table.....	17
3.3.2.6 Referenced Field Table.....	19
3.3.2.7 Referenced Method Table	19
3.3.2.8 Bytecode Block Structure.....	20
3.3.2.9 Exception Table List.....	21
3.3.2.10 Constant Data Section	21
3.3.3 Attributes Section	23
3.3.3.1 Attribute Type.....	24
3.3.3.2 Class Attributes.....	24
3.3.3.3 Attribute Table.....	25
3.3.4 Symbolic Data Section	25
3.3.5 Constant Data Pool	27
3.3.5.1 Constant Data Pool Structure	27
3.3.5.2 Descriptor.....	27
3.3.5.3 Method Descriptor.....	28
3.3.6 Digital Signature	28

This is a preview of "ISO/IEC 20970:2002". Click here to purchase the full version from the ANSI store.

4.1	Principles	29
4.2	Translations	29
4.2.1	The tableswitch Opcode	30
4.2.2	The lookupswitch Opcode	30
4.2.3	The new Opcode	31
4.2.4	Opcodes With a Class Operand	31
4.2.5	The newarray Opcode	32
4.2.6	The multianewarray Opcode	32
4.2.7	Field Opcodes	32
4.2.8	Method Opcodes	33
4.2.9	The ldc Opcodes	34
4.2.10	The wide <opcode> Opcodes	34
4.2.11	The wide iinc Opcode	35
4.2.12	Jump Opcodes	35
4.2.13	Long Jump Opcodes	36
4.2.14	The sipush Opcode	36
4.2.15	The newconstarray Opcode	37
4.3	Unchanged Instructions	37
4.3.1	One-Byte Instructions	37
4.3.2	Two-bytes Instructions	39
4.4	Complete Opcode Mnemonics by Opcode	39
5	Restrictions	41

This is a preview of "ISO/IEC 20970:2002". [Click here to purchase the full version from the ANSI store.](#)

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.