

Memorandum on the Standardization of The programming language COBOL

Mémorandum sur la normalisation du langage de programmation COBOL
Memorandum zur Normung der Programmiersprache COBOL

National foreword

This British Standard has been prepared under the direction of the Information Systems Technology Standards Policy Committee. It implements the European Standard EN ISO 1989 : 1989 — Programming languages. COBOL.

1. Scope

This British Standard records the international and European agreements that standardize the programming language COBOL.

2. Memorandum

The international standard, ISO 1989 : 1985, published by the International Organization for Standardization (ISO), endorsed ANSI standard X3.23 — 1985 as an international standard.

In December 1988, the European Committee for Standardization (CEN) adopted ISO 1989 : 1985 as a European Standard.

ISO 1989 : 1985 Amendment 1 : 1992 endorsed ANSI X3.23a - 1989, supplement to ANSI X3.23 - 1985, as an amendment to ISO 1989 : 1985.

In November 1993, the European Committee for Standardization (CEN), ratified the amendment as EN 21989 : 1989/A1 : 1993.

ISO 1989 : 1985 Amendment 2 : 1994 endorsed ANSI X3.23b - 1993, supplement to ANSI X3.23 - 1985, as an amendment to ISO 1989 : 1985.

In February 1995, the European Committee for Standardization (CEN), ratified the amendment as EN ISO 1989 : 1989/A2 : 1995.

This standard together with its amendments No. 1 and No. 2 constitutes the UK implementation of the European and International Standard and their respective amendments.

NOTE. With this British Standard is sold a copy of ANSI X3.23 — 1985.

The European Standard EN ISO 1989 has the status of a
British Standard

April 1995

ICS 35.060

Descriptors: Data processing, programming (computers), programming languages, COBOL

English version

Programming languages — COBOL — Correction and clarification amendment for COBOL

(ISO 1989 : 1985, Amendment 2 : 1994)

Langages de programmation — COBOL —
Amendement de correction et de clarification
pour le COBOL
(ISO 1989 : 1985, Amendement 2 : 1994)

Programmiersprachen — COBOL —
Berichtigungen und Klarstellungen zu COBOL
(ISO 1989 : 1985, Änderung 2 : 1994)

This amendment 2 modifies the European Standard EN ISO 1989 : 1989. This amendment was approved by CEN on 1995-02-18. CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this amendment the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Central Secretariat has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CEN

European Committee for Standardization
Comité Européen de Normalisation
Europäisches Komitee für Normung

Central Secretariat: rue de Stassart 36, B-1050 Brussels

© 1995 All rights of reproduction and communication in any form and by any means reserved in all countries to CEN and its members

Ref. No. EN ISO 1989 : 1989/A2 : 1995 E

Documents
Document Sharing Hub

Foreword

This European Standard has been taken over by CEN from the work of the International Technical Committee ISO JTC 1, Information technology.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by October 1995, and conflicting national standards shall be withdrawn at the latest by October 1995.

According to CEN/CENELEC Internal Regulations, the following countries are bound to implement this European Standard: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

Programming languages — COBOL

AMENDMENT 2: Correction and clarification amendment for COBOL

The following national standard is adopted as Amendment 2 to ISO International Standard 1989:1985:

ANSI X3.23b-1993, supplement to ANSI X3.23-1985.

NOTE — Copies of the ANSI standard may be obtained through the American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, N.Y. 10036, USA.



*
S
*



UDC 681.3.06 : 800.92

Descriptors: Data processing, programming (computers), programming languages, COBOL

English version

Programming languages — COBOL — Intrinsic function module

(ISO 1989 : 1985, Amendment 1 : 1992)

Langages de programmation — COBOL —
Module de fonction intrinsèque
(ISO 1989 : 1985, Amendement 1 : 1992)

Programmiersprachen — COBOL — Modul für
trigonometrische Funktionen
(ISO 1989 : 1985, Änderung 1 : 1992)

This amendment 1 modifies the European Standard EN 21989 : 1989.

This amendment was approved by CEN on 1993-11-25. CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this amendment the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CEN member. The European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Central Secretariat has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CEN

European Committee for Standardization
Comité Européen de Normalisation
Europäisches Komitee für Normung

Central Secretariat: rue de Stassart 36, B-1050 Brussels

© 1993 Copyright reserved to CEN members

Ref. No. EN 21989 : 1989/A1 : 1993 B



Documents

Document Sharing Hub

Foreword

The Technical Board decided to submit the Amendment:

'Programming languages — COBOL — Intrinsic function module (ISO 1989 : 1985 Amendment 1 : 1992)' to the Unique Acceptance Procedure (UAP).

The result of the Unique Acceptance Procedure was positive.

For the time being, this document exists only in English.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by May 1994, and conflicting national standards shall be withdrawn at the latest by May 1994.

In accordance with the CEN/CENELEC Internal Regulations, the following countries are bound to implement this European Standard: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

Endorsement notice

The text of the International Standard ISO 1989 : 1985, Amendment 1 : 1992 was approved by CEN as a European Amendment without any modification.

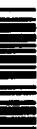
Programming languages — COBOL

AMENDMENT 1: Intrinsic function module

The following national standard is adopted as Amendment 1 to ISO International Standard 1989 : 1985 :

- ANSI X3.23a-1989, supplement to ANSI X3.23-1985.

NOTE — Copies of the ANSI standard may be obtained through the American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, N.Y. 10036, USA.

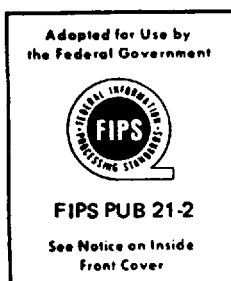


* 5 *

American National Standard

for information systems –

programming language –
COBOL



american national standards institute, inc.
1430 broadway, new york, new york 10018

This standard has been adopted as ISO International Standard 1989-1985. ISO (the International Organization for Standardization) is a worldwide federation of national standards institutes (ISO member bodies). The work of developing International Standards is carried out through ISO technical committees. Every member body interested in a subject for which a technical committee has been set up has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council.

International Standard ISO 1989 was established by Technical Committee ISO/TC 97, Information Processing Systems.

This International Standard cancels and replaces ISO 1989-1978, of which it constitutes a technical revision.

This standard has been adopted for Federal Government use.

Details concerning its use within the Federal Government are contained in Federal Information Processing Standards Publication 21-2, COBOL. For a complete list of the publications available in the Federal Information Processing Standards Series, write to the Standards Processing Coordinator (ADP), Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersburg, MD 20899.

American National Standard

for information systems – programming language – COBOL

Published by  American National Standards Institute, 1430 Broadway, New York, N.Y. 10018

March 1987

EDITORIAL CORRECTIONS

Page No.	Change
I-12	Under title "Reference Format", change "Asterisk (8) comment line" to "Asterisk (*) comment line".
I-41	Under "Character Set", change entry for "Characters used in punctuation =" from "1 STM" to "2 STM".
I-45	Under "OBJECT-COMPUTER paragraph", change entry for "SEGMENT-LIMIT clause" from "1 SEG Z" to "2 SEG Z".
I-51	In the entry following the "Level-number clause", change "may be 1 or 1 digits" to "may be 1 or 2 digits".
II-23	Paragraph 6.4.1.1, second paragraph, change "compiled program" to "compiled programs".
II-25	Paragraph 6.4.2.2, second paragraph, penultimate line, change "may be used by a called program to return to the" to "may be used by a called program to return a result to the".
IV-2	Paragraph 2.1.5, second paragraph, change "In the general format," to "In the general formats,".
V-15	Insert a terminal period following the last bracket in format 1.
V-15	Delete the commas between data-name-1 thru data-name-11, inclusively, in format 1.
V-16	Insert a terminal period following the last bracket in format 3.
V-16	Delete the commas between data-name-1 thru data-name-6, inclusively, in format 3.
V-28	Insert "[END-REWRITE]" at the end of the REWRITE statement.

(continued)

Page No.	Change
VI-20	Paragraph 5.3.2, OCCURS clause, delete the box around the first occurrence of " <u>INDEXED BY</u> { index-name-1 } ...]".
VI-31	General rule (8) of the PICTURE clause, change "explain" to "explained".
VI-50	General rule (6) of the VALUE clause, change "or in a entry" to "or in an entry".
VI-104	Paragraph 6.19.4, general rule 3, second sentence, change in part to read: "...categories: numeric, alphabetic, alphanumeric, numeric edited, alphanumeric edited."
VI-105	Paragraph 6.19.4, general rule 4c, change indentation to align with general rule 4b.
VII-6	Paragraph 2.1, in the general format, delete the box around "[<u>I-O-CONTROL</u> . [input-output-control-entry]]".
IX-6	Paragraph 1.3.5, second paragraph (2), change "If not exception" to "If no exception".
IX-33	Paragraph 4.6.3, syntax rule (3), change to read: "The INVALID KEY phrase must be specified in the REWRITE statement for indexed files, and for which an appropriate USE AFTER STANDARD EXCEPTION procedure is not specified."
X-I	Paragraph 1.1, fifth line, change "data value available" to "data values available".
XI-8	Paragraph 4.1.3, syntax rule (3), line five, change "in the file" to "in the files".
XII-4	Paragraph 2.4, general rule (9), third paragraph, second sentence, change "When a text word" to "When a text word specified in the BY phrase is introduced, it appears on a debugging line if the first library text word being replaced is specified on a debugging line. Except".
XIV-3	Delete the commas between data-name-1 thru data-name-11, inclusively, in format 1.
XIV-4	Delete the commas between data-name-1 thru data-name-6, inclusively, in format 3.
XIV-19	Paragraph 3.2.4, general rule (4), third line, change "SOURCE)" to "SOURCE))".
XVII-8	Paragraph 2.11(12), change "PROGRAM" to "PERFORM".
XVII-64	Paragraph (26), in the justification, delete the next to last paragraph that reads: "Under second Standard COBOL, PARA3 will be executed 8 times as shown above. Under third Standard COBOL, PARA3 will be executed 6 times as shown above."
XVII-70	Paragraph (37), second line, change "access made" to "access mode".

American National Standard for Information Systems – Programming Language – COBOL

Secretariat

Computer and Business Equipment Manufacturers Association

Approved September 10, 1985

American National Standards Institute, Inc

ACKNOWLEDGMENT

Any organization interested in reproducing the COBOL standard and specifications in whole or in part, using ideas from this document as the basis for an instruction manual or for any other purpose, is free to do so. However, all such organizations are requested to reproduce the following acknowledgment paragraphs in their entirety as part of the preface to any such publication (any organization using a short passage from this document, such as in a book review, is requested to mention "COBOL" in acknowledgment of the source, but need not quote the acknowledgment):

COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

No warranty, expressed or implied, is made by any contributor or by the CODASYL COBOL Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection therewith.

The authors and copyright holders of the copyrighted materials used herein

FLOW-MATIC (trademark of Sperry Rand Corporation), Programming for the UNIVAC (R) I and II, Data Automation Systems copyrighted 1958, 1959, by Sperry Rand Corporation; IBM Commercial Translator Form No. F 28-8013, copyrighted 1959 by IBM; FACT, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell

have specifically authorized the use of this material in whole or in part, in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of approval. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Published by

American National Standards Institute
1430 Broadway, New York, New York 10018

Copyright © 1985 by American National Standards Institute, Inc
All rights reserved.

No part of this publication may be reproduced in any form,
in an electronic retrieval system or otherwise, without
the prior written permission of the publisher.

Printed in the United States of America

PC10M1185/40

Foreword (This Foreword is not part of American National Standard X3.23-1985.)

This standard is a revision of American National Standard for Programming Language COBOL, ANSI X3.23-1974. The language specifications contained in this standard were drawn from both ANSI X3.23-1974 and the CODASYL COBOL Journal of Development. Like its predecessors, this document provides specifications for both the form and interpretation of programs expressed in COBOL. It is intended to provide a high degree of machine independence in such programs in order to permit their use on a variety of automatic data processing systems.

Technical Committee X3J4 on COBOL was responsible for the preparation of a revision of ANSI X3.23-1974. In performing this task, Technical Committee X3J4 held three public review and comment periods in which comments on the draft proposed revision were received from the data processing community. Technical Committee X3J4 reviewed and responded to all comments received during these public review periods. In April 1985, Technical Committee X3J4 approved the final version of the draft proposed COBOL standard.

Accredited Standards Committee on Information Processing Systems, X3, approved the draft proposed COBOL standard for submittal to ANSI as the revised American National Standard for the programming language COBOL in August 1985. The draft proposed COBOL standard was approved as an American National Standard by the American National Standards Institute on September 10, 1985.

This standard was processed and approved for submittal to ANSI by the Accredited Standards Committee on Information Processing Systems, X3. Committee approval of this standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

Edward Lohse, Chair
Catherine A. Kachurik, Administrative Secretary

<i>Organization Represented</i>	<i>Name of Representative</i>
American Library Association	Paul Peters
American Nuclear Society	Geraldine C. Main D. R. Vondy (Alt)
AMP Incorporated	Patrick E. Lannan Edward Kelly (Alt)
Association of American Railroads	R. A. Petrash
Association for Computing Machinery	Kenneth Magel Jon A. Meads (Alt)
Association of the Institute for Certification of Computer Professionals.	Thomas M. Kurihara Ardyn E. Dubnow (Alt)
AT&T Corporation	Henry L. Marchese Richard Gibson (Alt)
AT&T Information Systems.	Herbert V. Bertine Paul D. Bartoli (Alt) Stuart H. Garland (Alt)
Burroughs Corporation.	Ira R. Purchis Stanley Fenner (Alt)
Control Data Corporation	Charles E. Cooper Keith Lucke (Alt)
Cooperating Users of Burroughs Equipment.	Thomas Easterday Donald Miller (Alt)
Data General Corporation	John Pilat Lyman Chapin (Alt)
Data Processing Management Association	Christian G. Meyer
Digital Equipment Computer Users Society	William Hancock
Digital Equipment Corporation	Gary S. Robinson Delbert L. Shoemaker (Alt)
General Electric Company	William R. Kruesi

<i>Organization Represented</i>	<i>Name of Representative</i>
General Services Administration	William C. Rinehuls
GUIDE International	Larry L. Jackson (Alt)
Harris Corporation	Frank Kishenbaum
Hewlett-Packard	Thomas F. O'Leary, Jr (Alt)
Honeywell Information Systems	Sam Mathan
IBM Corporation	Rajiv Sinha (Alt)
IEEE Computer Society	Donald C. Loughry
Lawrence Berkeley Laboratory	Thomas J. McNamara
Moore Business Forms	David M. Taylor (Alt)
National Bureau of Standards	Mary Anne Gray
National Communications System	Robert H. Follett (Alt)
NCR Corporation	Sava I. Sherr
Perkin-Elmer Corporation	David Gelperin (Alt)
Prime Computer, Inc.	Thomas A. Varetoni (Alt)
Recognition Technology Users Association	David F. Stevens
SHARE, Inc.	John S. Colonias (Alt)
Sperry Corporation	Delmer H. Oddy
Texas Instruments, Inc.	Robert E. Rountree
3M Company	James H. Burrows (Alt)
Travelers Insurance Companies, Inc.	Marshall L. Cain
U.S. Department of Defense	George W. White (Alt)
VIM.	Thomas W. Kern
Wang Laboratories, Inc.	A. R. Daniels (Alt)
Xerox Corporation	Christopher Beling
	Russ Lombardo (Alt)
	Andrew F. Burlingame
	Jacqueline Barbour (Alt)
	Herbert F. Schantz
	G. W. Wetzel (Alt)
	Thomas B. Steel
	Daniel Schuster (Alt)
	Marvin W. Bass
	Jeanne G. Smith (Alt)
	Presley Smith
	Richard F. Trow, Jr (Alt)
	J. Wade Van Valkenburg
	Ray Smith (Alt)
	Joseph T. Brophy
	Fred Virtue
	Belkis Leong-Hong (Alt)
	Chris Tanner
	Madeleine Sparks (Alt)
	Marsha Hayek
	Joseph St. Amand (Alt)
	John L. Wheeler
	Arthur R. Machell (Alt)

Technical Committee X3J4 on COBOL, which developed this standard, had the following members:

D. R. Warren, Chair
L. V. Willis, Past Chair
J. T. Panttaja, Past Chair
J. Couperus, Past Chair
L. Skidmore, Vice-Chair
P. A. Beard, Secretary
M. V. Vickers,
International Representative

G. F. Archer
G. N. Baird
J. R. Brieschke
J. M. Buttler
D. Caraway
J. H. Ciminski
J. S. Cousins
D. Curry
J. W. Curtis
M. D. Dent
J. J. Edwards
B. L. Gaarder
M. P. Gerbauckas
J. Garfunkel
G. Gloss
H. Gordon
W. Haccou
P. B. Hall
R. B. Hally
K. Howard
A. Jackson

C. A. Johnson
L. A. Johnson
S. D. Klute
P. L'Allier
J. A. Machemehl
M. J. Maddison
L. K. Madison
B. Mathias
R. McKenzie
B. Miller
R. L. Miller
B. R. Nelson
J. R. Peters
A. O. Reimann
J. A. Rodriguez
M. E. Sanders
D. A. Schricker
F. D. Shea
J. A. Twentier
B. J. Verastegui
A. E. Vermilion

Others who contributed to the work on the revision were as follows:

R. M. Barton
B. Cagle
M. Candela
D. W. Christensen
M. M. Cook
D. M. Dougherty
A. L. Forsyth
A. N. Gordon
G. K. Haas
N. W. Hubacker
C. S. Hansen
D. Last
M. J. Lee
B. W. McCormick
O. Newmann

S. Ng
W. R. Osborne
P. Olshansky
B. M. Reynolds
M. J. Smith
K. Spence
G. Stephens
W. P. Storey
R. A. Surtees
R. H. Thompson
P. A. Trapp
J. M. Walker
J. F. Walton
D. Whalen
E. G. Williams

Preface

This document provides the definition of the programming language features that make up American National Standard X3.23-1985.

Within this document, the following terms are used:

- “First Standard COBOL” refers to American National Standard X3.23-1968.
- “Second Standard COBOL” refers to American National Standard X3.23-1974, which superseded American National Standard X3.23-1968.
- “Third Standard COBOL” or “Standard COBOL” refers to American National Standard X3.23-1985, which supersedes American National Standard X3.23-1974.

Contents

SECTION I: INTRODUCTORY INFORMATION

Chapter 1: Introduction to the Standard

1.1	Scope and Purpose	I-1
1.2	Structure of Language Specifications	I-1
1.3	Organization of Document	I-3
1.4	How to Use the Standard	I-3
1.5	Definition of an Implementation of Standard COBOL	I-6
1.6	Definition of a Conforming Source Program	I-9
1.7	Relationship of a Conforming Program to a Conforming Implementation	I-9

Chapter 2: Summary of Elements by Module

2.1	General Description	I-10
2.2	Summary of Elements in the Nucleus Module	I-11
2.3	Summary of Elements in the Sequential I-O Module	I-19
2.4	Summary of Elements in the Relative I-O Module	I-22
2.5	Summary of Elements in the Indexed I-O Module	I-25
2.6	Summary of Elements in the Inter-Program Communication Module ..	I-28
2.7	Summary of Elements in the Sort-Merge Module	I-30
2.8	Summary of Elements in the Source Text Manipulation Module	I-32
2.9	Summary of Elements in the Report Writer Module	I-33
2.10	Summary of Elements in the Communication Module	I-36
2.11	Summary of Elements in the Debug Module	I-38
2.12	Summary of Elements in the Segmentation Module	I-39

Chapter 3: Summary of Elements by COBOL Division

3.1	General Description	I-40
3.2	Summary of Elements in Language Concepts	I-41
3.3	Summary of Elements in Identification Division	I-44
3.4	Summary of Elements in Environment Division	I-45
3.5	Summary of Elements in Data Division	I-48
3.6	Summary of Elements in Procedure Division	I-53

SECTION II: CONCEPTS

1.	Introduction	II-1
2.	Files	II-1
3.	Report Writer	II-8
4.	Table Handling	II-12
5.	Shared Memory Area	II-17
6.	Program and Run Unit Organization and Communication	II-18
7.	Communication Facility	II-28

SECTION III: GLOSSARY

1.	Introduction	III-1
2.	Definitions	III-1

SECTION IV: OVERALL LANGUAGE CONSIDERATION

Chapter 1:	Introduction	IV-1
Chapter 2:	Notation Used in Formats	IV-1
Chapter 3:	Rules	IV-3
Chapter 4:	Language Concepts	
4.1	Character Set	IV-4
4.2	Language Structure	IV-4
4.3	Concept of Computer Independent Data Description	IV-13
4.4	Explicit and Implicit Specifications	IV-25
4.5	External Switch	IV-28
Chapter 5:	A COBOL Source Program	
5.1	Introduction	IV-29
5.2	Organization	IV-29
5.3	Structure	IV-29
Chapter 6:	Divisions	
6.1	Identification Division	IV-30
6.2	Environment Division	IV-31
6.3	Data Division	IV-33
6.4	Procedure Division	IV-35
Chapter 7:	Reference Format	IV-41
Chapter 8:	COBOL Reserved Words	IV-45

SECTION V: COMPOSITE LANGUAGE SKELETON V-1

SECTION VI: NUCLEUS MODULE

Chapter 1: Introduction to the Nucleus Module

1.1	Function	VI-1
1.2	Level Characteristics	VI-1
1.3	Level Restrictions on Overall Language	VI-1

Chapter 2: A COBOL Source Program

2.1	General Description	VI-3
2.2	Organization	VI-3
2.3	Structure	VI-3
2.4	End Program Header	VI-5

Chapter 3: Identification Division in the Nucleus Module

3.1	General Description	VI-6
3.2	Organization	VI-6
3.3	The PROGRAM-ID Paragraph	VI-7
3.4	The DATE-COMPILED Paragraph	VI-8

Chapter 4: Environment Division in the Nucleus Module

4.1	General Description	VI-9
4.2	Configuration Section	VI-9
4.3	The SOURCE-COMPUTER Paragraph	VI-10
4.4	The OBJECT-COMPUTER Paragraph	VI-11
4.5	The SPECIAL-NAMES Paragraph	VI-13

Chapter 5: Data Division in the Nucleus Module

5.1	General Description	VI-18
5.2	Working-Storage Section	VI-18
5.3	The Data Description Entry	VI-20
5.4	The BLANK WHEN ZERO Clause	VI-22
5.5	The Data-Name or FILLER Clause	VI-23
5.6	The JUSTIFIED Clause	VI-24
5.7	Level-Number	VI-25
5.8	The OCCURS Clause	VI-26
5.9	The PICTURE Clause	VI-29
5.10	The REDEFINES Clause	VI-38
5.11	The RENAMES Clause	VI-40
5.12	The SIGN Clause	VI-42
5.13	The SYNCHRONIZED Clause	VI-44
5.14	The USAGE Clause	VI-46
5.15	The VALUE Clause	VI-48

Chapter 6: Procedure Division in the Nucleus Module

6.1	General Description	VI-51
6.2	Arithmetic Expressions	VI-51
6.3	Conditional Expressions	VI-54
6.4	Common Options and Rules for Statements	VI-67
6.5	The ACCEPT Statement	VI-71
6.6	The ADD Statement	VI-73
6.7	The ALTER Statement	VI-75
6.8	The COMPUTE Statement	VI-76
6.9	The CONTINUE Statement	VI-77
6.10	The DISPLAY Statement	VI-78
6.11	The DIVIDE Statement	VI-80
6.12	The ENTER Statement	VI-83
6.13	The EVALUATE Statement	VI-84
6.14	The EXIT Statement	VI-88
6.15	The GO TO Statement	VI-89
6.16	The IF Statement	VI-90
6.17	The INITIALIZE Statement	VI-92
6.18	The INSPECT Statement	VI-94
6.19	The MOVE Statement	VI-103
6.20	The MULTIPLY Statement	VI-107
6.21	The PERFORM Statement	VI-109
6.22	The SEARCH Statement	VI-122
6.23	The SET Statement	VI-127
6.24	The STOP Statement	VI-130
6.25	The STRING Statement	VI-131
6.26	The SUBTRACT Statement	VI-134
6.27	The UNSTRING Statement	VI-136

Chapter 7: Debugging in the Nucleus Module

7.1	General Description	VI-141
7.2	Compile Time Switch	VI-141
7.3	Debugging Lines	VI-141

SECTION VII: SEQUENTIAL I-O MODULE

Chapter 1: Introduction to the Sequential I-O Module

1.1	Function	VII-1
1.2	Level Characteristics	VII-1
1.3	Language Concepts	VII-1

Chapter 2: Environment Division in the Sequential I-O Module

2.1	Input-Output Section	VII-6
2.2	The FILE-CONTROL Paragraph	VII-7
2.3	The File Control Entry	VII-7
2.4	The ACCESS MODE Clause	VII-9
2.5	The FILE STATUS Clause	VII-10
2.6	The ORGANIZATION IS SEQUENTIAL Clause	VII-11
2.7	The PADDING CHARACTER Clause	VII-12
2.8	The RECORD DELIMITER Clause	VII-13

Chapter 2: Environment Division in the Sequential I-O Module (Continued)

2.9	The RESERVE Clause	VII-14
2.10	The I-O-CONTROL Paragraph	VII-15
2.11	The MULTIPLE FILE TAPE Clause	VII-16
2.12	The RERUN Clause	VII-17
2.13	The SAME Clause	VII-19

Chapter 3: Data Division in the Sequential I-O Module

3.1	File Section	VII-21
3.2	The File Description Entry	VII-22
3.3	The BLOCK CONTAINS Clause	VII-23
3.4	The CODE-SET Clause	VII-24
3.5	The DATA RECORDS Clause	VII-25
3.6	The LABEL RECORDS Clause	VII-26
3.7	The LINAGE Clause	VII-27
3.8	The RECORD Clause	VII-30
3.9	The VALUE OF Clause	VII-33

Chapter 4: Procedure Division in the Sequential I-O Module

4.1	General Description	VII-34
4.2	The CLOSE Statement	VII-35
4.3	The OPEN Statement	VII-39
4.4	The READ Statement	VII-44
4.5	The REWRITE Statement	VII-48
4.6	The USE Statement	VII-50
4.7	The WRITE Statement	VII-52

SECTION VIII: RELATIVE I-O MODULE

Chapter 1: Introduction to the Relative I-O Module

1.1	Function	VIII-1
1.2	Level Characteristics	VIII-1
1.3	Language Concepts	VIII-1

Chapter 2: Environment Division in the Relative I-O Module

2.1	Input-Output Section	VIII-8
2.2	The FILE-CONTROL Paragraph	VIII-8
2.3	The File Control Entry	VIII-8
2.4	The ACCESS MODE Clause	VIII-10
2.5	The ORGANIZATION IS RELATIVE Clause	VIII-12
2.6	The I-O-CONTROL Paragraph	VIII-13

Chapter 3: Data Division in the Relative I-O Module

3.1	File Section	VIII-14
3.2	The File Description Entry	VIII-14

Chapter 4: Procedure Division in the Relative I-O Module

4.1	General Description	VIII-16
4.2	The CLOSE Statement	VIII-17
4.3	The DELETE Statement	VIII-19
4.4	The OPEN Statement	VIII-21
4.5	The READ Statement	VIII-26
4.6	The REWRITE Statement	VIII-30
4.7	The START Statement	VIII-33
4.8	The USE Statement	VIII-35
4.9	The WRITE Statement	VIII-37

SECTION IX: INDEXED I-O MODULE

Chapter 1: Introduction to the Indexed I-O Module

1.1	Function	IX-1
1.2	Level Characteristics	IX-1
1.3	Language Concepts	IX-1

Chapter 2: Environment Division in the Indexed I-O Module

2.1	Input-Output Section	IX-8
2.2	The FILE-CONTROL Paragraph	IX-8
2.3	The File Control Entry	IX-8
2.4	The ACCESS MODE Clause	IX-10
2.5	The ALTERNATE RECORD KEY Clause	IX-11
2.6	The ORGANIZATION IS INDEXED Clause	IX-13
2.7	The RECORD KEY Clause	IX-14
2.8	The I-O-CONTROL Paragraph	IX-15

Chapter 3: Data Division in the Indexed I-O Module

3.1	File Section	IX-16
3.2	The File Description Entry	IX-16

Chapter 4: Procedure Division in the Indexed I-O Module

4.1	General Description	IX-18
4.2	The CLOSE Statement	IX-19
4.3	The DELETE Statement	IX-21
4.4	The OPEN Statement	IX-23
4.5	The READ Statement	IX-28
4.6	The REWRITE Statement	IX-33
4.7	The START Statement	IX-36
4.8	The USE Statement	IX-39
4.9	The WRITE Statement	IX-41

SECTION X: INTER-PROGRAM COMMUNICATION MODULE

Chapter 1: Introduction to the Inter-Program Communication Module

1.1	Function	X-1
1.2	Level Characteristics	X-1
1.3	Language Concepts	X-1

Chapter 2: Nested Source Programs

2.1	General Description	X-8
2.2	Organization	X-8
2.3	Structure	X-8
2.4	Initial State of a Program	X-10
2.5	End Program Header	X-11

Chapter 3: Identification Division in the Inter-Program Communication Module

3.1	The PROGRAM-ID Paragraph and Nested Source Programs	X-12
-----	---	------

Chapter 4: Data Division in the Inter-Program Communication Module

4.1	Linkage Section	X-13
4.2	The File Description Entry in the Inter-Program Communication Module	X-15
4.3	The Data Description Entry in the Inter-Program Communication Module	X-19
4.4	The Report Description Entry in the Inter-Program Communication Module	X-22
4.5	The EXTERNAL Clause	X-23
4.6	The GLOBAL Clause	X-24

Chapter 5: Procedure Division in the Inter-Program Communication Module

5.1	The Procedure Division Header	X-25
5.2	The CALL Statement	X-27
5.3	The CANCEL Statement	X-31
5.4	The EXIT PROGRAM Statement	X-33
5.5	The USE Statement	X-34
5.6	The USE BEFORE REPORTING Statement	X-35

SECTION XI: SORT-MERGE MODULE

Chapter 1: Introduction to the Sort-Merge Module

1.1	Function	XI-1
1.2	Language Concepts	XI-1

Chapter 2: Environment Division in the Sort-Merge Module

2.1	Input-Output Section	XI-2
2.2	The FILE-CONTROL Paragraph	XI-2
2.3	The File Control Entry	XI-2
2.4	The I-O-CONTROL Paragraph	XI-3
2.5	The SAME RECORD/SORT/SORT-MERGE AREA Clause	XI-4

Chapter 3: Data Division in the Sort-Merge Module

3.1	File Section	XI-6
3.2	The Sort-Merge File Description Entry	XI-7

Chapter 4: Procedure Division in the Sort-Merge Module

4.1	The MERGE Statement	XI-8
4.2	The RELEASE Statement	XI-13
4.3	The RETURN Statement	XI-14
4.4	The SORT Statement	XI-16

SECTION XII: SOURCE TEXT MANIPULATION MODULE

Chapter 1: Introduction to the Source Text Manipulation Module

1.1	Function	XII-1
1.2	Level Characteristics	XII-1

Chapter 2: The COPY Statement

XII-2

Chapter 3: The REPLACE Statement

XII-6

SECTION XIII: REPORT WRITER MODULE

Chapter 1: Introduction to the Report Writer Module

1.1	Function	XIII-1
1.2	Language Concepts	XIII-1

Chapter 2: Environment Division in the Report Writer Module

2.1	Input-Output Section	XIII-3
2.2	The FILE-CONTROL Paragraph	XIII-3
2.3	The File Control Entry	XIII-3
2.4	The I-O-CONTROL Paragraph	XIII-5

Chapter 3: Data Division in the Report Writer Module

3.1	File Section.....	XIII-6
3.2	The File Description Entry.....	XIII-7
3.3	The REPORT Clause.....	XIII-9
3.4	Report Section.....	XIII-10
3.5	The Report Description Entry.....	XIII-11
3.6	The CODE Clause.....	XIII-14

Chapter 3: Data Division in the Report Writer Module (Continued)

3.7	The CONTROL Clause	XIII-15
3.8	The PAGE Clause	XIII-17
3.9	The Report Group Description Entry	XIII-20
3.10	Presentation Rules Tables	XIII-24
3.11	The COLUMN NUMBER Clause	XIII-42
3.12	The Data-Name Clause	XIII-43
3.13	The GROUP INDICATE Clause	XIII-44
3.14	Level-Number	XIII-45
3.15	The LINE NUMBER Clause	XIII-46
3.16	The NEXT GROUP Clause	XIII-48
3.17	The SIGN Clause	XIII-49
3.18	The SOURCE Clause	XIII-51
3.19	The SUM Clause	XIII-52
3.20	The TYPE Clause	XIII-55
3.21	The USAGE Clause	XIII-60
3.22	The VALUE Clause	XIII-61

Chapter 4: Procedure Division in the Report Writer Module

4.1	General Description	XIII-62
4.2	The CLOSE Statement	XIII-63
4.3	The GENERATE Statement	XIII-66
4.4	The INITIATE Statement	XIII-69
4.5	The OPEN Statement	XIII-70
4.6	The SUPPRESS Statement	XIII-74
4.7	The TERMINATE Statement	XIII-75
4.8	The USE AFTER EXCEPTION/ERROR PROCEDURE Statement	XIII-76
4.9	The USE BEFORE REPORTING Statement	XIII-78

SECTION XIV: COMMUNICATION MODULE

Chapter 1: Introduction to the Communication Module

1.1	Function	XIV-1
1.2	Level Characteristics	XIV-1

Chapter 2: Data Division in the Communication Module

2.1	Communication Section	XIV-2
2.2	The Communication Description Entry	XIV-3

Chapter 3: Procedure Division in the Communication Module

3.1	The ACCEPT MESSAGE COUNT Statement	XIV-17
3.2	The DISABLE Statement	XIV-18
3.3	The ENABLE Statement	XIV-20
3.4	The PURGE Statement	XIV-22
3.5	The RECEIVE Statement	XIV-23
3.6	The SEND Statement	XIV-26

SECTION XV: DEBUG MODULE

Chapter 1: Introduction to the Debug Module

1.1	Function	XV-1
1.2	Level Characteristics	XV-1
1.3	Language Concepts	XV-1

Chapter 2: Environment Division in the Debug Module

2.1	The WITH DEBUGGING MODE Clause	XV-3
-----	--------------------------------------	------

Chapter 3: Procedure Division in the Debug Module

3.1	General Description	XV-4
3.2	The USE FOR DEBUGGING Statement	XV-5

SECTION XVI: SEGMENTATION MODULE

Chapter 1: Introduction to the Segmentation Module

1.1	Function	XVI-1
1.2	Level Characteristics	XVI-1
1.3	Scope	XVI-1
1.4	Organization	XVI-1
1.5	Segmentation Classification	XVI-2
1.6	Segmentation Control	XVI-3

Chapter 2: Environment Division in the Segmentation Module

2.1	Configuration Section	XVI-4
2.2	The OBJECT-COMPUTER Paragraph	XVI-4
2.3	The SEGMENT-LIMIT Clause	XVI-5

Chapter 3: Procedure Division in the Segmentation Module

3.1	General Description	XVI-6
3.2	Segment-Numbers	XVI-7
3.3	Restrictions on Program Flow	XVI-8

SECTION XVII: APPENDICES

APPENDIX A: THE HISTORY OF COBOL

Chapter 1: The Development of COBOL

1.1	Organization of COBOL Effort	XVII-1
1.2	The COBOL Maintenance Committee	XVII-1
1.3	The COBOL Committee	XVII-2
1.4	The Programming Language Committee	XVII-2
1.5	The CODASYL COBOL Committee	XVII-2

Chapter 2: The Evolution of CODASYL COBOL

2.1	COBOL-60	XVII-3
2.2	COBOL-61	XVII-3
2.3	COBOL-61 Extended	XVII-3
2.4	COBOL, Edition 1965	XVII-3
2.5	CODASYL COBOL Journal of Development 1968	XVII-4
2.6	CODASYL COBOL Journal of Development 1969	XVII-4
2.7	CODASYL COBOL Journal of Development 1970	XVII-5
2.8	CODASYL COBOL Journal of Development 1973	XVII-5
2.9	CODASYL COBOL Journal of Development 1976	XVII-6
2.10	CODASYL COBOL Journal of Development 1978	XVII-7
2.11	CODASYL COBOL Journal of Development 1981	XVII-8
2.12	CODASYL COBOL Journal of Development 1984	XVII-9

Chapter 3: The Standardization of COBOL

3.1	Initial Standardization Effort	XVII-11
3.2	USA Standard COBOL 1968	XVII-11
3.3	American National Standard COBOL 1974	XVII-12
3.4	American National Standard COBOL 1985	XVII-12

Chapter 4: International Standardization of COBOL

4.1	ISO Recommendation R-1989-1972 for COBOL	XVII-14
4.2	ISO Standard 1989-1978 for COBOL	XVII-14
4.3	ISO Standard 1989-1985 for COBOL	XVII-15

APPENDIX B: DIFFERENCES BETWEEN SECOND AND THIRD STANDARD COBOL

Chapter 1: Summary of Differences Between Second and Third Standard COBOL

XVII-16

1.1	Summary of Differences in Language Concepts	XVII-17
1.2	Summary of Differences in Identification Division	XVII-20
1.3	Summary of Differences in Environment Division	XVII-21
1.4	Summary of Differences in Data Division	XVII-25
1.5	Summary of Differences in Procedure Division	XVII-30
1.6	Additional Summary of Differences	XVII-41

Chapter 2: Substantive Changes

2.1	Substantive Changes Not Affecting Existing Programs	XVII-42
2.2	Substantive Changes Potentially Affecting Existing Programs	XVII-51

APPENDIX C: LANGUAGE ELEMENT LISTS

Chapter 1:	Obsolete Language Element List	XVII-81
Chapter 2:	Implementor-Defined Language Element List	XVII-87
Chapter 3:	Hardware Dependent Language Element List	XVII-94
Chapter 4:	Undefined Language Element List	XVII-96
<u>INDEX</u>		XVIII-1

American National Standard for Information Systems –

Programming Language – COBOL

SECTION I: INTRODUCTORY INFORMATION

1. INTRODUCTION TO THE STANDARD

1.1 SCOPE AND PURPOSE

The scope of this standard is to specify both the form and interpretation of programs expressed in COBOL. Its purpose is to promote a high degree of machine independence in such programs in order to permit their use on a variety of automatic data processing systems.

1.2 STRUCTURE OF LANGUAGE SPECIFICATIONS

The organization of COBOL specifications in this standard is based on a functional processing module concept. The standard defines 11 functional processing modules: Nucleus, Sequential I-O, Relative I-O, Indexed I-O, Inter-Program Communication, Sort-Merge, Source Text Manipulation, Report Writer, Communication, Debug, and Segmentation. Nine of the modules have the elements within the module divided into level 1 elements and level 2 elements. Level 1 elements of a module are a subset of level 2 elements of the same module. Two of the modules contain only level 1 elements.

The Nucleus module contains the language elements for internal processing of data within the basic structure of the four divisions of a program. The Nucleus also contains the language elements for the definition and access of tables. The elements of the Nucleus are divided into two levels. Nucleus level 1 supplies elements that perform basic internal operations, i.e., the more elementary options of the various clauses and verbs. Nucleus level 2 provides elements for more extensive and sophisticated internal processing capabilities.

The Sequential I-O module contains the language elements for the definition and access of sequentially organized files. The elements of the Sequential I-O module are divided into two levels. Sequential I-O level 1 provides elements for the basic facilities of definition and access of sequential files. Sequential I-O level 2 provides elements for the complete facilities of definition and access of sequential files.

Introduction

The Relative I-O module contains the language elements for the definition and access of mass storage files in which records are identified by relative record numbers. The elements of the Relative I-O module are divided into two levels. Relative I-O level 1 provides elements for the basic facilities of definition and access of relative files. Relative I-O level 2 provides elements for more complete facilities, including the capability of accessing the file both randomly and sequentially in the same COBOL program.

The Indexed I-O module contains the language elements for the definition and access of mass storage files in which records are identified by the value of a key and accessed through an index. The elements of the Indexed I-O module are divided into two levels. Indexed I-O level 1 provides elements for the basic facilities of definition and access of indexed files. Indexed I-O level 2 provides elements for more complete facilities, including alternate keys and the capability of accessing the file both randomly and sequentially in the same COBOL program.

The Inter-Program Communication module contains the language elements which enable a program to communicate with one or more other programs. The elements of the Inter-Program Communication module are divided into two levels. Inter-Program Communication level 1 provides elements for the transfer of control to another program known at compile time; it also provides for the access of certain common data items by both programs. Inter-Program Communication level 2 provides elements for the transfer of control to another program not identified at compile time; it also provides for the nesting of programs within other programs.

The Sort-Merge module contains the language elements for the ordering of one or more files. The Sort-Merge module also contains the language elements for the combining of two or more identically ordered files. Optionally, a user may apply some special processing to each of the individual records by input or output procedures. The Sort-Merge module contains only level 1 elements.

The Source Text Manipulation module contains the language elements for the insertion and replacement of source program text as part of the compilation of the source program. The elements of the Source Text Manipulation module are divided into two levels. Source Text Manipulation level 1 provides the facility for copying text from a single library into the source program. Source Text Manipulation level 2 provides the additional capability of replacing library text during the copying process, specifying more than one COBOL library at compile time, and replacing source program text.

The Report Writer module contains the language elements for the semi-automatic production of printed reports. The Report Writer module contains only level 1 elements.

The Communication module contains the language elements to access, process, and create messages or portions thereof, and to communicate through a message control system with communication devices. The elements of the Communication module are divided into two levels. Communication level 1 provides elements for the basic facilities to send or receive complete messages. Communication level 2 provides elements for a more sophisticated facility including the capability to send or receive segments of a message.

The Debug module provides a means by which the user can specify his debugging algorithm -- the conditions under which data or procedure items are monitored during execution of the program. The elements of the Debug module are divided into two levels. Debug module level 1 provides a basic debugging capability, including the ability to specify selective or full paragraph monitoring. Debug module level 2 provides the full COBOL debugging capability.

The Segmentation module provides for the overlaying at object time of Procedure Division sections. The elements of the Segmentation module are divided into two levels. Segmentation level 1 provides for section segment-numbers and fixed segment limits. Segmentation level 2 adds the capability for varying the segment limit.

1.3 ORGANIZATION OF DOCUMENT

This document is divided into eighteen sections. Section I is composed of the introduction, a summary of elements by module, and a summary of elements by COBOL division. Section II presents concepts pertaining to the use and organization of features within the COBOL language. Section III is composed of a glossary defining terms in accordance with their meaning in COBOL.

Section IV contains a presentation of overall language considerations. Section V contains a composite language skeleton.

Sections VI through XVI contain specifications for the eleven functional processing modules. Within these sections, specifications unique to level 2 of the modules are enclosed in boxes.

Sections II through XVI comprise the detailed specifications of Standard COBOL.

Section XVII contains the appendices to the document. Section XVIII contains the index for the document.

1.4 HOW TO USE THE STANDARD

It is envisioned that the standard will be examined from several different viewpoints. In addition to the table of contents and the index, the summary of elements by module and the summary of elements by COBOL division are also intended to serve as a key to the standard.

Determination of the content of any level within a module is made from the summary of elements beginning on page I-10. This list contains a detailed breakdown of each element of Standard COBOL within a given module. For example, to ascertain the content of level 1 of the Sequential I-O module, reference is made to that module within the summary of elements by module (see page I-19). There will be found a list of COBOL elements including overall language considerations, Environment Division entries, Data Division entries, and Procedure Division verbs that pertain to the Sequential I-O module.

Determination of the modules and levels within modules in which a specific language feature appears is made from the summary of elements by COBOL division beginning on page I-40. This list shows in detail all elements of Standard

Introduction

COBOL and their occurrences within the various modules. Those elements which are not completely contained within one level of a module are shown in sufficient detail to specify the location of each subelement. For example, the READ statement appears in level 1 of the Sequential I-O module, the Relative I-O module, and the Indexed I-O module. Because certain phrases of the READ statement appear only in level 2 of these modules, the subelements of the READ statement are listed separately (see page I-59).

The schematic diagram on page I-5 is a graphic representation of the 11 functional processing modules forming the content of Standard COBOL. This schematic diagram shows the hierarchy of levels within each functional processing module. Within the schematic diagram a shorthand notation (such as 2 INX 0,2) indicates the hierarchical position of any level within the functional processing module as well as the number of levels into which the elements of the module have been divided. This shorthand notation is composed of, from left to right, a one-digit number indicating the level's position in the hierarchy, a three-character module abbreviation, and a two-digit number indicating the minimum and maximum levels of the module to which the level belongs. The number zero indicates a null level for the lowest level within a module. For example, 2 INX 0,2 indicates that this level is the second non-null level (level 2) of the Indexed I-O module which contains a null level and two non-null levels (level 1 and level 2). 2 NUC 1,2 indicates that this level is the second non-null level of the Nucleus module which contains two non-null levels (level 1 and level 2).

The three-character module abbreviations are as follows:

NUC	Nucleus
SEQ	Sequential I-O
REL	Relative I-O
INX	Indexed I-O
IPC	Inter-Program Communication
SRT	Sort-Merge
STM	Source Text Manipulation
RPW	Report Writer
COM	Communication
DEB	Debug
SEG	Segmentation



5-1

		REQUIRED MODULES (Required in Subsets)							OPTIONAL MODULES (Not Required in Subsets)			
		Nucleus	Sequential I-O	Relative I-O	Indexed I-O	Inter-Program Communication	Sort-Merge	Source Text Manipulation	Report Writer	Communication	Debug	Segmentation
COBOL SUBSETS	High	2 NUC 1,2	2 SEQ 1,2	2 REL 0,2	2 INX 0,2	2 IPC 1,2	1 SRT 0,1	2 STM 0,2	1 RPW 0,1	2 COM 0,2	2 DEB 0,2	2 SEG 0,2
	Intermediate	1 NUC 1,2	1 SEQ 1,2	1 REL 0,2	1 INX 0,2	1 IPC 1,2	1 SRT 0,1	1 STM 0,2				
	Minimum	1 NUC 1,2	1 SEQ 1,2	Null	Null	1 IPC 1,2	Null	Null		1 COM 0,2	1 DEB 0,2	1 SEG 0,2

1.5 DEFINITION OF AN IMPLEMENTATION OF STANDARD COBOL

This document provides a definition of the language features that comprise Standard COBOL. Standard COBOL consists of 11 modules, seven of which are required and four of which are optional. Paragraph 1.5 and its subparagraphs identify the criteria which must be met in order for a valid claim to be made that an implementation conforms to Standard COBOL.

1.5.1 Definition of Subsets

The three subsets of Standard COBOL are: the high subset, the intermediate subset, and the minimum subset. Each subset is composed of a level of the seven required modules: Nucleus, Sequential I-O, Relative I-O, Indexed I-O, Inter-Program Communication, Sort-Merge, and Source Text Manipulation. In terms of the schematic diagram on page I-5, a subset of Standard COBOL is represented by one of the three horizontal rows within the required module columns. The four optional modules (Report Writer, Communication, Debug, and Segmentation) are not required in the three subsets of Standard COBOL.

The high subset of Standard COBOL is composed of all language elements of the highest level of all required modules, that is:

- Level 2 elements from Nucleus, Sequential I-O, Relative I-O, Indexed I-O, Inter-Program Communication, and Source Text Manipulation
- Level 1 elements from Sort-Merge

The intermediate subset of Standard COBOL is composed of all language elements of level 1 of all required modules, that is:

- Level 1 elements from Nucleus, Sequential I-O, Relative I-O, Indexed I-O, Inter-Program Communication, Sort-Merge, and Source Text Manipulation

The minimum subset of Standard COBOL is composed of all language elements of level 1 of the Nucleus, Sequential I-O, and Inter-Program Communication modules.

1.5.2 Definition of a Conforming Implementation

A conforming implementation of Standard COBOL must fully support any of the three subsets defined in paragraph 1.5.1 above and may include none, all, or any combination of the levels of the optional modules.

A conforming implementation of a given subset of Standard COBOL must fully support all the language elements of that subset except as qualified in paragraph 1.5.2.5 on page I-8.

A conforming implementation of a given level of an optional module of Standard COBOL must fully support all the language elements of that level of the optional module except as qualified in paragraph 1.5.2.5 on page I-8.

Furthermore, any implementation must also meet the requirements of paragraphs 1.5.2.1 through 1.5.2.4.

1.5.2.1 Substitute or Additional Language Elements

An implementation must not require the inclusion of substitute or additional language elements in the source program in order to accomplish a function identical to that of a Standard COBOL language element. Additionally, throughout the Standard COBOL specification there are certain language elements whose syntax or function is specified to be, in part, implementor defined (see page XVII-87, Implementor-Defined Language Element List). While the implementor specifies the constraints on that portion of each element's syntax or rules that is indicated in Standard COBOL to be implementor defined, such constraints must not include any requirements for the inclusion in the source program of substitute or additional language elements.

1.5.2.2 Acceptance of Standard Language Elements

An implementation must accept the syntax and provide the function for all Standard COBOL language elements as specified in a given level of a module which is claimed as being included in the implementation, except those language elements dependent on specific hardware components which are specifically exempted by paragraph 1.5.2.5.1 on page I-8. When an implementation supports the syntax of Standard COBOL language elements from a given level of a module other than that for which support is claimed, that implementation must provide the function specified in Standard COBOL for that syntax or identify those language elements as nonstandard extensions (see paragraph 1.5.2.5.2 on page I-8).

1.5.2.3 Obsolete Language Elements

Obsolete language elements are identified as language elements in Standard COBOL which will be deleted from the next revision of Standard COBOL (see page XVII-81, Obsolete Language Element List). Obsolete language elements have been neither enhanced nor modified during the preparation of Standard COBOL. The interaction between obsolete language elements and other language elements is undefined unless otherwise specified in Standard COBOL. Language elements to be deleted from Standard COBOL will first be identified as obsolete language elements prior to being deleted.

A conforming implementation of Standard COBOL is required to support obsolete language elements of the subset and levels of optional modules for which support is claimed. Documentation associated with an implementation must identify all obsolete language elements in the implementation.

A conforming implementation of Standard COBOL must provide a warning mechanism, which optionally may be invoked by the user at compile time to indicate, if appropriate, that a program contains obsolete language elements (see page XVII-81).

1.5.2.4 Externally Provided Functions

If any function is provided outside the source program that accomplishes a function specified by a Standard COBOL language element contained in a given level of a module which is claimed as being included in an implementation, then the implementation must not require the specification of the external function in place of, or in addition to, that Standard COBOL language element.

Implementation Definition

An implementation may require specifications outside the source program to interface with the operating environment to support functions specified in a source program.

1.5.2.5 Qualifications

The following qualifications apply to an implementation of the Standard COBOL specifications:

1.5.2.5.1 Hardware Dependent Language Elements

There are certain language elements which pertain to specific types of hardware components (see page XVII-94, Hardware Dependent Language Element List). In order for an implementation to meet the requirements for this Standard COBOL, the implementor must specify the hardware components that the implementation supports. Furthermore, when support is claimed for a specific hardware component, all language elements that pertain to that component must be implemented if the module in which they appear is included in the implementation. Language elements that pertain to specific hardware components for which support is not claimed need not be implemented. The absence of such elements from an implementation of Standard COBOL must be specified.

1.5.2.5.2 Extension Language Elements

An implementation that includes language elements in addition to the subset and levels of optional modules for which support is claimed meets the requirements of Standard COBOL. This is true even though it may imply the extension of the list of reserved words by the implementor, and thereby may prevent proper translation of some programs that meet the requirements of Standard COBOL.

Documentation associated with an implementation must identify any standard extensions (language elements not defined in the supported subset or supported levels of optional modules but defined elsewhere in Standard COBOL) or nonstandard extensions (language elements or functions not defined in Standard COBOL) that are included in the implementation.

A conforming implementation of Standard COBOL must provide a warning mechanism, which optionally may be invoked by the user at compile time to indicate, if appropriate, that a program contains nonstandard extensions that are included in the implementation.

1.5.2.5.3 Reserved Words

An implementation of Standard COBOL must recognize as reserved words all of the COBOL reserved words occurring in the specification of the seven required modules and the four optional modules. (See page IV-45, COBOL Reserved Words.)

1.5.2.5.4 Character Substitution

The definition of the COBOL character set on page III-3 presents the complete COBOL character set for Standard COBOL. When an implementation does not provide for a graphic representation for all the COBOL character set, substitute graphics may be specified by the implementor to replace the characters not represented.

1.5.2.5.5 The ENTER Statement

An implementation of Standard COBOL may include the ENTER statement or not, at the option of the implementor.

1.6 DEFINITION OF A CONFORMING SOURCE PROGRAM

A conforming source program is one which does not violate the explicitly stated provisions and specifications of Standard COBOL. In order for a source program to conform to Standard COBOL, it must not include any language elements not specified in this standard. The execution of a program, the source text of which conforms to Standard COBOL, is predictable only to the extent defined in this standard. The results of violating the formats or rules of Standard COBOL are undefined unless otherwise specified in this standard.

In order for a source program to conform to a specified subset of Standard COBOL, it must include only language elements of that subset.

There are, in Standard COBOL, situations in which the results of executing a statement are undefined or unpredictable (see page XVII-96, Undefined Language Element List). A COBOL source program which allows this to happen may nevertheless be a conforming program, although the resultant execution is not defined by Standard COBOL.

1.7 RELATIONSHIP OF A CONFORMING PROGRAM TO A CONFORMING IMPLEMENTATION

The translation of a conforming source program by a conforming implementation and the subsequent execution of the resultant object program is defined only to the extent specified in Standard COBOL. However, the preceding statement does not imply that the program will be translated or executed successfully; translation and execution depends on other factors, such as the use of implementor-defined language elements, the logical correctness of the program, and the data upon which the program operates.

In general, Standard COBOL specifies no upper limit on such things as the number of statements in a program and the number of operands permitted in certain statements. It is recognized that these limits will vary from one implementation of Standard COBOL to another and may prevent the successful translation of some programs that meet the requirements of Standard COBOL.

2. SUMMARY OF ELEMENTS BY MODULE

2.1 GENERAL DESCRIPTION

This chapter contains a summary of all elements in Standard COBOL organized according to the functional processing modules.

The column titled "LEVEL 1" specifies the level 1 elements of the module. The column titled "LEVEL 2" specifies the level 2 elements of the module.

The letter X in a column indicates the presence of the specified element within the specified level of the module. The letter N in a column indicates the absence of the specified element from the specified level of the module. The letter Z in a column indicates the presence of the specified element within the specified level of the module; however this element is an obsolete element in Standard COBOL because it is to be deleted from the next revision of Standard COBOL.

The following is a list of the summary of elements by module shown on pages I-11 through I-39.

- Pages I-11 through I-18: Summary of elements in the Nucleus module
- Pages I-19 through I-21: Summary of elements in the Sequential I-O module
- Pages I-22 through I-24: Summary of elements in the Relative I-O module
- Pages I-25 through I-27: Summary of elements in the Indexed I-O module
- Pages I-28 and I-29: Summary of elements in the Inter-Program
Communication module
- Pages I-30 and I-31: Summary of elements in the Sort-Merge module
- Page I-32: Summary of elements in the Source Text Manipulation module
- Pages I-33 through I-35: Summary of elements in the Report Writer module
- Pages I-36 and I-37: Summary of elements in the Communication module
- Page I-38: Summary of elements in the Debug module
- Page I-39: Summary of elements in the Segmentation module

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

ELEMENT	LEVEL 1	LEVEL 2
<u>LANGUAGE CONCEPTS</u>		
<u>Character Set</u>		
Characters used in words 0-9 A-Z - (hyphen)	X	X
Characters used in punctuation " () . , ; space	X	X
Characters used in punctuation : (colon)	N	X
Characters used in editing B + - . , Z * \$ 0 CR DB /	X	X
Characters used in arithmetic operations + - * / **	N	X
Characters used in relation conditions = > < >= <=	X	X
Characters used in subscripting + -	X	X
Single character substitution allowed	X	X
Double character substitution allowed	Z	Z
<u>Separators</u>		
" () . , ; space	X	X
: (colon)	N	X
<u>Character-Strings</u>		
COBOL words		
Maximum of 30 characters	X	X
User-defined words		
Alphabet-name	X	X
Class-name	X	X
Condition-name	N	X
Data-name	X	X
Index-name '.....	X	X
Level-number	X	X
Mnemonic-name	X	X
Paragraph-name	X	X
Program-name	X	X
Routine-name	Z	Z
Section-name	X	X
Symbolic-character	N	X
System-names		
Computer-name	X	X
Implementor-name	X	X
Language-name	Z	Z
Reserved words		
Required words	X	X
Key words	X	X
Special character words		
Arithmetic operators + - * / **	N	X
Arithmetic operators used in subscripting + -	X	X
Relation characters = > < >= <=	X	X
Optional words	X	X

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

<u>ELEMENT</u>	<u>LEVEL 1</u>	<u>LEVEL 2</u>
<u>Character-Strings (Continued)</u>		
Special purpose words		
Figurative constants		
ZERO, SPACE, HIGH-VALUE, LOW-VALUE, QUOTE	X	X
ALL option	N	X
ZEROS, ZEROES, SPACES, HIGH-VALUES, LOW-VALUES, QUOTES ...	X	X
ALL option	N	X
Symbolic-character	N	X
ALL option	N	X
ALL literal	N	X
Literals		
Numeric literals: 1 through 18 digits	X	X
Nonnumeric literals: 1 through 160 characters	X	X
PICTURE character-string	X	X
Comment-entries	Z	Z
<u>Uniqueness of Reference</u>		
Qualification		
No qualification permitted; names must be unique if		
referenced	X	N
50 qualifiers	N	X
Subscripting		
3 levels of subscripts	X	N
7 levels of subscripts	N	X
Subscripting with a literal	X	X
Subscripting with a data-name	X	X
Subscripting with an index-name	X	X
Relative subscripting	X	X
Reference modification	N	X
<u>Reference Format</u>		
Sequence number	X	X
Continuation of lines		
Continuation of nonnumeric literal	X	X
Continuation of COBOL word, numeric literal,		
PICTURE character-string	N	X
Blank lines	X	X
Comment lines		
Asterisk (8) comment line	X	X
Slant (/) comment line	X	X
Debugging line with D in indicator area	X	X
<u>Source Program Structure</u>		
Identification Division required	X	X
Environment Division optional	X	X
Data Division optional	X	X
Procedure Division optional	X	X
End program header	N	X

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

<u>ELEMENT</u>	<u>LEVEL 1</u>	<u>LEVEL 2</u>
<u>IDENTIFICATION DIVISION</u>		
PROGRAM-ID paragraph	X	X
Program-name	X	X
AUTHOR paragraph	Z	Z
INSTALLATION paragraph	Z	Z
DATE-WRITTEN paragraph	Z	Z
DATE-COMPILED paragraph	N	Z
SECURITY paragraph	Z	Z
<u>ENVIRONMENT DIVISION</u>		
<u>Configuration Section</u>	X	X
SOURCE-COMPUTER paragraph	X	X
Computer-name	X	X
WITH DEBUGGING MODE clause	X	X
OBJECT-COMPUTER paragraph	X	X
Computer-name	X	X
MEMORY SIZE clause	Z	Z
PROGRAM COLLATING SEQUENCE clause	X	X
SPECIAL-NAMES paragraph	X	X
ALPHABET clause	X	X
STANDARD-1 option	X	X
STANDARD-2 option	X	X
NATIVE option	X	X
Implementor-name option	X	X
Literal option	N	X
CLASS clause	X	X
CURRENCY SIGN clause	X	X
DECIMAL-POINT clause	X	X
Implementor-name clause	X	X
IS mnemonic-name option	X	X
ON STATUS IS condition-name option	X	X
OFF STATUS IS condition-name option	X	X
SYMBOLIC CHARACTERS clause	N	X
<u>DATA DIVISION</u>		
<u>Working-Storage Section</u>	X	X
Record description entry	X	X
77 level description entry	X	X
Data description entry	X	X
BLANK WHEN ZERO clause	X	X
Data-name clause	X	X
FILLER clause	X	X
JUSTIFIED clause	X	X
Level-number clause	X	X
01 through 49; level-number may be 1 or 2 digits	X	X
66	N	X
77	X	X
88	N	X

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

ELEMENT	LEVEL 1	LEVEL 2
OCCURS clause	X	X
Integer TIMES	X	X
ASCENDING/DESCENDING KEY phrase	N	X
INDEXED BY phrase	X	X
Integer-1 TO integer-2 TIMES DEPENDING ON phrase	N	X
PICTURE clause	X	X
Character-string has a maximum of 30 characters	X	X
Data characters X 9 A	X	X
Operational symbols S V P	X	X
Nonfloating insertion characters B + - . , \$ 0 CR DB /	X	X
Replacement or floating insertion characters \$ + - Z *	X	X
Currency sign substitution	X	X
Decimal point substitution	X	X
REDEFINES clause	X	X
May not be nested	X	N
May be nested	N	X
RENAMES clause	N	X
SIGN clause	X	X
SYNCHRONIZED clause	X	X
USAGE clause	X	X
BINARY	X	X
COMPUTATIONAL	X	X
DISPLAY	X	X
INDEX	X	X
PACKED-DECIMAL	X	X
VALUE clause	X	X
Literal	X	X
Literal series	N	X
Literal-1 THROUGH literal-2	N	X
Literal range series	N	X
<u>PROCEDURE DIVISION</u>		
Arithmetic expression	N	X
Binary arithmetic operators + - * / **	N	X
Unary arithmetic operators + -	N	X
Conditional expressions	X	X
Simple condition	X	X
Relation condition	X	X
Relational operators	X	X
[NOT] GREATER THAN	X	X
[NOT] >	X	X
[NOT] LESS THAN	X	X
[NOT] <	X	X
[NOT] EQUAL TO	X	X
[NOT] = ...	X	X
GREATER THAN OR EQUAL TO	X	X
>=	X	X
LESS THAN OR EQUAL TO	X	X
<=	X	X
Comparison of numeric operands	X	X

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

ELEMENT	LEVEL 1	LEVEL 2
Conditional expressions (continued)		
Simple condition (continued)		
Comparison of nonnumeric operands	X	X
Comparison of index-names and/or index data items	X	X
Class condition	X	X
NUMERIC	X	X
ALPHABETIC	X	X
ALPHABETIC-LOWER	X	X
ALPHABETIC-UPPER	X	X
Class-name	X	X
Condition-name condition	N	X
Sign condition	N	X
Switch-status condition	X	X
Complex condition	N	X
Logical operators AND OR NOT	N	X
Negated condition	N	X
Combined condition	N	X
Parenthesized conditions	X	X
Abbreviated combined relation conditions	N	X
Arithmetic statements	X	X
Arithmetic operands limited to 18 digits	X	X
Composite of operands limited to 18 digits	X	X
ACCEPT statement	X	X
Identifier	X	X
Only one transfer of data	X	N
No restriction on number of transfers of data	N	X
FROM mnemonic-name phrase	N	X
FROM DATE/DAY/DAY-OF-WEEK/TIME phrase	N	X
ADD statement	X	X
Identifier/literal	X	X
Identifier/literal series	X	X
TO identifier	X	X
TO identifier series	X	X
TO identifier/literal GIVING identifier	X	X
TO identifier/literal GIVING identifier series	X	X
ROUNDED phrase	X	X
ON SIZE ERROR phrase	X	X
NOT ON SIZE ERROR phrase	X	X
END-ADD phrase	X	X
CORRESPONDING phrase	N	X
ALTER statement	Z	Z
Only one procedure-name	Z	N
Procedure-name series	N	Z
COMPUTE statement	N	X
Arithmetic expression	N	X
Identifier series	N	X
ROUNDED phrase	N	X
ON SIZE ERROR phrase	N	X
NOT ON SIZE ERROR phrase	N	X
END-COMPUTE phrase	N	X

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

ELEMENT	LEVEL 1	LEVEL 2
CONTINUE statement	X	X
DISPLAY statement	X	X
Only one transfer of data	X	N
No restriction on number of transfers of data	N	X
Identifier/literal	X	X
Identifier/literal series	X	X
UPON mnemonic-name phrase	N	X
WITH NO ADVANCING phrase	N	X
DIVIDE statement	X	X
BY identifier/literal	X	X
INTO identifier	X	X
INTO identifier series	X	X
GIVING identifier	X	X
GIVING identifier series	X	X
ROUNDED phrase	X	X
REMAINDER phrase	N	X
ON SIZE ERROR phrase	X	X
NOT ON SIZE ERROR phrase	X	X
END-DIVIDE phrase	X	X
ENTER statement	Z	Z
EVALUATE statement	N	X
Identifier/literal	N	X
Arithmetic expression	N	X
Conditional expression	N	X
TRUE/FALSE	N	X
ALSO phrase	N	X
WHEN phrase	N	X
ALSO phrase	N	X
WHEN OTHER phrase	N	X
END-EVALUATE phrase	N	X
EXIT statement	X	X
GO TO statement	X	X
Procedure-name is required	X	N
Procedure-name is optional	N	Z
DEPENDING ON phrase	X	X
IF statement	X	X
Only imperative statements	X	N
Imperative and/or conditional statements	N	X
Nested IF statements	X	X
THEN optional word	X	X
NEXT SENTENCE phrase	X	X
ELSE phrase	X	X
END-IF phrase	X	X
INITIALIZE statement	N	X
Identifier series	N	X
REPLACING phrase	N	X
REPLACING series	N	X

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

ELEMENT	LEVEL 1	LEVEL 2
INSPECT statement	X	X
Only single character data item	X	N
Multi-character data item	N	X
TALLYING phrase	X	X
BEFORE/AFTER phrase	X	X
BEFORE/AFTER phrase series	N	X
TALLYING phrase series	N	X
REPLACING phrase	X	X
BEFORE/AFTER phrase	X	X
BEFORE/AFTER phrase series	N	X
REPLACING phrase series	N	X
TALLYING and REPLACING phrases	X	X
CONVERTING phrase	N	X
MOVE statement	X	X
TO identifier	X	X
TO identifier series	X	X
De-editing of numeric edited items	N	X
CORRESPONDING phrase	N	X
MULTIPLY statement	X	X
BY identifier	X	X
BY identifier series	X	X
GIVING identifier	X	X
GIVING identifier series	X	X
ROUNDED phrase	X	X
ON SIZE ERROR phrase	X	X
NOT ON SIZE ERROR phrase	X	X
END-MULTIPLY phrase	X	X
PERFORM statement	X	X
Procedure-name is optional	X	X
THROUGH procedure-name phrase	X	X
Imperative-statement option	X	X
END-PERFORM phrase	X	X
TIMES phrase	X	X
UNTIL phrase	X	X
TEST BEFORE/AFTER phrase	N	X
VARYING phrase	N	X
TEST BEFORE/AFTER phrase	N	X
AFTER phrase	N	X
At least 6 AFTER phrases permitted	N	X
SEARCH statement	N	X
VARYING phrase	N	X
AT END phrase	N	X
WHEN phrase	N	X
WHEN phrase series	N	X
END-SEARCH phrase	N	X
SEARCH ALL statement	N	X
AT END phrase	N	X
WHEN phrase	N	X
END-SEARCH phrase	N	X

SUMMARY OF ELEMENTS IN THE NUCLEUS MODULE

ELEMENT	LEVEL 1	LEVEL 2
SET statement	X	X
Index-name/identifier TO	X	X
Index-name UP BY/DOWN BY	X	X
Mnemonic-name TO ON/OFF	X	X
Condition-name TO TRUE	N	X
STOP statement	X	X
RUN	X	X
Literal	Z	Z
STRING statement	N	X
DELIMITED BY series	N	X
WITH POINTER phrase	N	X
ON OVERFLOW phrase	N	X
NOT ON OVERFLOW phrase	N	X
END-STRING phrase	N	X
SUBTRACT statement	X	X
Identifier/literal	X	X
Identifier/literal series	X	X
FROM identifier	X	X
FROM identifier series	X	X
GIVING identifier	X	X
GIVING identifier series	X	X
ROUNDED phrase	X	X
ON SIZE ERROR phrase	X	X
NOT ON SIZE ERROR phrase	X	X
END-SUBTRACT phrase	X	X
CORRESPONDING phrase	N	X
UNSTRING statement	N	X
DELIMITED BY phrase	N	X
DELIMITER IN phrase	N	X
COUNT IN phrase	N	X
WITH POINTER phrase	N	X
TALLYING phrase	N	X
ON OVERFLOW phrase	N	X
NOT ON OVERFLOW phrase	N	X
END-UNSTRING phrase	N	X

SUMMARY OF ELEMENTS IN THE SEQUENTIAL I-O MODULE

<u>ELEMENT</u>	<u>LEVEL 1</u>	<u>LEVEL 2</u>
<u>LANGUAGE CONCEPTS</u>		
User-defined words		
File-name	X	X
Record-name	X	X
Reserved words		
Special register LINAGE-COUNTER	N	X
I-O status	X	X
<u>ENVIRONMENT DIVISION</u>		
<u>Input-Output Section</u>		
FILE-CONTROL paragraph	X	X
File control entry	X	X
SELECT clause	X	X
OPTIONAL phrase	N	X
Input, I-O, and extend files only	N	X
ACCESS MODE IS SEQUENTIAL clause	X	X
ASSIGN clause	X	X
Implementor-name	X	X
Literal	X	X
FILE STATUS clause	X	X
ORGANIZATION IS SEQUENTIAL clause	X	X
PADDING CHARACTER clause	N	X
RECORD DELIMITER clause	N	X
RESERVE AREA clause	N	X
I-O-CONTROL paragraph	X	X
MULTIPLE FILE TAPE clause	N	Z
RERUN clause	Z	Z
SAME AREA clause	X	X
SAME RECORD AREA clause	N	X
<u>DATA DIVISION</u>		
<u>File Section</u>		
File description entry	X	X
FD level indicator	X	X
BLOCK CONTAINS clause	X	X
Integer RECORDS/CHARACTERS	X	X
Integer-1 TO integer-2 RECORDS/CHARACTERS	N	X
CODE-SET clause	X	X
DATA RECORDS clause	Z	Z
LABEL RECORDS clause	Z	Z
LINAGE clause	N	X
FOOTING phrase	N	X
TOP phrase	N	X
BOTTOM phrase	N	X
RECORD clause	X	X
Integer-1 CHARACTERS	X	X
VARYING IN SIZE phrase	N	X
FROM integer-2 TO integer-3 CHARACTERS	N	X
DEPENDING ON phrase	N	X
Integer-4 TO integer-5 CHARACTERS	X	X

SUMMARY OF ELEMENTS IN THE SEQUENTIAL I-O MODULE

<u>ELEMENT</u>	<u>LEVEL 1</u>	<u>LEVEL 2</u>
File description entry (continued)		
VALUE OF clause	Z	Z
Implementor-name IS literal	Z	Z
Implementor-name IS literal series	Z	Z
Implementor-name IS data-name	N	Z
Implementor-name IS data-name series	N	Z
Record description entry	X	X
<u>PROCEDURE DIVISION</u>		
Declarative procedures	X	X
DECLARATIVES	X	X
END DECLARATIVES	X	X
CLOSE statement	X	X
File-name	X	X
File-name series	X	X
REEL/UNIT phrase	X	X
FOR REMOVAL phrase	N	X
WITH NO REWIND/LOCK phrase	N	X
OPEN statement	X	X
File-name	X	X
File-name series	X	X
INPUT phrase	X	X
WITH NO REWIND phrase	N	X
REVERSED phrase	N	Z
OUTPUT phrase	X	X
WITH NO REWIND phrase	N	X
I-O phrase	X	X
EXTEND phrase	N	X
INPUT, OUTPUT, and I-O series	X	X
EXTEND series	N	X
READ statement	X	X
NEXT phrase	N	X
INTO phrase	X	X
AT END phrase	X	X
NOT AT END phrase	X	X
END-READ phrase	X	X
REWRITE statement	X	X
FROM phrase	X	X
END-REWRITE phrase	X	X
USE statement	X	X
EXCEPTION/ERROR PROCEDURE phrase	X	X
ON file-name	X	X
ON file-name series	N	X
ON INPUT	X	X
ON OUTPUT	X	X
ON I-O	X	X
ON EXTEND	N	X