

Specification for

Microprocessor universal format object modules

Committees responsible for this British Standard

The preparation of this British Standard was entrusted by the Information Systems Technology Standards Policy Committee (IST/-) to Technical Committee IST/6, upon which the following bodies were represented:

- Association for Payment Clearing Services
- British Computer Society
- British Telecommunications plc
- Business Equipment and Information Technology Association
- Department of Trade and Industry (Information Technology Standards Unit)
- Department of Trade and Industry (National Physical Laboratory)
- Electricity Supply Industry in England and Wales
- Electronic Engineering Association
- HM Treasury (Central Computer and Telecommunications Agency)
- Information Technology Users' Standards Association
- Institute of Data Processing Management
- Inter-Universities Computing Committee
- Joint Network Team
- National Computing Centre Ltd.
- National Health Service
- OFTEL (Office of Telecommunications)
- Post Office
- Telecommunication Engineering and Manufacturing Association
- Telecommunications Managers' Association

This British Standard, having been prepared under the direction of the Information Systems Technology Standards Policy Committee, was published under the authority of the Board of BSI and comes into effect on 31 October 1990

© BSI 11-1999

The following BSI references relate to the work on this standard:
Committee reference IST/6
Draft for comment 89/66390 DC

ISBN 0 580 18505 2

Amendments issued since publication

Amd. No.	Date	Comments

Contents

	Page
Committees responsible	Inside front cover
National foreword	iii
1 Scope	1
2 Definitions	1
3 References	3
4 Introduction	3
4.1 Commands	3
4.2 Free-form syntax	3
5 Basic constructions	4
5.1 Notation used for syntax definitions	4
5.2 Elementary syntactic objects	4
6 Expressions	4
6.1 Types of operands	5
6.2 Functions with no operands	5
6.3 Monadic operators and functions	5
6.4 Dyadic operators and functions	5
6.5 Data manipulation operations	6
6.6 Other operations	6
7 Variables	6
7.1 G-variable (execution starting address)	6
7.2 I-variables (external definition — symbol values)	7
7.3 L-variables (low limit)	7
7.4 N-variables (name)	7
7.5 P-variables (load pointer)	7
7.6 R-variables (relocation reference)	7
7.7 S-variables (section size)	7
7.8 W-variables (working register)	7
7.9 X-variables (external symbol reference)	8
8 Module-level commands	8
8.1 MB (module begin) command	8
8.2 ME (module end) command	8
8.3 DT (date and time of creation) command	8
8.4 AD (address descriptor) command	8
9 Comments and checksum	9
9.1 Comments	9
9.2 CS (checksum) command	9
10 Sectioning	9
10.1 SB (section begin) command	9
10.2 ST (section type) command	10
10.3 SA (section alignment) command	11
11 Symbolic name and type declaration	11
11.1 NI (name of internal symbol) command	11
11.2 NX (name of external symbol) command	11
11.3 NN (name) command	12
11.4 AT (attributes) command	12
11.5 (type) command	12
12 AS (assignment) command	13
13 Loading commands	13

	Page
13.1 LD (load) command	13
13.2 IR (initialize relocation base) command	13
13.3 LR (load relocate) command	13
13.4 RE (replicate) command	14
14 Linkage edition	14
14.1 RI (retain internal symbol) command	14
14.2 WX (weak external symbol) command	14
15 Libraries	15
15.1 LI (specify default library search list) command	15
15.2 LX (library external) command	15
16 MUFOM kernel	15
17 Binary format	15
18 Bibliography	15
Appendix A (informative) Collected syntax	16
Appendix B (informative) Suggested binary encoding	18
Appendix C (informative) Command and variable usage by processes	20
Publication(s) referred to	Inside back cover

National foreword

This British Standard, prepared under the direction of the Information Systems Technology Standards Policy Committee, is identical with IEC 975:1989 “*Microprocessor universal format for object modules*” published by the International Electrotechnical Committee (IEC).

Cross-references

The Technical Committee has reviewed the provisions of ANSI X3.4-1977¹⁾, ANSI X3.43-1977¹⁾ and ISO 3307-1975²⁾, to which reference is made in the text, and has decided that they are acceptable for use in conjunction with this standard.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i to iv, pages 1 to 22, an inside back cover and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

¹⁾ Published by the American National Standards Institute (ANSI) and available from BSI Sales Department, Linford Wood, Milton Keynes MK14 6LE.

²⁾ Published by the International Organization for Standardization (ISO) and available from BSI Sales Department, Linford Wood, Milton Keynes MK14 6LE.

1 Scope

This standard specifies the format of linkable, relocatable, and absolute object modules for binary computers of arbitrary word size and architecture. Two levels of compliance are specified, minimum and full.

The minimum compliance level affords sufficient flexibility to link separately compiled modules, to relocate addresses in simple ways, and to load the resulting absolute object modules with a minimal loader.

The full compliance level affords all of the functionality of the minimum level and adds to it arbitrary address expression handling, type checking capability, librarian control commands, and other useful functions for full generality.

A conforming implementation may extend the command or function set of MUFOM for greater efficiency in dealing with machine-specific requirements, but object modules containing such extensions shall not be said to be conforming to this standard. Such extensions are not specified in this standard.

This standard also does not specify:

- 1) The source language features to be supported by way of MUFOM modules.
- 2) The structure of relocatable code libraries.
- 3) The format of the optional binary equivalent.
- 4) Algorithms for linking, relocating and loading.
- 5) The architecture of the host machine.
- 6) The architecture of the target machine, including word size, size of memory, memory organization or instruction format.

This standard is derived from documents in the public domain and contains no patented or proprietary material.

2 Definitions

The following definitions give the meanings of the technical words as they are used in this standard and are intended to reflect the accepted use of the defined words:

absolute code

data or executable machine code in memory or an image thereof. Distinguish from relocatable code

absolute loader

a process which can load one or more sections of absolute code only at the locations specified by the sections. See: relocater

binary information

bit patterns to be loaded into memory

character form

the (printable) character representation of binary information as opposed to the bit pattern representation

checksum

a deterministic function of a file's contents. If a file is copied and the checksum of the copy is different from the original, there has been an error in copying

code

data or executable machine code. See: absolute code and relocatable code

command

control information for a linker, relocater, or loader. Distinguish from code

comments

information that is readable by people. To be distinguished from binary information and character form

external definition

the definition, within a section, of a symbol which is referenced by other sections. An exported global symbol

external reference

the specification, within a section, of a symbol which is defined outside that section. An imported global symbol

format

the language in which object modules are specified

librarian

the process which performs operations, such as maintenance, upon a library

library

a set of object modules

linkage editor (or linker)

a process which combines object modules into a single object module satisfying links between the object modules

load pointer

a pointer for a section which is dynamically maintained by the loader. It indicates where the next item of code is to be loaded. It is initialized to a starting load address

loader

same as absolute loader

MAU

abbreviation for minimum addressable unit

minimum addressable unit

for a given processor, the amount of memory located between an address and the next address. It is not necessarily equivalent to a word or a byte. Abbreviated MAU

module

same as object module

MUFOM

acronym for Microprocessor Universal Format for Object Modules

object module

a set of sections of absolute code or relocatable code, together with commands

process

a program which is executed by a processor. In this standard, the denotation is restricted to linker, relocater, expression evaluator and loader

processor

a computing machine, albeit perhaps a virtual machine

program

a set of data and instructions which defines the operations to be performed by a processor

relocatable code

position-independent code which can be loaded at an arbitrary memory location. It generally requires both relocation and expression evaluation to become absolute code

relocater

a process which assigns absolute addresses to a section of relocatable code to produce absolute code

relocation

the function of the relocater

replication

the loading of repetitive data by means of an abbreviated specification

section

a part of a program with ancillary information (commands) which becomes a segment when loaded

segment

a contiguous region in memory with arbitrary boundaries

shall

is required to. . . , are required to. . .

should

is advised to. . . , are advised to. . .

starting load address

the address at which the loading of a section begins

type

an attribute of a symbol, value or section. The definition of a symbol type is left to the implementer

weak external

an external reference which need not be satisfied during linking

3 References

- [1] ANSI X3.4:1977, *American National Standard Code for Information Interchange*.
- [2] ANSI X3.43:1977, *American National Standard Representations of Local Time of the Day for Information Interchange*.
- [3] ISO 3307:1975, *Representations of Time of the Day — Information Interchange*.

4 Introduction

4.1 Commands

A MUFOM object module is a sequence of commands. These commands are introduced by a two-letter mnemonic and are terminated by a period. Thus,

MBI8080.

ASP, 100.

LDC30001.

ME.

is a well-formed module. Detailed descriptions of the commands in this example are contained in following sections, but an informal paraphrase of this example is:

Module begin “Intel 8080.”

Assign load pointer value to be 256 (decimal).

Load three bytes with the binary information represented by the hexadecimal specification “C3,” “00,” and “01.”

Module end.

4.2 Free-form syntax

An object module is treated as one contiguous string of ASCII characters according to ANSI standard X3.4 [1]³⁾. All ASCII control characters (such as a line feed or a carriage return) are ignored in processing.

The length of a MUFOM command is variable and there is no limitation on that length. To represent a MUFOM object module as a text file on a host processor (with limited line size), a single command can occupy several successive lines (for example, encompass several line feeds, carriage returns, or other control characters).

³⁾ The numbers in square brackets correspond to the references listed in Section 3.