INTERNATIONAL ISO/IEC

This is a preview of "ISO/IEC 11756:1999". Click here to purchase the full version from the ANSI store.

Second edition
1999-06-01

Information technology — Programming languages — M

Technologies de l'information — Langages de programmation — M

Reference number
ISO/IEC 11756:1999(E)

This is a preview of "ISO/IEC 11756:1999". Click here to purchase the full version from the ANSI store.

Contents

1. Scope 1

2. Normative references 1

3. Conformance 2
3.1 Implementations 2
3.2 Programs 2

4. Definitions 4

5. Metalanguage description 7

6. Routine routine 9
6.1 Routine head routinehead 9
6.2 Routine body routinebody 10
6.2.1 Level line levelline 10
6.2.2 Formal line formalline 10
6.2.3 Label label 10
6.2.4 Label separator ls 11
6.2.5 Line body linebody 11
6.3 Routine execution 11
6.3.1 Transaction processing 12
6.3.2 Error processing 13
6.4 Embedded programs 14

7. Expression expr 15
7.1 Expression atom expratom 15
7.1.1 Variables 15
7.1.2 Variable name glvn 16
7.1.2.1 Local variable name lvn 16
7.1.2.2 Local variable handling 16
7.1.2.3 Process-stack 19
7.1.2.4 Global variable name gvn 20
7.1.3 Structured system variable ssvn 21
7.1.3.1 ^$CHARACTER 21
7.1.3.2 ^$DEVICE 23
7.1.3.3 ^$GLOBAL 23
7.1.3.4 ^$JOB 24
7.1.3.5 ^$LOCK 24
7.1.3.6 ^$ROUTINE 24
7.1.3.7 ^$SYSTEM 25
7.1.3.8 ^$Z[unspecified] 25
7.1.3.9 ssvns specifying default environments 25
7.1.4 Expression item expritem 26
7.1.4.1 String literal strlit 26

© ISO/IEC 1999
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 the publisher.
ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

This is a preview of "ISO/IEC 11756:1999". Click here to purchase the full version from the ANSI store.

7.1.4.2 Numeric literal numlit 27
7.1.4.3 Numeric data values 27
7.1.4.4 Meaning of numlit 28
7.1.4.5 Numeric interpretation of data 28
7.1.4.6 Integer interpretation 29
7.1.4.7 Truth-value interpretation 29
7.1.4.8 Extrinsic function exfunc 29
7.1.4.9 Extrinsic special variable exvar 30
7.1.4.10 Intrinsic special variable names svn 30
7.1.4.11 Unary operator unaryop 34
7.1.4.12 Name value namevalue 35
7.1.5 Intrinsic function function 35
7.1.5.1 $ASCII 36
7.1.5.2 $CHAR 37
7.1.5.3 $DATA 37
7.1.5.4 $EXTRACT 37
7.1.5.5 $FIND 38
7.1.5.6 $FNUMBER 38
7.1.5.7 $GET 39
7.1.5.8 $JUSTIFY 40
7.1.5.9 $LENGTH 40
7.1.5.10 $NAME 40
7.1.5.11 $ORDER 41
7.1.5.12 $PIECE 42
7.1.5.13 $QLENGTH 43
7.1.5.14 $QSUBSCRIPT 44
7.1.5.15 $QUERY 44
7.1.5.16 $RANDOM 45
7.1.5.17 $REVERSE 45
7.1.5.18 $SELECT 45
7.1.5.19 $STACK 45
7.1.5.20 $TEXT 47
7.1.5.21 $TRANSLATE 47
7.1.5.22 $VIEW 48
7.1.5.23 $Z 48
7.2 Expression tail exprtail 48
7.2.1 Binary operator binaryop 48
7.2.1.1 Concatenation operator 49
7.2.1.2 Arithmetic binary operators 49
7.2.2 Truth operator truthop 49
7.2.2.1 Relational operator relation 49
7.2.2.2 Numeric relations 50
7.2.2.3 String relations 50
7.2.2.4 Logical operator logicalop 50
7.2.3 Pattern match pattern 51

8. Commands 52
8.1 General command rules 52
8.1.1 Spaces in commands 53
8.1.2 Comment comment 54
8.1.3 Command argument indirection 54
8.1.4 Post conditional postcond 54
8.1.5 Command timeout timeout 55
8.1.6 Line reference lineref 55
8.1.6.1 Entry reference entryref 55
8.1.6.2 Label reference labelref 56
8.1.6.3 External reference externref 57

This is a preview of "ISO/IEC 11756:1999". Click here to purchase the full version from the ANSI store.

8.1.7 Parameter passing 57
8.2 Command definitions 59
8.2.1 BREAK 59
8.2.2 CLOSE 59
8.2.3 DO 60
8.2.4 ELSE 61
8.2.5 FOR 61
8.2.6 GOTO 63
8.2.7 HALT 63
8.2.8 HANG 63
8.2.9 IF 64
8.2.10 JOB 64
8.2.11 KILL 65
8.2.12 LOCK 66
8.2.13 MERGE 67
8.2.14 NEW 68
8.2.15 OPEN 69
8.2.16 QUIT 70
8.2.17 READ 71
8.2.18 SET 73
8.2.19 TCOMMIT 75
8.2.20 TRESTART 76
8.2.21 TROLLBACK 76
8.2.22 TSTART 76
8.2.23 USE 77
8.2.24 VIEW 78
8.2.25 WRITE 78
8.2.26 XECUTE 79
8.2.27 Z 80

9. Character Set Profile charset 80

10. Character set 81

11. Expression elements 81
11.1 Names 81
11.2 External routines and names 81
11.3 Local variables 81
11.3.1 Number of local variables 81
11.3.2 Number of subscripts 81
11.3.3 Values of subscripts 82
11.4 Global variables 82
11.4.1 Number of global variables 82
11.4.2 Number of subscripts 82
11.4.3 Values of subscripts 82
11.4.4 Number of nodes 82
11.5 Data types 82
11.6 Number range 83
11.7 Integers 83
11.8 Character strings 83
11.9 Special variables 83

12. Expressions 83
12.1 Nesting of expressions 83
12.2 Results 83
12.3 External references 84

This is a preview of "ISO/IEC 11756:1999". Click here to purchase the full version from the ANSI store.

13 Routines and command lines 84
- 13.1 Command lines 84
- 13.2 Number of command lines 84
- 13.3 Number of commands 84
- 13.4 Labels 84
- 13.5 Number of labels 84
- 13.6 Number of routines 84

14 External routine calls 85

15 Character Set Profiles 85

16 Indirection 85

17 Storage space restrictions 85

18 Process-stack 86

19 Formats 86
- 19.1 mnemonicspace 86
- 19.2 controlmnemonic 86
- 19.3 Parameters 87

20 Transaction processing 87
- 20.1 Number of modifications in a TRANSACTION 87
- 20.2 Number of nested TSTARTs within a TRANSACTION 87

21 Other portability requirements 87

22 The binding 88
- 22.1 Control-functions with an effect on $X or $Y or both 88
- 22.2 Control-functions with an effect on $KEY 89
- 22.3 Control-functions with an effect on $DEVICE 89
- 22.4 Open-ended definitions 90

23 Portability issues 91
- 23.1 Implementation 91
- 23.2 Application 91

24 Conformance 92

Annex A 93

Annex B 96

Annex C 97

Annex D 99

Annex E 101

Annex F 103

Annex G 105

Annex H 106

Index 109

This is a preview of "ISO/IEC 11756:1999". 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. 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.

International Standard ISO/IEC 11756 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee 22, *Programming languages, their environments and system software interfaces*.

This second edition cancels and replaces the first edition (ISO/IEC 11756:1992), which has been technically revised.

Annex A forms an integral part of this International Standard. Annexes B to H are for information only.

This is a preview of "ISO/IEC 11756:1999". Click here to purchase the full version from the ANSI store.

Introduction

Section 1 consists of nine clauses that describe the MUMPS language. Clause 1 describes the metalanguage used in the remainder of Section 1 for the static syntax. The remaining clauses describe the static syntax and overall semantics of the language. The distinction between "static" and "dynamic" syntax is as follows. The static syntax describes the sequence of characters in a routine as it appears on a tape in routine interchange or on a listing. The dynamic syntax describes the sequence of characters that would be encountered by an interpreter during execution of the routine. (There is no requirement that MUMPS actually be interpreted). The dynamic syntax takes into account transfers of control and values produced by indirection.

Clauses 10 through 21 highlight, for the benefit of implementors and application programmers, aspects of the language that must be accorded special attention if M program transferability (i.e., portability of source code between various M implementations) is to be achieved. It provides a specification of limits that must be observed by both implementors and programmers if portability is not to be ruled out. To this end, implementors <u>must meet or exceed</u> these limits, treating them as a minimum requirement. Any implementor who provides definitions in currently undefined areas must take into account that this action risks jeopardizing the upward compatibility of the implementation, upon subsequent revision of the M Language Specification. Application programmers striving to develop portable programs must take into account the danger of employing ``unilateral extensions" to the language made available by the implementor.

The following definitions apply to the use of the terms *explicit limit* and *implicit limit* within this document. An explicit limit is one which applies directly to a referenced language construct. Implicit limits on language constructs are second-order effects resulting from explicit limits on other language constructs. For example, the explicit command line length restriction places an implicit limit on the length of any construct which must be expressed entirely within a single command line.

Clauses 22 through 24 describe the binding between M and ANSI X3.64. ANSI X3.64 is a functional standard for additional control functions for data interchange with two-dimensional character-imaging input and/or output devices. It is an ANSI standard, but also an ISO standard with roughly similar characteristics exists (ISO 2022). As such, it has been implemented in many devices worldwide. It is expected that M can be easily adapted to these implementations.

The standard defined as ANSI X3.64 defines a format for device-control. No physical device is required to be able to perform all possible control-functions. In reality, as some functions rely on certain physical properties of specific devices, no device will be able to perform all functions. The standard, however, does not specify which functions a device should be able to do, but if it is able to perform a function, how the control-information for this function is to be specified.

This binding is to the functional definitions included in X3.64. The actual dialogue between the M implementation and the device is left to the implementor.

This is a preview of "ISO/IEC 11756:1999". Click here to purchase the full version from the ANSI store.

This page intentionally left blank