File Number S360-24
Form C28-6570-0

IBM Systems Reference Library

IBM System/360 Transition Aids

COBOL Language Differences

This publication summarizes the System/360 and Current Systems specifications for major IBM COBOL language elements requiring evaluation, and possible conversion, during the transition to IBM System/360 COBOL. System/360 COBOL differences from Current Systems COBOL are also summarized. Where pertinent, the distinction is made between Level E and Level F COBOL for System/360.

PREFACE

This publication provides a convenient reference to the major language differences between the IBM Current Systems COBOL languages and the IBM System/360 COBOL language.

The information presented in this publication was drawn from the existing IBM COBOL Language manuals (listed under "Prerequisite Knowledge") that contain the specifications for all the Current Systems compilers and the System/360 COBOL compiler. The material presented is entirely dependent upon these publications and derives its authority from them. In cases of conflict this publication does not supersede any of the publications.

PREREQUISITE KNCWLEDGE

The reader of this publication should have a basic knowledge of the COBOL language, as well as a working knowledge of at least one IBM COBOL compiler.

System/360 COBOL is described in IBM System/360 Operating Systems; COBOL Language, Form C28-6516-2.

The basic COBOL language implemented by the IBM Current Systems compilers is described in Common Business Oriented Language (COBOL); General Information Form F28-8053. In most instances, the specifications contained in the tables in this publication are in addition to the specifications contained in the general information manual. Therefore, a thorough familiarity with the general information manual is essential to the effective use of this publication.

Additional specifications for the individual Current Systems Compilers are contained in the following publications:

COBOL (on Tape) Specifications; IBM 1401, Form C24-1492-1.
COBOL (on Disk) Specifications; IBM 1401, 1440, and 1460, Form C24-3235-1.
IBM 1410/7010 Operating System; COBOL, Form C28-0327-2.
IBM 7040/7044 Operating Systems (16/32K); COBOL Language, Form C28-6336-2.
IBM 7070-Series Programming Systems; COBOL Processor, Form J28-6182-2.
IBM 7070-Series COBOL/FORTRAN Operating System; COBOL, Form C28-6901-1.
IBM 705/7080 Programming Systems; COBOL: Additional Specifications, Form J28-6177-3.
IBM 7090/7094 IBSYS Operating System-Version 13; COBOL Language, Form C28-6391-0. (Note: For the purposes of this publication, differences are based on Version 5 of the 7090/7094 COBOL compiler.)

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained for an IBM 1403 Printer with a 120-character chain containing upper and lower case letters, special characters, and numerals.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form for readers' comments appears at the back of this publication. Address any additional comments concerning this publication to the IBM Corporation, Programming Systems Publications, Dept. D39, 1271 Avenue of the Americas, New York, N.Y., 10020.

CONTENTS

ACKNOWLEDGMENT

In accordance with a request contained in COBOL-1961-Extended, the following extract from that manual is presented for the information and guidance of the user:

This publication is based on the COBOL System developed in 1959 by a committee composed of government users and computer manufacturers. The organizations participating in the original development were:

Air Materiel Command, United States Air Force
Bureau of Standards, Department of Commerce
David Taylor Model Basin, Bureau of Ships, U. S. Navy
Electronic Data Processing Division, Minneapolis-Honeywell Regulator Company
Burroughs Corporation
International Business Machines Corporation
Radio Corporation of America
Sylvania Electric Products, Inc.
Univac Division of Sperry-Rand Corporation

In addition to the organizations listed above, the following other organizations participated in the work of the Maintenance Group:
Allstate Insurance Company
Bendix Corporation, Computer Division
Control Data Corporation
DuPont Company
General Electric Company
General Motors Corporation
Lockheed Aircraft Corporation
National Cash Register Company
Philco Corporation
Royal McBee Corporation
Standard Oil Company (N. J.)
United States Steel Corporation

This manual is the result of contributions made by all of the above-mentioned organizations. No warranty, expressed or implied, is made by any contributor or by the 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.

It is reasonable to assume that a number of improvements and additions will be made to COBOL. Every effort will be made to insure that the improvements and corrections will be made in an orderly fashion, with due recognition of existing users' investments in programming. However, this protection can be positively assumed only by individual implementors.

Procedures have been established for the maintenance of COBOL. Inquiries concerning the procedures and the methods for proposing changes should be directed to the Executive Committee of the Conference on Data Systems Languages.

The authors and copyright holders of the copyrighted material used herein: FLOW-MATIC (Trade-mark of Sperry-Rand Corporation), Programming for the UNIVAC R I and II, Data Automation Systems c 1958, 1959, Sperry-Rand Corporation; IBM Commercial Translator, Form No. F28-8013, copyrighted 1959 by IBM, FACT, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell, have specifically authorized the

The full potential of the expanded IBM System/360 capability may best be realized by completely redesigning rather than by duplicating current system programs. However, programs written in higher-level languages, such as COBOL, can in most cases be run on System/360 with minor changes. The use of the tables, as indicated in the section "COBOL Differences," should make it possible to annotate an existing program, or to design a projected one, with a minimum of research. Maximum attention could then be given to the task of investigating and resolving conversion problems that are beyond the scope and intention of this publication. In general, there are three major areas -- translation, interpretation, and environment -- that should be considered before attempting cross-computer compatibility.

TRANSLATION

The first area of conversion is the most straightforward: translation. Because the COBOL language is a common language, the restrictions and extensions imposed on it by individual compilers are few, compared to the broad base that is common to all compilers. Thus, the translation of language formats, while important, does not represent a significant portion of the entire problem. System/360 has maintained the structure and meaning of the basic COBOL language. However, it has restricted some redundant variations for the sake of speed, has eliminated some vocabulary for the sake of conciseness, and has provided some new vocabulary for the sake of greater capability. The tables in this publication will indicate the important translations that will be required when conversion is sought.

INTERPRETATION

The second area of conversion, interpretation, is somewhat more complex because it involves the problem of how the COBOL language is implemented by various compilers. It is not enough to translate the words; the sense must be translated, as well. When converting a source program from one format to another, great care must be taken to ensure that language elements are compatible in specification and intent, as well as in written format.

The question of interpretation must be considered in the light of the programming principles and concepts that characterize a particular environment. System/360 COBOL is implemented in conjunction with the System/360 Operating System, which employs a concept of device-independence and provides full subprogram capability. System/360 COBOL incorporates a technique, called asynchronous processing, that provides COBOL programs with the capability for multiprogramming. These extended implementations have, necessarily, imposed new and different interpretations on some of the basic COBOL language elements. The tables in this publication will make it possible to compare, in terms of individual compiler specifications, the major language elements that are conversion-sensitive.

ENVIRONMENT

The third area of conversion is the area of environment. It is only to be expected that a change in environment will necessitate a careful evaluation of those program elements that are environment-oriented. In particular, the following computer characteristics warrant detailed attention:

Character Set
Collating Sequence
Data Classification
Data Representation
Input-Output
Processing Characteristics
Storage Characteristics

The tables in this publication do not include a detailed treatment of environmental differences. The Environment Division is included merely to indicate the necessity for complete rewriting whenever a program is to be processed for a different computer. A complete discussion of environmental language elements should be sought in the applicable IBM COBOL publications.

The COBOL Differences to be found in the tables are, in fact, the COBOL language elements that differ in format or specification between one or more Current Systems and System/360. The language elements are listed alphabetically in the first column of Table 2. The supplementary columns are intended to provide sufficient information relating to each language element in order that the conversion problem posed by the element can be readily determined.

AN EXPLANATION OF THE COLUMNS

Compiler: The following compilers are referred to in the tables:

Abbreviations	COBOL Compiler
1401/40/60	IBM 1401/1460 (on Tape)
	IBM 1401/1440/1460 (on Disk)
1410/7010	IBM 1410/7010 Operating System
7040/7044	IBM 7040/7044 Operating Systems
7070/7074	IBM 7070-Series Programming Systems
7070-FCS	IBM 7070-Series COBOL/FORTRAN Operating System
705/7080	IBM 705/7080 Programming Systems
7090/7094	IBM 7090/7094 Operating System
System/360	Operating System/360

COBOL Language Element: A generic element of the COBOL language; e.g.,

ADD Statement
File Description Entry
SIZE ERROR Condition
USE Sentence

In some cases, for convenience, entries have been included in this column which are not, properly speaking, elements of the COBOL language (for example, CALL Statement).

In general, the presence of a language element in Table 2 indicates that a difference is associated with the element. Furthermore, the presence of certain elements in the table is related to the presence of others. For example, the RECORD CONTAINS Clause, as specified by the 1410/7010 compiler, differs from the System/360 RECORD CONTAINS Clause. One difference lies in the fact that the clause is required in a 1410/7010 File Description Entry, while it is optional in System/360. Therefore, the File Description Entry itself is associated with a difference, and is included in the language element column.

Syntax: This column outlines the basic format of a language element, as implemented by at least one compiler. If the language element is abstract (for example, Collating Sequence), then this column is used to itemize or summarize the particulars concerning the element. The format notation used is described at the end of this section. However, the notation is intended only to indicate the general language formats, and does not supersede the format specifications contained in the applicable IBM COBOL publications.

Additional Specifications or Comment: Specifications indicated in this column are condensed from the applicable IBM COBOL publications and, in

general, represent extensions to the general information manual specifications. Only such specifications as are indicative of a possible difference are denoted. For example, the specifications for the SIZE Clause are limited to the actions taken by the various Current Systems compilers when the clause is omitted, since the clause must be omitted for compatibility with System/360. The fact that System/360 does not implement the clause is indicated by a comment to that effect.

Key Word or Phrase: Any important reserved COBOL word, or descriptive phrase, that appears in the column headed "Syntax" can be refered to as a "Key Word or Phrase."

AN EXPLANATION OF THE TABLES

Table 1, Compiler Index to COBOL Differences: This table is designed to provide a recapitulation of differences for each compiler. By referring to the elements listed for his compiler, a programmer can quickly determine which language elements pose a possible conversion problem.

For the Current Systems compilers, differences are always with respect to System/360. For System/360, differences are with respect to at least one Current Systems compiler. For example, the ACCEPT Statement is listed for the 1401/40/60 compilers. This means that the 1401/40/60 ACCEPT Statement is not completely compatible with the System/360 ACCEPT Statement. For particulars concerning the difference, the ACCEPT Statement entry in Table 2 should be consulted. On the other hand, the fact that ACCEPT Statement is listed for System/360 indicates that the System/360 ACCEPT Statement is incompatible with at least one Current Systems compiler.

Table 2, COBOL Differences: The information contained in this table constitutes a complete presentation of COBOL language differences, insofar as they are discussed in this publication. Each conversion-sensitive language element is listed alphabetically in the first column, together with the syntax(es) and specifications associated with the compiler(s) involved.

Whenever possible, entries that indicated similarities were eliminated from the table; thus, the table is intended only to consider differences.

For example, the 7070-FOS compiler is not listed for the NOTE Statement entry, since the 7070-FOS compiler's NOTE Statement is completely compatible with System/360, in both directions. That is, a 7070-FOS NOTE Statement is always compatible with System/360, and a System/360 NOTE Statement is always compatible with the 7070-FOS compiler.

However, omission does not always imply compatibility. The fact that the 1401/1440/1460 compilers are not listed for the USAGE clause does not mean that there is complete compatibility with System/360. In fact, the 1401/1440/1460 compilers do not implement a USAGE clause. The ambiguity of omission may be resolved by referring to an element in the table that subordinates the element in question. Thus, by referring to the entry "Procedure Verbs" it may be determined that the 7070-FOS compiler implements the NOTE verb. Similarly, by referring to the entry "Record Description Entry" it may be determined that 1401/1440/1460 compiles do not implement the USAGE clause.

It is not intended that all the information associated with a particular entry in the table is essential for an individual application. If the direction of conversion is from System/360 to a

current compiler, the information associated with the current compiler in question is the most relevant. On the other hand, if the direction is to System/360, then the System/360 syntax and specifications are of the most importance.

Table 3, Key Word Index to COBOL Differences: In order to assist the reader in locating information contained in Table 2, all important COBOL words, and descriptive phrases, which appear anywhere in the "Syntax" column of Table 2, are listed in this table, indicating the language element entry under which they are discussed. The table may be useful in determining the nomenclature used for various language elements. For example, if information is required pertaining to "margins", reference to this table would indicate that "Margin Restrictions" are discussed under the entry "COBOL Program Sheet" in Table 2.

EXCLUSIONS

The following areas of conversion are excluded from consideration in this publication:

1. Environment-oriented programming considerations are either excluded entirely from the tables, or are treated only generally, merely to indicate their existence.
2. The problem of data conversion is excluded from the tables.
3. The tables do not take the peculiarities of particular compilers into account. If a compiler tolerates liberties taken with its specifications, this publication does not discuss such liberties.
4. For the purposes of conversion, warning messages issued during compilation of a program signify an error and the proper steps should be taken to eliminate them.

FORMAT NOTATION

Throughout Table 2 of this publication, basic formats are used for various elements of COBOL, as explained in the following paragraphs. This notation is useful in describing COBOL, although it is not part of COBOL.

1. All words printed entirely in capital letters are reserved words. These are words which have preassigned meanings in the COBOL language. In all formats, words in capital letters represent an actual occurrence of those words.
2. All underlined reserved words are required unless the portion of the format containing them is itself optional. These are key words. If any such word is missing or is incorrectly spelled, it is considered an error in the program. Reserved words not underlined may be included or omitted at the option of the programmer. These words are used only for the sake of readability. These words are called optional words.
3. All lower case words represent information which must be supplied by the programmer.
4. Certain groups of words in the formats consist of capitalized portions followed by lower-case or initially capitalized portions. These designate clauses, statements, paragraphs, or entries whose formats are not pertinent to the overall format being described.
5. When material is enclosed in braces ({ }), one, and only one, of the enclosed items is required; the others are to be omitted. The choice is to be determined by the programmer.
6. Square brackets ([ ]) are used to indicate that the enclosed item

may be used or omitted, depending on the requirements of the particular program. When two or more items are stacked within brackets, one or none of them may occur.

7. The ellipsis (...) indicates:
   For Systems/360 formats: that the immediately preceding unit may occur once, or any number of times in succession. A unit means either a single lower-case word, or a group of lower-case words and one or more reserved words enclosed in brackets or braces. If a term is enclosed in brackets or braces, the entire unit of which it is a part must be repeated when repetition is specified.
   For Current Systems formats: that the last complete element preceding it may occur once, or any number of times in succession. If a group of operands and key words are enclosed within brackets the entire group must be repeated if any repetition is required, not merely the last operand.

Table 1. Compiler Index to COBOL Differences

COMPILER	COBOL LANGUAGE ELEMENT
System/360	ACCEPT Statement
	ADD Statement
	Alphabetic Item
	Alphanumeric Item
	Arithmetic Expression
	AT END Condition
	Binary Item
	BLANK Clause
	BLOCK CONTAINS Clause
	CALL Statement
	Character Set
	Class Test
	CLOSE Statement
	Collating Sequence
	Conditional Statement
	Continuation Indicator
	DATA DIVISION
	DATA RECORDS Clause
	Declaratives
	DISPLAY Statement
	ENTER Statement
	ENTRY Statement
	ENVIRONMENT DIVISION
	Ext. Floating Point Item
	External Decimal Item
	File Description Entry
	GO TO Statement
	Group Item
	Identification Code
	IDENTIFICATION DIVISION
	Int. Floating Point Item

COMPILER	COBOL LANGUAGE ELEMENT
System/360	Internal Decimal Item
	INVALID KEY Condition
	JUSTIFIED Clause
	LABEL RECORDS Clause
	Literals
	MOVE Statement
	MULTIPLY Statement
	OCCURS Clause
	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	Procedure Verbs
	Punctuation
	Qualification
	READ Statement
	RECORD CONTAINS Clause
	Record Description Entry
	REDEFINES Clause
	Relation Test
	Report Item
	Reserved S/360 Only
	RETURN Statement
	Series Separator
	SIZE ERROR Condition
	STOP Statement
	SUBTRACT Statement
	SYNCHRONIZED Clause
	Test Conditions
	USAGE Clause
	USE Sentence
	VALUE Clause
	VALUE OF Clause
	WRITE Statement

Table 1. Compiler Index to COBOL Differences

COMPILER	COBOL LANGUAGE ELEMENT
1401/40/60	ACCEPT Statement
	ADD Statement
	Arithmetic Expressions
	AT END Condition
	BLANK Clause
	BLOCK CONTAINS Clause
	Character Set
	CLASS Clause
	Class Test
	CLOSE Statement
	COBOL Program Sheet
	Collating Sequence
	Continuation Indicator
	DATA DIVISION
	DATA RECORDS Clause
	DECLARATIVES
	DISPLAY Statement
	Editing Clause
	ENTER Statement
	ENVIRONMENT DIVISION
	Figurative Constant
	File Description Entry
	GO TO Statement
	IDENTIFICATION DIVISION
	INVALID KEY Condition
	JUSTIFIED Clause
	LABEL RECORDS Clause
	Literals
	MOVE Statement
	MULTIPLY Statement
	NOTE Statement
	OCCURS Clause

COMPILER	COBOL LANGUAGE ELEMENT
1410/40/60	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	POINT Clause
	Procedure Verbs
	READ Statement
	RECORD CONTAINS Clause
	Record Description Entry
	RECORDING MODE Clause
	REDEFINES Clause
	Relation Test
	Series Separator
	SIZE Clause
	SIZE ERROR Condition
	STOP Statement
	Test Conditions
	Types Of Data Items
	USE Sentence
	VALUE Clause
	VALUE OF Clause
	WRITE Statement
1410/7010	ACCEPT Statement
	ADD Statement
	Arithmetic Expressions
	BLANK Clause
	BLOCK CONTAINS Clause
	CALL Statement
	Character Set
	CLASS Clause
	Class Test
	CLOSE Statement
	COBOL Program Sheet

Table 1. Compiler Index to COBOL Differences

COMPILER	COBOL LANGUAGE ELEMENT
1410/7010	Collating Sequence
	Continuation Indicator
	DATA DIVISION
	DATA RECORDS Clause
	DISPLAY Statement
	ENTER Statement
	ENVIRONMENT DIVISION
	File Description Entry
	GO TO Statement
	Identification Code
	IDENTIFICATION DIVISION
	LABEL RECORDS Clause
	Literals
	MOVE Statement
	MULTIPLY Statement
	NOTE Statement
	OCCURS Clause
	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	POINT Clause
	Procedure Verbs
	Qualification
	READ Statement
	RECORD CONTAINS Clause
	Record Description Entry
	RECORDING MODE Clause
	REDEFINES Clause
	Relation Test
	Series Separator
	SIGNED Clause
	SIZE Clause
	STOP Statement

COMPILER	COBOL LANGUAGE ELEMENT
1410/7010	SUBTRACT Statement
	SYNCHRONIZED Clause
	Test Conditions
	Types Of Data Items
	USAGE Clause
	USE Sentence
	VALUE Clause
	VALUE OF Clause
	WRITE Statement
7040/7044	ACCEPT Statement
	ADD Statement
	Arithmetic Expressions
	AT END Condition
	BLANK Clause
	BLOCK CONTAINS Clause
	Character Set
	CLASS Clause
	Class Test
	CLOSE Statement
	COBOL Program Sheet
	Collating Sequence
	Continuation Indicator
	DATA DIVISION
	DATA RECORDS Clause
	DISPLAY Statement
	Editing Clause
	ENTER Statement
	ENVIRONMENT DIVISION
	File Description Entry
	GO TO Statement
	IDENTIFICATION DIVISION
	JUSTIFIED Clause

Table 1. Compiler Index to COBOL Differences

COMPILER	COBOL LANGUAGE ELEMENT
7040/7044	LABEL RECORDS Clause
	Literals
	MOVE Statement
	MULTIPLY Statement
	NOTE Statement
	OCCURS Clause
	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	POINT Clause
	PROCEDURE Verbs
	READ Statement
	RECORD CONTAINS Clause
	Record Description Entry
	RECORDING MODE Clause
	REDEFINES Clause
	Relation Test
	Series Separator
	SIGNED Clause
	SIZE Clause
	SIZE ERROR Condition
	STOP Statement
	SUBTRACT Statement
	SYNCHRONIZED Clause
	Test Conditions
	Types Of Data Items
	USAGE Clause
	USE Sentence
	VALUE Clause
	VALUE OF Clause
	WRITE Statement
705/7080	ACCEPT Statement

COMPILER	COBOL LANGUAGE ELEMENT
705/7080	ADD Statement
	Arithmetic Expressions
	AT END Condition
	BLANK Clause
	BLOCK CONTAINS Clause
	Character Set
	CLASS Clause
	Class Test
	CLOSE Statement
	COBOL Program Sheet
	Collating Sequence
	Continuation Indicator
	DATA DIVISION
	DATA RECORDS Clause
	DISPLAY Statement
	Editing Clause
	ENTER Statement
	ENVIRONMENT DIVISION
	Figurative Constant
	File Description Entry
	GO TO Statement
	IDENTIFICATION DIVISION
	LABEL RECORDS Clause
	Literals
	MOVE Statement
	MULTIPLY Statement
	NOTE Statement
	OCCURS Clause
	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	POINT Clause
	Procedure Verbs

Table 1. Compiler Index to COBOL Differences

COMPILER	COBOL LANGUAGE ELEMENT
705/7080	READ Statement
	RECORD CONTAINS Clause
	Record Description Entry
	RECORDING MODE Clause
	REDEFINES Clause
	Relation Test
	Series Separator
	SIGNED Clause
	SIZE Clause
	SIZE ERROR Condition
	STOP Statement
	SUBTRACT Statement
	SYNCHRONIZED Clause
	Test Conditions
	Types Of Data Items
	USAGE Clause
	VALUE Clause
	VALUE OF Clause
	WRITE Statement
7070-FOS	ACCEPT Statement
	ADD Statement
	Arithmetic Expressions
	BLANK Clause
	BLOCK CONTAINS Clause
	CALL Statement
	Character Set
	Class Test
	Collating Sequence
	DATA DIVISION
	DISPLAY Statement
	DIVIDE Statement
	ENTER Statement

COMPILER	COBOL LANGUAGE ELEMENT
7070-FOS	ENVIRONMENT DIVISION
	File Description Entry
	IDENTIFICATION Division
	JUSTIFIED Clause
	LABEL RECORDS Clause
	Literals
	MOVE Statement
	MULTIPLY Statement
	OCCURS Clause
	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	Procedure Verbs
	RECORD CONTAINS Clause
	Record Description Entry
	RECORDING MODE Clause
	REDEFINES Clause
	Relation Test
	STOP Statement
	SUBTRACT Statement
	SYNCHRONIZED Clause
	Test Conditions
	Types Of Data Items
	USAGE Clause
	USE Sentence
	VALUE Clause
	VALUE OF Clause
	WRITE Statement
7070/7074	ACCEPT Statement
	ADD Statement
	Arithmetic Expressions
	AT END Condition

Table 1. Compiler Index to COBOL Differences

COMPILER	COBOL LANGUAGE ELEMENT
7070/7074	BLOCK CONTAINS Clause
	Character Set
	CLASS Clause
	Class Test
	CLOSE Statement
	COBOL Program Sheet
	Collating Sequence
	Continuation Indicator
	DATA DIVISICN
	DATA RECORDS Clause
	DISPLAY Statement
	ENTER Statement
	ENVIRONMENT DIVISICN
	File Description Entry
	GO TO Statement
	IDENTIFICATION DIVISION
	LABEL RECORDS Clause
	Literals
	MOVE Statement
	MULTIPLY Statement
	NCTE Statement
	OCCURS Clause
	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	POINT Clause
	Procedure Verbs
	READ Statement
	RECORD CONTAINS Clause
	Record Description Entry
	RECORDING MCDE Clause
	REDEFINES Clause
	Relation Test

CCMPILER	COBOL LANGUAGE ELEMENT
7070/7074	Series Separator
	SIGNED Clause
	SIZE Clause
	STOP Statement
	Subscripting
	SYNCHRONIZED Clause
	Test Conditions
	Types Of Data Items
	USAGE Clause
	VALUE Clause
	VALUE OF Clause
	WRITE Statement
7090/7094	ACCEPT Statement
	ADD Statement
	Arithmetic Expressions
	AT END Condition
	BLANK Clause
	BLOCK CONTAINS Clause
	CALL Statement
	Character Set
	CLASS Clause
	Class Test
	CLOSE Statement
	COBOL Program Sheet
	Collating Sequence
	Continuation Indicator
	DATA DIVISION
	DATA RECCRDS Clause
	DISPLAY Statement
	ENTER Statement
	ENTRY Statement

Table 1. Compiler Index to COBOL Differences

COMPILER	COBOL LANGUAGE ELEMENT
7090/7094	ENVIRONMENT DIVISION
	Figurative Constant
	File Description Entry
	GO TO Statement
	IDENTIFICATION DIVISION
	LABEL RECORDS Clause
	MOVE Statement
	MULTIPLY Statement
	NOTE Statement
	OCCURS Clause
	OPEN Statement
	PERFORM Statement
	PICTURE Clause
	POINT Clause
	Procedure Verbs
	READ Statement
	RECORD CONTAINS Clause
	Record Description Entry

COMPILER	COBOL LANGUAGE ELEMENT
7090/7094	RECORDING MODE Clause
	REDEFINES Clause
	Relation Test
	RETURN Statement
	Series Separator
	SIGNED Clause
	SIZE Clause
	SIZE ERROR Condition
	STOP Statement
	SUBTRACT Statement
	SYNCHRONIZED Clause
	Test Conditions
	USAGE Clause
	USE Sentence
	VALUE Clause
	VALUE OF Clause
	WRITE Statement

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
ACCEPT Statement	ACCEPT data-name [FROM SYSIN1]	When the FROM option is not specified, the card reader is assumed. Data-name may not exceed 72 characters.	7090/7094
	ACCEPT data-name FROM { CARD-READER / 1401-CARD-READER / IN }	When the FROM option is not specified the card reader is assumed. Sucessive groups of 80 characters are read until data-name is filled. If data-name is not a multiple of 80 characters excess characters are lost.	7040/7044
	ACCEPT data-name		7070-FOS 7070/7074
		Data is read from the operating system standard input unit. If data-name specifies less than 80 characters in length, sufficient card images are read until the area is filled.	1410/7010
	ACCEPT data-name [FROM mnemonic-name]	The IBM 1402 Card Read-Punch is the standard input unit. If more than 80 positions are defined by data-name, multiple cards will be read until the area is filled.	1401/40/60
		The standard ACCEPT device is the card reader. Mnemonic-name, if specified, must be defined in the SPECIAL-NAMES paragraph.	705/7080
	ACCEPT data-name [FROM CONSOLE]	When FROM CONSOLE is specified, data-name may not exceed 72 character positions in length. When the FROM CONSOLE option is not written, one logical record is read from the system logical input device (SYSIN).	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
ADD Statement	ADD {data-name-1 / literal-1} [{data-name-2 / literal-2} ...] {TO / GIVING} data-name-n [ROUNDED] [SIZE ERROR condition]		7070-FOS
	ADD CORRESPONDING data-name 1 TO data-name-2 [ROUNDED] [SIZE ERROR condition]	Either TO or GIVING is required in an ADD statement. There must be no OCCURS clause governing any of the operands involved in an ADD CORRESPONDING statement. (The CORRESPONDING option is implemented only by the System/360 level 'F' compiler.) The SIZE ERROR option is discussed in turn in this table.	System/360
	ADD {data-name-1 / literal-1} [{data-name-2 / literal-2} ...] {TO / GIVING} data-name-n [ROUNDED] [SIZE ERROR condition]	The CORRESPONDING option is not implemented.	7070/7074
		The CORRESPONDING option is not implemented. The SIZE ERROR option is discussed in turn in this table.	1401/40/60
	ADD {data-name-1 / literal-1} [{data-name-2 / literal-2} ...] {TO / GIVING} data-name-n [ROUNDED] [SIZE ERROR condition]		1410/7010
	ADD CORRESPONDING data-name-1 TO data-name-2 [ROUNDED] [SIZE ERROR condition]	Computation on report items is permitted.	705/7080 7040/7044 7090/7094

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Alphabetic Item	level-number {FILLER / data-name} [REDEFINES clause] [OCCURS clause] PICTURE IS alpha-form [JUSTIFIED RIGHT] [USAGE IS DISPLAY] [VALUE IS alphabetic-literal]	Record description clauses which are not shown in this format specifically forbidden. Mandatory clauses are written without brackets.	System/360
Alphanumeric Item	level-number {FILLER / data-name} [REDEFINES clause] [OCCURS clause] PICTURE IS an-form [JUSTIFIED RIGHT] [USAGE IS DISPLAY] [VALUE IS non-numeric-literal	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clauses are written without brackets.	System/360
Arithmetic Expressions	Exponentiation Unary Operators	An expression having a succession of exponentiation operators separating a series of operands is evaluated as though parenthesization were inserted from the right. Thus, A**B**C is evaluated as though it were written (A** (B**C)). The minus sign is the only allowable unary operator (having only one operand). The floating point feature is required to handle non-integer exponents.	System/360
		When the sequence of execution is not specified by the parentheses, the order of execution of consecutive operations of the same hierarchical level is from left to right. Both the plus and minus signs are valid unary operators.	1401/40/60 705/7080 7070-FOS 7070/7074 7090/7094
		When the sequence of execution is not specified by parentheses, the order of execution of consecutive operations of the same hierarchical level is from left to right. Both the plus and minus signs are valid unary operators. Exponents in COMPUTE statements must be integral in value.	1410/7010

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
AT END Condition	AT END {imperative-sentence / NEXT SENTENCE}		7040/7044 7090/7094
	AT END {statement-1 / NEXT SENTENCE} {ELSE / OTHERWISE} {statement-2 / NEXT SENTENCE}		705/7080
	AT END {statement-1 / NEXT SENTENCE} [{ELSE / OTHERWISE} {statement-2 / NEXT SENTENCE}]	The implied AT END is permissible.	1401/40/60
	AT END imperative-statement...	AT END may not be implied.	System/360
		The implied AT END is permissible.	7070/7074
Binary Item	level-number {FILLER / data-name} [REDEFINES clause] [OCCURS clause] PICTURE IS numeric-form USAGE IS COMPUTATIONAL [VALUE IS numeric-literal] [SYNCHRONIZED clause]	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clauses are written without brackets.	System/360
BLANK Clause	BLANK WHEN ZERO	This clause may be used only in the description of a elementary alphanumeric report item.	7070-FOS
		This clause may be used only in the description of an elementary item.	1401/40/60 1410/7010 705/7080

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
BLANK Clause	BLANK WHEN ZERO	This clause may be used only in the description of an elementary item. In the absence of a CLASS clause, class is assumed to be alphanumeric.	7040/7044 7090/7094
		This clause may be used only in the description of elementary report or numeric-form items. Numeric-form items to which it is applied are considered to be report items. The BLANK clause may not be applied to an item having an * in its picture.	System/360
BLOCK CONTAINS Clause	BLOCK CONTAINS [integer-1 TO] integer-2 { RECORD / RECORDS / CHARACTER / CHARACTERS }	This clause is not permitted if the DATA RECORDS clause specifies more than one record-name and the records are of variable length. Block size is determined from the product of the maximum (or actual) record length and the maximum (or actual) number of records specified. Padding records are not expected on input nor are they generated on output.	7090/7094
		This clause is not permitted if the DATA RECORDS clause specifies more than one record-name and the records are of variable length. Block size is determined from the product of the maximum (or actual) record length and the maximum (or actual) number of records specified. Padding records are not expected on input nor are they generated on output. When the CHARACTER(S) form of the clause is used, integer-1 and integer-2 must be divisible by 6.	7040/7044

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
BLOCK CONTAINS Clause	BLOCK CONTAINS integer-1 { RECORD / RECORDS / CHARACTER / CHARACTERS }	The CHARACTER(S) option is applicable only when the file consists of form 3 records. This clause is required for form 3 records.	7070/7074
		The CHARACTER(S) option is required when the file consists of variable length records.	1401/40/60 1410/7010
	BLOCK CONTAINS integer-1 { RECORD / RECORDS / CHARACTERS }	Refer to the 705/7080 COBOL publication for a complete discussion of this clause.	705/7080
	BLOCK CONTAINS integer-1 { RECORD / RECORDS / CHARACTERS }	The CHARACTER(S) option is applicable only when the file consists of form 3 records. The clause is required for form 3 records.	7070-FOS
	BLOCK CONTAINS integer-1 RECORDS	This clause specifies the number of logical records of maximum length in a physical record.	System/360
CALL Statement	CALL { linkage-symbol / subprogram-name / system- symbol } [USING { file-name-1 / data-name-1 / literal-1 } ...] CALL CHAIN USING { data-name-1 / literal-1 }	The CALL statement specifies the subprogram(s) to be included in the object program. The USING option specifies required parameters for the subprogram. CALL CHAIN permits the CHAIN feature to be used with the operating system's linkage loader.	1410/7010
	CALL 'entry-name-1' USING { data-name-1... / file-name-1... } [{ file-name-n / data-name-m }] [data-name-m+1 ...] [data-name-m+n] [RETURNING procedure-name-1 ...].	CALL is used in a COBOL main program to transfer to a subprogram. The USING option specifies the parameters that are passed between the main program and the RETURNING option specifies the point in the main program to which control is returned.	7090/7094

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
CALL Statement	CALL entry-name [USING argument-1 ...].	This statement is used to effect transfer of control to a subprogram. Entry-name represents the name of the subprogram's entry point. Argument-1 may be a data-name (when calling a COBOL subprogram) or a data-name, file-name, or procedure-name (when calling a subprogram written in a language other than COBOL).	System/360
	CALL subprogram-1 USING {data-name-1 / literal-1} [{data-name-2 / literal-2} ...]		7070-FOS
Character Set	Character Set	In addition to the complete COBOL character set ('H') the programmer can use, as characters in non-numeric literals, any characters (except the quotation mark) included in the IBM extended BCD interchange character (EBCDIC) set.	System/360
		The IBM character set 'H' must be used for COBOL source programs. The computer character set may be used for non-numeric literals and for comments.	1401/40/60 1410/7010 7040/7044 7070-FOS 7070/7074 7090/7094
CLASS Clause	CLASS IS {ALPHABETIC / NUMERIC / ALPHANUMERIC / AN}	This clause is not implemented by System/360. A PICTURE clause should be used to express the functions of this clause.	1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
Class Test	data-name IS [NOT] {NUMERIC / ALPHABETIC}	The item to be tested can be elementary alphanumeric, alphabetic, internal decimal, or external decimal. The item is tested to determine whether it is solely alphabetic or solely numeric.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Class Test	IF data-name IS [NOT] {NUMERIC / ALPHABETIC}	Data-name must be defined in the Data Division as an elementary non-report item of the alphanumeric class. The item is tested to determine whether it is solely alphabetic or solely numeric.	7070-FOS
		Data-name must be defined in the Data Division as being of the alphanumeric class. The item is tested to determine whether it is solely alphabetic or solely numeric.	1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
CLOSE Statement	CLOSE file-name-1 [WITH {LOCK / NO REWIND}] [file-name-2...]	The 1401/40/60 compiler does not provide the ability to process a given file as both an input file and an output file in the same program, unless it is a direct-access file.	1401/40/60
	CLOSE file-name-1 [UNIT] [WITH {LOCK / NO REWIND}] [file-name-2...]	The reserved word UNIT replaces the current systems equivalent, REEL. If the UNIT option is specified, the IOCS volume switching procedures are instituted. The LOCK option causes the current volume of the file to be removed.	System/360
	CLOSE file-name-1 [REEL] [WITH {LOCK / NO REWIND}] [file-name-2...]	The REEL option is used when a reel of a file is to be closed prior to its normal end.	1410/7010 7040/7044 7070/7074 7090/7094
	CLOSE file-name-1 [WITH LOCK] [file-name-2 ...]		705/7080

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
COBOL Program Sheet	Margin Restrictions	For System/360 COBOL, margin A (columns 8 through 11) is restricted to the following entries... Division Names Section Names Paragraph Names Level Indicators. No other entries may begin before column 12.	1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
Collating Sequence	Relation Test	Comparison of non-numeric items is with respect to the binary collating sequence of characters in the IBM extended BCD interchange code.	System/360
		Each computer has its own character set, with a specified order of precedence.	1401/40/60 1410/7010 7040/7044 705/7080 7070-FOS 7070/7074 7090/7094
Conditional Statement	IF condition-1 [THEN] {statement-1... / NEXT SENTENCE} [{ELSE / OTHERWISE}{statement-2... / NEXT SENTENCE}]	For System/360 level E statement-1 must be imperative (no true path nesting is permitted).	System/360
Continuation Indicator	Continuation Indicator	A hyphen must be punched in column 7 of any line in which a word or literal is being continued.	1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
		If compatibility with current systems compilers is desired, a hyphen must be punched in column 7 of each line in which a non-numeric literal is being continued. Continuation of words or numeric literals is not permitted.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Data Division	DATA DIVISION. FILE SECTION. File Description Entries. Record Description Entries. WORKING-STORAGE SECTION. Record Description Entries.		7070-FOS
	DATA DIVISION. FILE SECTION. File Description Entries. Record Description Entries WORKING-STORAGE SECTION. Record Description Entries. CONSTANT SECTION. Record Description Entries.	System/360 does not implement a Constant Section.	1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
	DATA DIVISION. FILE SECTION. File Description Entries. Record Description Entries. Sort Description Entries. Record Description Entries. Saved Area Entries. Record Description Entries. WORKING-STORAGE SECTION. Record Description Entries. LINKAGE SECTION. Record Description Entries. REPORT SECTION. Record Description Entries. Report Group Description Entries. Report Description Entries.	The division header is required, whether or not any sections are present within the division. The Sort, Saved Area, and Report elements of the language are for the level F compiler only.	System/360
DATA RECORDS Clause	DATA {RECORD / RECORDS} {IS / ARE} data-name-1 [data-name-2...]	This clause is required in every file description entry. More than one data-name implies redefintion. Multiple records in a file may be of different sizes and formats.	1401/40/60 1401/7010 7040/7044 705/7080 7070/7074 7090/7094
	DATA {RECORD IS / RECORDS ARE} record-name...	The clause is never required.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Declaratives	PROCEDURE DIVISION. DECLARATIVES. {section-name SECTION. USE sentence. {[paragraph-name.] sentence... .}...}... END DECLARATIVES.	The USE sentence identifies the type of declarative. There are four options of the USE sentence. Each identifies one of the following types of procedures... 1. Label-checking procedures 2. Label-writing procedures 3. Asynchronous processing procedures 4. Report-writing procedures	System/360
	PROCEDURE DIVISION. DECLARATIVES. section-name SECTION. USE FOR KEY-CONVERSION ON ALL FILES file-name [file-name...] imperative-statement(s).		1401/40/60
DISPLAY Statement	DISPLAY {data-name-1 / literal-1} [{data-name-2 / literal-2}...]		7040/7044 7070-FOS 7070/7074
	DISPLAY {literal-1 / figurative-constant-1 / data-name-1} [{literal-2 / figurative-constant 2 / data-name-2}...] [UPON SYSOU1]	When the UPON option is omitted, the standard display device, the on-line printer, is assumed. A line of display information is 72 characters long. If the UPON option is specified, the information displayed is in strings of 180 characters.	7090/7094
	DISPLAY {data-name-1 / literal-1} [{data-name-2 / literal-2}...] [UPON mnemonic-name]	If UPON is omitted, the 1403 printer is assumed as the standard display device. If UPON is omitted, the console typewriter is assumed as the standard display device.	1401/40/60 705/7080

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
DISPLAY Statement	DISPLAY {data-name-1 / literal-1} ... [UPON mnemonic-name]	If the UPON option is omitted, the console printer is used as the standard display device. There is no restriction on the length of the data to be displayed.	1410/7010
	DISPLAY {data-name-1 / literal-1} ... [UPON {CONSOLE / SYSPCH}]	When the UPON option is omitted, the system logical output device (SYSOUT) is assumed. When SYSPCH is specified, the system logical punch device is assumed. If the UPON option is specified, the sum of the sizes of the operands may not exceed 72 character positions. When UPON is omitted, the sum of the operands may not exceed the maximum logical record length for the system logical output device (SYSOUT).	System/360
DIVIDE Statement	DIVIDE {data-name-1 / literal-1} INTO {data-name-2 / literal-2} GIVING data-name-3 [ROUNDED] [SIZE ERROR condition]	GIVING is always required in a DIVIDE statement.	7070-FOS
Editing Clause	{ZERO SUPPRESS / CHECK PROTECT / FLOAT DOLLAR SIGN} [LEAVING integer-1 {PLACE / PLACES}]	This clause is not implemented by System/360. A PICTURE clause should be used to express the functions of this clause.	1401/40/60 705/7080
ENTER Statement	ENTER LINKAGE-MODE. {CALL statement. / ENTRY statement. / RETURN statement.} ENTER COBOL.	The ENTER statement provides for communication between a COBOL program and a subprogram that may be written in COBOL, FORTRAN IV, or MAP.	7090/7094
	ENTER {ASSEMBLY-PROGRAM / COBOL}.		7040/7044

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
ENTER Statement	ENTER {AUTOCODER / COBOL}.		1401/40/60 705/7080 7070/7074
	ENTER {COMMUNICATION-MODE / COBOL}.		7070-FOS
		When used in conjunction with the CALL statement, an ENTER statement allows the construction of multi-phase COBOL programs and the incorporation of FORTRAN and/or autocoder subprograms in a COBOL program.	1410/7010
	ENTER LINKAGE. {CALL statement. / ENTRY statement. / RETURN statement.} ENTER COBOL.	The ENTER statement, used in conjunction with ENTRY or statements, permits communication between a COBOL object program and one or more subprograms or other language subprograms.	System/360
ENTRY Statement	ENTRY entry-name [USING data-name...].	This statement is used to establish an entry point in a COBOL subprogram. Control is transferred to the entry point by a CALL statement in another program. Entry-name defines the entry point. Data-name must be defined in the LINKAGE SECTION.	System/360
	ENTRY POINT IS 'entry-name' [RECEIVE [data-name-1...] data-name-m] [PROVIDE [data-name-m+1...] data-name-m+n].	ENTRY POINT identifies the place in the subprogram to which control is transferred. The RECEIVE option specifies the data items that receive data from the main program. The PROVIDE option specifies the data items that return data to the main program.	7090/7094
Environment Division	ENVIRONMENT DIVISION. [CONFIGURATION SECTION.] [SOURCE-COMPUTER paragraph.] [OBJECT-COMPUTER paragraph.] [INPUT-OUTPUT SECTION.] [FILE-CONTROL.] SELECT file-name ASSIGN clause [RESERVE clause] [ACCESS clause] [ORGANIZATION clause] [SYMBOLIC KEY clause] [ACTUAL KEY clause]. [I-O-CONTROL.] [SAME clause.] [RERUN clause.] [APPLY clause.]	The System/360 concept of asynchronous processing, as well as its concept of device-independence, has placed extended interpretations on the language of the Environment Division. The Environment Division should be rewritten entirely whenever a program is to be processed for a different computer.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Environment Division	ENVIRONMENT DIVISION. CONFIGURATION SECTION. SOURCE-COMPUTER paragraph. OBJECT-COMPUTER paragraph. [SPECIAL-NAMES paragraph.] INPUT-OUTPUT SECTION. FILE-CONTROL. SELECT clause ASSIGN clause [RESERVE clause] [I-O-CONTROL.]	The nature of the Environment Division is such that it must be rewritten entirely whenever the program is to be processed for a different computer.	7070-FOS
	ENVIRONMENT DIVISION. CONFIGURATION SECTION. SOURCE-COMPUTER paragraph. OBJECT-COMPUTER paragraph. [SPECIAL-NAMES paragraph.] [INPUT-OUTPUT SECTION.] [FILE-CONTROL.] SELECT clause [RENAMING clause] ASSIGN clause. [I-O-CONTROL. {APPLY clause / RERUN clause} .]	System/360 does not implement the RENAMING clause. The nature of the Environment Division is such that it must be rewritten entirely whenever the program is to be processed for a different computer.	7090/7094
	ENVIRONMENT DIVISION. CONFIGURATION SECTION. SOURCE-COMPUTER paragraph. OBJECT-COMPUTER paragraph. [SPECIAL-NAMES paragraph.] [INPUT-OUTPUT SECTION.] [FILE-CONTROL.] SELECT clause [RENAMING clause] ASSIGN CLAUSE [RESERVE clause]. [I-O-CONTROL. {APPLY clause / RERUN clause} .]	System/360 does not implement the RENAMING clause. The nature of the Environment Division is such that it must be rewritten entirely whenever the program is to be processed for a different computer.	7040/7044
	ENVIRONMENT DIVISION. CONFIGURATION SECTION. SOURCE-COMPUTER paragraph. OBJECT-COMPUTER paragraph. [SPECIAL-NAMES paragraph.] [INPUT-OUTPUT SECTION.] [FILE-CONTROL.] [RENAMING clause] ASSIGN clause [RESERVE clause] [I-O-CONTROL.] APPLY clause [RERUN clause]	System/360 COBOL does not implement the RENAMING clause. The nature of the Environment Division is such that it must be rewritten entirely whenever the program is to be processed for a different computer.	1410/7010 705/7080 7070/7074

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Environment Division	ENVIRONMENT DIVISION. CONFIGURATION SECTION. SOURCE-COMPUTER paragraph. OBJECT-COMPUTER paragraph. [SPECIAL-NAMES paragraph.] [INPUT-OUTPUT SECTION.] [FILE-CONTROL.] SELECT clause ASSIGN clause [RESERVE clause]	The nature of the Environment Division is such that it must be rewritten entirely whenever the program is to be processed for a different computer.	1401/40/60
Extended Floating Point Item	level-number {FILLER / data-name} [REDEFINES clause] [OCCURS clause] PICTURE IS fp-form [USAGE IS DISPLAY]	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clauses are written without brackets.	System/360
External Decimal Item	level-number {FILLER / data-name} [REDEFINES clause] [OCCURS clause] [VALUE IS numeric-literal] PICTURE IS numeric-form	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clauses are written without brackets.	System/360
Figurative Constant	ALL 'literal'	System/360 requires that literal be a single character bounded by quotation marks.	1401/40/60 705/7080
		System/360 requires that literal be a single character bounded by quotation marks. The use of a figurative constant in place of literal, as in ALL SPACES is not permitted in System/360.	7090/7094
File Description Entry	{FD / SD / SA} file-name COPY library-name [FROM LIBRARY]. FD file-name VALUE OF clause [BLOCK CONTAINS clause] [RECORD CONTAINS clause] [LABEL RECORDS clause] [DATA RECORDS clause] [REPORT clause]	An SD entry is used to describe each sort-file used in the program. (Level-F only.) An SA entry is used to define a saved-area, and is allowed only when used in conjunction with a USE FOR RANDOM PROCESSING declarative in the Procedure Division. (Level-F only.)	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
File Description Entry	SD sort-file-name VALUE OF clause [RECORD CONTAINS clause] [DATA RECORDS clause] SA saved-area-name [RECORD CONTAINS clause]	File-name, sort-file-name, and saved-area-name are highest level qualifiers for their respective record description entries. Each saved-area-name defined by an SA entry must appear in option 4 APPLY clause.	System/360 (cont.)
	{FD / MD} file-name [RECORDING MODE clause] [BLOCK CONTAINS clause] [RECORD CONTAINS clause] [LABEL RECORDS clause] [VALUE OF clause] DATA RECORDS clause		1401/40/60
	FD file-name [RECORDING MODE clause] BLOCK CONTAINS clause RECORD CONTAINS clause LABEL RECORDS clause [VALUE OF clause]		7070-FOS
	FD file-name [RECORDING MODE clause] [BLOCK CONTAINS clause] RECORD CONTAINS clause LABEL RECORDS clause [VALUE OF clause] DATA RECORDS clause		7040/7044 7090/7094
		A file description entry with no associated record description entry is ignored by the compiler.	1410/7010
	FD file-name [RECORDING MODE clause] [BLOCK CONTAINS clause] [RECORD CONTAINS clause] LABEL RECORDS clause [VALUE OF clause] DATA RECORDS clause		705/7080 7070/7074

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
GO TO Statement	GO TO [procedure-name] GO TO procedure-name-1 procedure-name-2 [procedure-name-3...] DEPENDING ON data-name		1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
		Data-name must be an elementary integral numeric item whose length does not exceed four digits.	System/360
Group Item	level-number {FILLER / data-name} [REDEFINES clause] [OCCURS clause] [USAGE clause]	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clauses are written without brackets.	System/360
Identification Code	Columns 73-80	Any character from the COBOL character set may be used. The program identification code has no effect on the object program or the compiler.	System/360
		Columns 73-80 of the PROGRAM-ID entry must begin with an alphabetic character, cannot begin with the characters 'IB', and cannot contain either the slash or any blank characters. The identification code of the PROGRAM-ID entry will be used as the title of the object program.	1410/7010
Identification Division	{ID DIVISION. / IDENTIFICATION DIVISION.} [PROGRAM-ID. program-name.] {AUTHOR / AUTHORS}. author-name.	The division header is required. The PROGRAM-ID paragraph is optional.	7070-FOS
	{ID DIVISION. / IDENTIFICATION DIVISION.} PROGRAM-ID. program-name. {AUTHOR / AUTHORS}. author-name.	The division may be omitted. The first six characters of the program-name will be used as the deck-name where required by the operating system loader.	7090/7094

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Identification Division	IDENTIFICATION DIVISION. PROGRAM-ID. program-name. {AUTHOR / AUTHORS}. author-name.	The division header is required. The ident field (columns 72-80) of the PROGRAM-ID entry (1) Must begin with an alphabetic character. (2) Cannot begin with characters 'IB'. (3) Cannot contain slash or blank characters. (4) Will be modified by compiler, in the event of erroneous characters by substituting the character 'A'. (5) Will be used as the title of the object program.	1410/7010
		The division header is required. The first five characters of program-name will be passed along as program identification to the Autocoder processor.	7070/7074
		The division header is required. The first six characters of the program-name will be used as the deck-name where required by the operating system loader.	7040/7044
		The division header is required. Program-name must be either a name or literal.	1401/40/60 705/7080
	IDENTIFICATION DIVISION. PROGRAM-ID. program-name. [AUTHOR. author-name.]	The division header is required. Any characters in the EBCDIC character set may be used. Program-name must consist of from one to eight letters and/or digits, first character must be alphabetic. The program-name identifies the object program to the control program.	System/360
Internal Floating Point Item	level-number {data-name / FILLER} [REDEFINES clause] [OCCURS clause] USAGE IS {COMPUTATIONAL-1 / COMPUTATIONAL-2} [VALUE IS floating-point-literal] [SYNCHRONIZED clause]	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clause are written without brackets. The USAGE clause may be implied from the group level description.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Internal Decimal Item	level-number {data-name / FILLER} [REDEFINES clause] [OCCURS clause] PICTURE IS numeric-form USAGE IS COMPUTATIONAL-3 [VALUE IS numeric-literal]	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clauses are written without brackets.	System/360
INVALID KEY Condition	INVALID KEY {statement-1 / NEXT SENTENCE} [{ELSE / OTHERWISE} {statement-2... / NEXT SENTENCE}]		1401/40/60
	INVALID KEY imperative-statement...		System/360
JUSTIFIED Clause	JUSTIFIED RIGHT	The JUSTIFIED clause can only be used with alphabetic or alphanumeric items. If the sending field has an assumed decimal point, it is ignored during the move.	7070-FOS
		This clause can only be used with elementary alphabetic or alphanumeric items.	System/360
	JUSTIFIED {LEFT / RIGHT}	This clause cannot be used if the item is numeric and an assumed decimal point has been specified.	1401/40/60
LABEL RECORDS Clause	LABEL RECORDS ARE {OMITTED / STANDARD [data-name-1] / NON-STANDARD [data-name-2]}		7070-FOS

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
LABEL RECORDS Clause	LABEL RECORDS ARE { OMITTED / data-name }	For other than unit-record devices, the OMITTED option implies either (1) That there are no user labels on input, or (2) That input user labels are to be bypassed, or (3) That user labels are not created on output. Use of the data-name option means that user labels can be processed in the declaratives section. Data-name must be described in the linkage section.	System/360
	LABEL { RECORD / RECORDS } { IS / ARE } { OMITTED / STANDARD / NON-STANDARD }		7070/7074
	LABEL { RECORD / RECORDS } { IS / ARE } { { STANDARD / NON-STANDARD } [WITH integer-1 CHARACTERS] [BEGINNING-LABEL] [ENDING-LABEL] / OMITTED }	This clause is required in every file description entry. OMITTED must not be specified when an input file contains tape labels.	1410/7010
	LABEL { RECORD / RECORDS } { IS / ARE } { OMITTED / STANDARD [BEGINNING-TAPE-LABEL] [NO-TAPEMARK] [ENDING-TAPE-LABEL] }		705/7080
	LABEL { RECORD / RECORDS } { IS / ARE } { STANDARD / OMITTED }		1401/40/60 7040/7044 7090/7094
Literals	Numeric Literal Non-numeric Literal Figurative Constant	For 7070-FOS, (1) HIGH-VALUE(S) and LOW-VALUE(S) are not implemented. (2) ZERO and its' plural forms are not permitted in arithmetic statements.	1401/40/60 1410/7010 7040/7044 705/7080 7070-FOS 7070/7074

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Literals	Numeric Literal Non-numeric Literal Floating-point Literal Figurative Constant		System/360
MOVE Statement	MOVE {data-name-1 / literal-1} TO data-name-2 [data-name-3 ...]	Refer to the 705/7080 COBOL publication for a detailed discussion of valid MOVE operands.	705/7080
		Refer to 7070/7074 COBOL publication for a more detailed discussion of valid MOVE operands.	7070/7074
		The following moves are not valid System/360 moves: ZERO to report, alphabetic. ALL 'literal' to report, alphabetic. HIGH VALUES, LOW VALUES, QUOTES to report, alphabetic. report to report. group to report. alphabetic to report. alphanumeric to report.	1401/40/60
		The following moves are not valid System/360 moves: numeric to alphanumeric. ALL 'literal' to alphabetic.	7040/7044
	MOVE {literal-1 / data-name-1} TO data-name-2 [data-name-3...] MOVE CORRESPONDING data-name-1 TO data-name-2 [data-name-3 ...]	Refer to 7070-FOS COBOL publication for a detailed discussion of valid MOVE operands.	7070-FOS

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
MOVE Statement	MOVE {literal-1 / data-name-1} TO data-name-2 [data-name-3...] MOVE CORRESPONDING data-name-1 TO data-name-2 [data-name-3...]	The MOVE CORRESPONDING option is not implemented by System/360 level E. For a full discussion of valid and invalid System/360 move operands, consult the System/360 COBOL publication.	System/360
		The following moves are not valid System/360 moves: SPACES to numeric. ALL 'numeric-literal' to numeric. numeric to alphanumeric.	7090/7094
		The following moves are not valid System/360 moves: ZERO to report, alphabetic. ALL 'literal' to report, alphabetic. HIGH VALUES, LOW VALUES, QUOTES to report, alphabetic. report to report. group to report. alphabetic to report. alphanumeric to report.	1410/7010
MULTIPLY Statement	MULTIPLY {data-name-1 / literal-1} BY {data-name-2 [GIVING data-name-3] / literal-2 GIVING data-name-3} [ROUNDED] [SIZE ERROR condition]	System/360 permits a floating point literal for literal-1 and literal-2.	1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
	MULTIPLY {data-name-1 / literal-1} BY {data-name-2 / literal-2} GIVING data-name-3 [ROUNDED] [SIZE ERROR condition]	GIVING is required in a 7070-FOS MULTIPLY statement.	7070-FOS
NOTE Statement	NOTE any-statement	System/360 requires that the verb NOTE be the first word of any sentence in which it appears.	1401/40/60 1410/7010 7040/7044 7070/7074 7090/7094

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
OCCURS Clause	OCCURS integer-1 {TIME / TIMES} [DEPENDING ON data-name]	The OCCURS clause must not be used in a record description entry having a level number of 01. The value of data-name must be a positive non-zero integer. If data-name appears within the record description, it must precede the variable portion of the record. In this case integer-1 is considered to be the maximum number of occurrences and is used to determine storage reservation. Data-name should be qualified when necessary, but subscripting is not permitted. to be the maximum number of occurrences and is used to determine storage reservation. Data-name should be qualified when necessary, but subscripting is not permitted.	7090/7094
	OCCURS integer-1 {TIME / TIMES}	The OCCURS clause must not be used in a record description entry having a level number of 77. Integer-1 must be a positive numeric literal having an integral value greater than zero.	1401/40/60 1410/7010 705/7080
		The OCCURS clause must not be used in a record description entry having a level number of 77. Integer-1 must be a positive numeric literal having an integral value greater than zero. Items to be subscripted and that do not occupy full words are packed, unless synchronized.	7070/7074
	OCCURS integer-1 TIMES	In addition to the specifications applicable to the 7070/7074 compiler the 7070 COBOL/FORTRAN Operating System compiler requires that elements of an array must be an integral number of machine words apart. The description of such data and the placement of the actual data in storage must be made with this limitation in mind.	7070-FOS

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
OCCURS Clause	OCCURS integer-1 TIMES [DEPENDING ON data-name]	System/360 does not permit the use of an OCCURS clause in a level-88 entry. The level E compiler specifies that only one OCCURS DEPENDING clause per logical record may appear in the description of either a group that contains the last elementary item of the record, or in the description of the last elementary item itself. An OCCURS DEPENDING clause may not be subordinate to an item containing an OCCURS clause.	System/360
		The OCCURS clause must not be used in a record description entry having a level number of 77. The value of data-name must be a positive non-zero integer. If data-name appears within the record description, it must precede the variable portion of the record. In this case integer-1 is considered to be the maximum number of occurrences and is used to determine storage reservation. Data-name should be qualified when necessary, but subscripting is not permitted.	7040/7044
OPEN Statement	OPEN [INPUT file-name-1 ...] [OUTPUT file-name-2] [{INPUT-OUTPUT / I-O} file-name-3 ...]		1401/40/60

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
OPEN Statement	OPEN {INPUT file-name [WITH NO REWIND] [REVERSED] OUTPUT file-name [WITH NO REWIND] I-O file-name} [[INPUT] file-name [WITH NO REWIND] [REVERSED] [OUTPUT] file-name [WITH NO REWIND] [I-O] file-name] ...	An OPEN statement initiates execution of label-handling procedures specified by a USE declarative. The I-O option permits the opening of a direct-access file for both input and output operations.	System/360
	OPEN {INPUT file-name-1 [file-name-2...] OUTPUT file-name-3 [file-name-4...] INPUT file-name-5 [file-name-6...] OUTPUT file-name-7 [file-name-8...]}		1410/7010 7040/7044 705/7080 7070/7074 7090/7094
	OPEN [INPUT file-name-1 [WITH NC REWIND] [file-name-2 ...]] [OUTPUT file-name-3 [WITH NC REWIND] [file-name-4 ...]]		7070-FOS
PERFORM Statement	PERFORM procedure-name-1 [THRU procedure-name-2] {integer / data-name} TIMES	The use of THROUGH in lieu of THRU is permitted.	7070-FOS
	PERFORM procedure-name-1 [THRU procedure-name-2] {integer / data-name} TIMES	TIME (singular) is not permitted. THROUGH may not be used for THRU. Data-name must have an integral value and data-name or integer must have a positive value, less than 32,768.	System/360
		The use of TIME and THROUGH in lieu of TIMES and THRU, respectively is permitted.	1401/40/60 1401/7010 7040/7044 705/7080 7070/7074 7090/7094

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
PICTURE Clause	PICTURE IS { alphabetic-form numeric-form alphanumeric-form report-form floating-point-form }	The picture symbol 'X' is the only valid symbol for an alphanumeric item. Report-form is not a subdivision of the alphanumeric class. The System/360 COBOL publication should be consulted for a complete discussion of both the report-form and floating-point-form options of the System/360 PICTURE clause.	System/360
	PICTURE IS { alphabetic-form numeric-form alphanumeric-non-report alphanumeric-report }	For report items, maximum number of characters can be represented by a picture is 99. A picture symbol 'V' to the right or left of 'P' is redundant. Picture symbols Z, *, +, and - may appear to the right of a decimal point in a picture only if all numeric character positions are represented by the same symbol. The single character pictures 'J' or 'K' represent record mark and group mark, respectively.	1410/7010
		If the item is computational, the 'S' picture character is redundant. If the item is display, the 'S' picture character specifies a sign overpunch in the units position. 'P' is a valid report item character. The single character picture 'J' specifies a record mark, for peripheral equipment.	7040/7044
		The picture character 'S' should not be used. When the object computer is equipped with the expanded print-edit feature, the following special editing functions are available: High-order CR or minus signs. High-order DB or plus signs. Floating plus or minus signs. Floating dollar signs. Check protection (asterisk fill). Decimal suppression for blank or zero fields. Editing of a single-digit field cannot be specified.	1401/40/60

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
PICTURE Clause	PICTURE IS { alphabetic-form numeric-form alphanumeric-non-report alphanumeric-report }	The single character picture 'J' represents a record mark.	7070-FOS 7070/7074
		The single character pictures 'J' and 'K' represent a record mark and group mark, respectively. If the picture of an item specifies an implied decimal point, (symbol 'V'), it cannot contain more than 99 positions on either side of the implied decimal point. Numeric items in arithmetic expressions are limited to 18 numeric characters. Report items are limited to 18 numeric characters. Z and * cannot appear in the picture. They may not appear in the same picture with a floating dollar sign. They may appear to the right of an assumed or actual decimal point only if no numerics appear.	705/7080
	PICTURE IS { alphabetic-form numeric-form alphanumeric-non-report alphanumeric-report scientific-decimal }	If an item is computational, the 'S' picture character is redundant. If the item is display, the 'S' picture character specifies a sign overpunch in the units position. 'P' is a valid report item character. The single character picture 'J' specifies a record mark, for peripheral equipment.	7090/7094
POINT Clause	POINT LOCATION IS { LEFT RIGHT }	This clause is not implemented by System/360. A PICTURE clause should be used to express the function of this clause.	1401/40/60 1410/7010 7040/7044 7070/7074 7090/7094

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX		ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
PROCEDURE Verbs	OPEN READ WRITE CLOSE DISPLAY USE MOVE EXAMINE	COMPUTE ADD SUBTRACT MULTIPLY DIVIDE STOP GO TO ALTER PERFORM ENTER EXIT NOTE		7040/7044
	OPEN READ WRITE CLOSE DISPLAY ACCEPT MOVE	COMPUTE ADD SUBTRACT MULTIPLY DIVIDE STOP GO TO ALTER PERFORM ENTER EXIT NOTE		7070/7074
	OPEN READ WRITE CLOSE DISPLAY ACCEPT MOVE EXAMINE	COMPUTE ADD SUBTRACT MULTIPLY DIVIDE STOP GO TO ALTER PERFORM ENTER EXIT NOTE		705/7080
	OPEN READ WRITE CLOSE DISPLAY ACCEPT SEEK MOVE EXAMINE	COMPUTE ADD SUBTRACT MULTIPLY DIVIDE STOP GO TO ALTER PERFORM ENTER EXIT NOTE		1401/40/60

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Procedure Verbs	OPEN READ WRITE CLOSE DISPLAY ACCEPT USE MOVE EXAMINE COMPUTE ADD SUBTRACT MULTIPLY DIVIDE STOP GO TO ALTER PERFORM ENTER EXIT NOTE		1410/7010 7070-FOS 7090/7094
	OPEN READ WRITE REWRITE CLOSE DISPLAY ACCEPT PROCESS HOLD MOVE EXAMINE TRANSFORM COMPUTE ADD SUBTRACT MULTIPLY DIVIDE STOP ALTER PERFORM GO TO ENTER EXIT NOTE USE	Omitted from this list are additional verbs that are used when the report writer feature, the Sort feature, the source program library facility and the COBOL debugging language. Where differences from current systems exist, the individual verbs are discussed in this table.	System/360
Punctuation	Semicolon	A semicolon may be used optionally to separate a series of statements or clauses.	System/360
Qualification	Qualifiers	A name plus all its qualifiers cannot exceed a total of 300 characters. If a name requiring qualification is not qualified, the compiler assumes reference to the first appearance of the name in the source program.	1410/7010
		An item may not have more than 49 qualifiers.	7040/7044
		For Level E, the length of a name plus its qualifiers is restricted to 300 bytes.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
READ Statement	READ file-name RECORD [INTO data-name] {AT END / INVALID KEY} imperative-statement	An implied AT END or INVALID KEY is not permitted. If the INTO option is specified, the current record becomes available in the input area as well as in the area specified by data-name.	System/360
	READ file-name RECORD [INTO area-name] AT END imperative-statement		1410/7010
		When the INTO option is specified, the record is read directly into area-name. The record currently in the record area associated with the input file area remains available.	705/7080
		An implied AT END is permitted.	7070/7074
	READ file-name RECORD [INTO data-name] AT END {any imperative statement / NEXT SENTENCE}		7040/7044 7090/7094
	READ file-name RECORD [INTO area-name] {[AT END imperative-statement] / [INVALID KEY imperative-statement]}	An implied AT END or INVALID KEY is permitted.	1401/40/60

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
RECORD CONTAINS Clause	RECORD CONTAINS [integer-1 TO] integer-2 {CHARACTER / CHARACTERS} [DEPENDING ON data-name-1]	The RECORD CONTAINS clause is required. It is used to specify the size and form of the records to be processed. If all the records of a file are of the exact same size, integer-2 only should be specified.	1410/7010
	RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS [WITHOUT COUNT CONTROL]	This clause is required if variable length records are described for the file. It is not necessary for a file having equal length records. If the WITHOUT COUNT CONTROL option is specified, the file is assumed to contain undefined-length or fixed length records and the BLOCK CONTAINS clause may not be specified for the file.	System/360
	RECORD CONTAINS [integer-1 TO] integer-2 {CHARACTER / CHARACTERS}		7040/7044 7070/7074 7090/7094
		The 'integer-1 TO' option is not permitted for unit-record files.	1401/40/60
	RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS		705/7080 7070-FOS
Record Description Entry	{77 / 01} data-name COPY library-name [FROM LIBRARY] level-number {FILLER / data-name} [REDEFINES Clause] [USAGE Clause] [OCCURS Clause] [VALUE Clause] [SYNCHRONIZED Clause] [BLANK Clause] [PICTURE Clause] [JUSTIFIED Clause]	Data-name may be either a group item, or one of the following types of elementary items: Alphabetic Alphanumeric Report External decimal Internal decimal Binary External floating point Internal floating point Refer to these specific categories in the 'Language Element' column for information concerning their description.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Record Description Entry	level-number {FILLER / data-name} [REDEFINES Clause] [USAGE Clause] [OCCURS Clause] [VALUE Clause] [SYNCHRONIZED Clause] [BLANK Clause] [PICTURE Clause] [JUSTIFIED Clause]		7070-FOS
	level-number {FILLER / data-name} [REDEFINES Clause] [SIZE Clause] [OCCURS Clause] [VALUE Clause] [Editing Clause] [POINT Clause] [CLASS Clause] [PICTURE Clause] [JUSTIFIED Clause] [BLANK Clause]		1401/40/60
	level-number {FILLER / data-name} [REDEFINES Clause] [SIZE Clause] [OCCURS Clause] [VALUE Clause] [USAGE Clause] [POINT Clause] [CLASS Clause] [PICTURE Clause] [SIGNED Clause] [BLANK Clause] [SYNCHRONIZED Clause]	If the REDEFINES clause is not specified first, the compiler generates a warning message. However, the clause is acceptable. A record description entry with no associated file description entry is placed in Working-storage.	1410/7010
	level-number {FILLER / data-name} [REDEFINES Clause] [SIZE Clause] [OCCURS Clause] [VALUE Clause] [USAGE Clause] [POINT Clause] [CLASS Clause] [PICTURE Clause] [SIGNED Clause] [BLANK Clause] [SYNCHRONIZED Clause] [Editing Clause]		705/7080

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Record Description Entry	level-number {FILLER / data-name} [REDEFINES Clause] [SIZE Clause] [OCCURS Clause] [VALUE Clause] [USAGE Clause] [POINT Clause] [CLASS Clause] [PICTURE Clause] [SIGNED Clause] [BLANK Clause] [SYNCHRONIZED Clause]		7040/7044 7090/7094
	level-number {FILLER / data-name} [REDEFINES Clause] [SIZE Clause] [OCCURS Clause] [VALUE Clause] [USAGE Clause] [POINT Clause] [CLASS Clause] [PICTURE Clause] [SIGNED Clause] [SYNCHRONIZED Clause]		7070/7074
RECORDING MODE Clause	RECORDING MODE IS {BCD / BINARY} [{LOW / HIGH} DENSITY]	If this clause is omitted, the compiler assumes BCD, HIGH DENSITY. System/360 does not implement this clause.	7040/7044
	RECORDING MODE IS {BCD / BINARY} [{LOW / HIGH} DENSITY] [WITHOUT COUNT CONTROL]	If this clause is omitted, the compiler assumes BCD, HIGH DENSITY. System/360 does not implement this clause.	7090/7094
	RECORDING MODE IS {LOAD / MOVE} mode {EVEN / ODD} PARITY	If this clause is omitted, the 1410 compiler assumes MOVE mode, EVEN parity. System/360 does not implement this clause.	1410/7010
	RECORDING MODE IS mode	Refer to the applicable COBOL publication for a detailed discussion of the RECORDING MODE clause. System/360 does not implement this clause.	705/7080 7070-FCS 7070/7074

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
RECORDING MODE Clause	RECORDING MODE IS 1	If this clause is omitted, the processor assumes recording mode 1, which specifies MOVE mode, even parity. System/360 does not implement this clause.	1401/40/60
REDEFINES Clause	data-name-1 REDEFINES data-name-2	REDEFINES clauses may be specified for items subordinate to items containing REDEFINES clauses. Data-name-1 may not contain nor subordinate an OCCURS DEPENDING clause, nor be subordinate to an item containing an OCCURS clause. Data-name-2 may not be subordinate to nor contain an OCCURS clause, nor subordinate an item containing an OCCURS DEPENDING clause. No lower level-number entries may intervene between data-name-1 and data-name-2. Data-name-1 must not be written with qualifiers. Length of data-name-1 times occurrences must be not greater than length of data-name-2.	System/360
		Data-name-2 must not be subscripted. The sizes of data-name-1 and data-name-2 must be equal.	1410/7010
		If data-name-2 is not unique, it must be qualified. Data-name-2 must not be subscripted. If data-name-1 is a group item with a level-number other than 01, the REDEFINES clause must not be used in the description of any item within that group (no nested redefinition).	1401/40/60 7070/7074 7090/7094
		If data-name-2 is not unique, it must be qualified. Data-name-2 must not be subscripted. If data-name-1 is a group item, the REDEFINES clause must not be used in the description of any item within that group (no nesting redefinition).	7070-FOS

TABLE 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
REDEFINES Clause	data-name-1 REDEFINES data-name-2	If data-name-2 is not unique, it must be qualified. Data-name-2 must not be subscripted. When the level-number of data-name-2 is other than 01, it must specify a storage area of the same size as data-name-1.	7040/7044
		Multiple redefinitions are possible, provided that they occur at the same level and without other entries intervening at that level. An entry containing the REDEFINES clause must not be written at a lower level within another entry containing the REDEFINES clause, unless the latter entry is at the 01 level, in which case one such lower level redefinition is permitted at a time.	705/7080
Relation Test	{data-name / literal / arithmetic expression / figurative-constant} IS [NOT] {< / > / = / GREATER THAN / LESS THAN / EQUAL TO} {data-name / literal / arithmetic expression / figurative-constant}	Implied subjects and implied operators are not implemented on level E.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Relation Test	{data-name / literal / arithmetic expression / figurative-constant} IS [NOT] {GREATER THAN / LESS THAN / EQUAL TO / =} {data-name / literal / arithmetic expression / figurative-constant}	Implied subjects or operators are not permitted.	7070-FOS
	{data-name / figurative constant / arithmetic expression / literal} {IS [NOT] GREATER THAN / IS [NOT] LESS THAN / IS [NOT] EQUAL TO} {data-name / figurative constant / literal / arithmetic expression}	Implied subjects and implied operators are permitted.	1401/40/60 1410/7010 7070/7074
	IF {data-name / literal / figurative constant / arithmetic expression} IS [NOT] {GREATER THAN / LESS THAN / = / EQUAL TO} {data-name / literal / figurative constant / arithmetic expression}	Implied subjects and implied operators are permitted.	7040/7044 7090/7094
Report Item	level-number {FILLER / data-name} [REDEFINES Clause] [OCCURS Clause] PICTURE IS {numeric-form BLANK WHEN ZERO / report-form [BLANK WHEN ZERO]} [USAGE IS DISPLAY]	Record description clauses which are not shown in this format are specifically forbidden. Mandatory clauses are written without brackets.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Reserved System/360 Only	CHANGED INDEXED REWRITE CHECKING LABELS SA CONSOLE LINKAGE SEARCH CREATING MORE-LABELS SYSPCH CYCLES NAMED TRACE DISPLAY-ST ORGANIZATION TRACKS ENTRY PRINT-SWITCH TRANSFORM EXHIBIT PROCESS UNIT-RECORD FILE-ID READY UNITS HOLD RELATIVE UTILITY IBM-360 RESTRICTED WITHOUT COMPUTATIONAL-3 DIRECT-ACCESS FORM-OVERFLOW	These words are not reserved in any of the current systems vocabularies.	System/360
RETURN Statement	RETURN [VIA entry-name-1] .	The return statement enables restoration of the necessary registers saved at an entry point.	System/360
	RETURN VIA 'entry-name' [DEPENDING ON data-name]	RETURN is used to identify the end of the Section of the subprogram to be executed and provides for return to the main program. The DEPENDING ON option can be used with the RETURNING option in the CALL statement to specify an alternate return.	7090/7094
Series Separation	AND , ,AND	A series of operands may be separated by the comma, the word AND or the comma followed by the word AND.	1401/40/60 1410/7010 7040/7044 705/7080 7070/7074 7090/7094
		A comma may be used optionally as a separator between successive operands of a statement, or to separate a series of clauses.	System/360
SIGNED Clause	SIGNED	This clause is not implemented by System/360. A PICTURE clause should be used to express the functions of this clause.	1410/7010 System/360 7040/7044 705/7080 7070/7074 7090/7094

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
SIZE Clause	SIZE IS [integer-1 TO] integer-2 [{CHARACTER / CHARACTERS / DIGIT / DIGITS}] [DEPENDING ON] data-name	This clause is not implemented by System/360. A PICTURE clause should be used to express the functions of this clause.	1401/40/60
	SIZE IS integer-1 {CHARACTER / CHARACTERS / DIGIT / DIGITS}		1410/7010 7040/7044 705/7080 7070/7074 7090/7094
SIZE ERROR Condition	ON SIZE ERROR {statement-1 / NEXT SENTENCE} [{OTHERWISE / ELSE} {statement-2 / NEXT SENTENCE}]	The ELSE or OTHERWISE option will cause a warning message to be issued. For compatibility with System/360, the option should not be used.	1401/40/60 705/7080
	ON SIZE ERROR {any imperative statement / NEXT SENTENCE}		7040/7044
	ON SIZE ERROR imperative-statement		System/360 7090/7094
STOP Statement	STOP {RUN / literal}		1401/40/60 7040/7044 705/7080 7070-FOS 7070/7074 7090/7094
		STOP RUN returns control to the operating system. STOP literal causes a compiler-generated message code, as well as the specified literal, to be displayed. The message code must be keyed in on the console in order to resume.	System/360
		STOP RUN returns control to the system monitor. STOP literal causes the literal to be displayed, after which the resident monitor wait-loop is entered.	1410/7010

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
Subscripting	Subscript Format	System/360 specifies that for multiple subscripting each subscript must be followed by an immediate comma and a blank. For example, data-name (A,B,C) is acceptable, but data-name (A,B,C) is not valid.	7070/7074
SUBTRACT Statement	SUBTRACT CORRESPONDING data-name-1 FROM data-name-2 [ROUNDED] [SIZE ERROR Condition]	SUBTRACT CORRESPONDING is not implemented by the 1401/40/60 and 7070/7074 compilers.	System/360
	SUBTRACT CORRESPONDING data-name-1 FROM data-name-2 [ROUNDED] [SIZE ERROR Condition]	System/360 level E does not implement SUBTRACT CORRESPONDING.	1401/7010 7040/7044 7090/7094 705/7080 7070-FOS
SYNCHRONIZED Clause	SYNCHRONIZED {LEFT / RIGHT}	A right synchronized computational item is most efficient for computation. It will appear in the least significant position of a word (2 words if more than ten digits) with its sign bit in the sign position of the word. Numeric computational items without a synchronized clause will occupy the minimum number of six bit blocks capable of holding it and its sign (in the leftmost bit). The item will occupy storage in six-bit bytes, although it is computational and therefore specified as being in binary form, when not synchronized.	7040/7044 7090/7094
		Refer to the applicable COBOL publication for a detailed discussion of the SYNCHRONIZED clause.	705/7080 7070-FOS 7070/7074

Table 2. COBOL DIFFERENCES.

COBOL EIEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
SYNCHRONIZED Clause	SYNCHRONIZED {LEFT / RIGHT}	The SYNCHRONIZED clause is never necessary in a System/360 COBOL program. If the clause is specified, it is treated as comments by the compiler. Storage is assigned so that the starting byte of a binary or internal floating-point item is on the next available half-word, full word, or double word boundary, as appropriate. If a data hierarchy contains binary or floating point items intermixed =slack bytes= (implicit synchronization) may be introduced. Slack bytes are inserted on output and expected on input files.	System/360
		This clause is recognized, but ignored by the 1410/7010 compiler.	1410/7010
Test Condition	Relation Test Sign Test Class Test Condition-name Test Form-overflow Test		System/360
	Relation Test Sign Test Class Test Condition-name Test Switch-status-name Test	The System/360 environment does not provide for testing switch-status.	1401/40/60 7040/7044 705/7080 7070-FOS 7070/7074 7090/7094
Types of Data Items	Group Items Elementary Item Alphabetic Item Numeric Item Alphabetic Item Report Item Non-report Item	Refer to these specific categories in the 'Language Element' column for information concerning the valid clauses which may be used in their description for System/360.	1401/40/60 1410/7010 7040/7044 705/7080 7070-FOS 7070/7074

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
USAGE Clause	USAGE IS { COMPUTATIONAL / COMPUTATIONAL-1 / COMPUTATIONAL-2 / DISPLAY }	DISPLAY specifies BCD storage. COMPUTATIONAL specifies binary storage. If a group item is specified COMPUTATIONAL, the elementary items in the group are treated as COMPUTATIONAL, but not the group item itself. COMPUTATIONAL internal decimal COMPUTATIONAL-1 single precision floating-point COMPUTATIONAL-2 double precision floating-point Note: The only clause required for a floating-point item is USAGE. COMPUTATIONAL, COMPUTATIONAL-1, and COMPUTATIONAL-2 items are stored in the least multiple of six bits capable of holding the item including its sign.	7090/7094
	USAGE IS { COMPUTATIONAL / DISPLAY }	COMPUTATIONAL corresponds to 7070 numeric mode, Display corresponds to 7070 alphanumeric mode. Computational and display fields cannot occur in the same machine word. Two or more computational fields in the same machine word must have the same sign. Usage specified at the group level supersedes subordinate specifications.	7070-FOS 7070/7074
		DISPLAY specifies BCD storage. COMPUTATIONAL specifies binary storage. If a group item is specified COMPUTATIONAL, the elementary items in the group are treated as computational, but not the group item itself. COMPUTATIONAL items are stored in the least multiple of six bits capable of holding the item including the sign.	7040/7044
		All data represented within the 705/7080 is in external format. Hence, the DISPLAY option is not meaningful and will be ignored. However, if an item is specified as being COMPUTATIONAL, it must be a numeric item.	705/7080

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
USAGE Clause	USAGE IS { COMPUTATIONAL / DISPLAY }	This clause does not affect internal representation, which is always BCD. If COMPUTATIONAL is specified, and the CLASS clause is omitted, the item is assumed to be numeric.	1410/7010
	USAGE IS { DISPLAY / COMPUTATIONAL / COMPUTATIONAL-1 / COMPUTATIONAL-2 / COMPUTATIONAL-3 }	DISPLAY specifies character form, one character per byte. COMPUTATIONAL specifies a binary item occupying 2, 4, or 8 byte positions corresponding to the specified decimal lengths of 1-4, 5-9, and 10-18 respectively. The leftmost bit of the reserved area is the operational sign. Computational items are aligned at the next half or full word boundaries, as appropriate. COMPUTATIONAL-1 short precision floating point COMPUTATIONAL-2 long-precision floating point COMPUTATIONAL-3 packed decimal	System/360
USE Sentence	USE FOR KEY-CONVERSION ON { ALL FILES / file-name [file-name...] } imperative-statement(s).	The USE declarative is used to specify the KEY CONVERSION procedure which is to be used for developing disk addresses.	1401/40/60
	USE FOR CREATING [BEGINNING / ENDING] LABELS CN OUTPUT [file-name...]. USE FOR CHECKING [BEGINNING / ENDING] LABELS ON INPUT [file-name...]. USE FOR RANDCM PROCESSING. USE BEFORE REPORTING data-name-1.	The first two options are used to provide user label processing procedures. BEGINNING refers to header labels, ENDING refers to trailer labels. Absence of both words indicates processing of both headers and trailers. The third option is used to specify out-of-line procedural statements for asynchronous processing. The fourth option is used to designate procedures to be executed by the REPORT WRITER facility.	System/360

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
USE Sentence	USE Sentence	Refer to the applicable COBOL publication for detailed discussion of the USE sentence.	7090/7094 7040/7044 1410/7010 7070-FOS
VALUE Clause	{VALUE IS / VALUES ARE} literal-1 [THRU literal-2] ...	If the VALUE clause specifies a numeric literal, a sign will be placed over the units position of the numeric field only if the literal is preceded by a plus or minus sign. The THRU option may be used only with condition-names.	1401/40/60
	VALUE IS literal	A VALUE clause can be used... For elementary items only. In File and Linkage Section, in level 88 entries only. The VALUE clause cannot be used: For group items. For report items. For external floating-point items. For items containing, or subordinate to an OCCURS clause or a REDEFINES clause (except for level 88 entries) .	System/360
		A numeric literal with preceding sign does not create an operational sign unless an 'S' appears in an associated PICTURE clause, or a SIGNED clause is used.	1410/7010
		If a PICTURE clause associated with the item contains an implied decimal, the VALUE clause literal must specify the actual decimal. If the VALUE clause is used at the group level, the numeric elementary items which are part of the group can consist only of unsigned, non-scaled integers. The VALUE clause cannot specify more than 120 alphabetic or alphanumeric characters.	7070/7074

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
VALUE Clause	VALUE IS literal	System/360 does not permit the use of the VALUE clause in the description of a group item.	7070-FOS
		The VALUE clause cannot be used in an item containing, or subordinate to an OCCURS clause. It cannot be used to specify the initial value of the variable portion of a record following the DEPENDING option.	7040/7044 7090/7094
		The VALUE clause cannot be used to describe a report item. It can be used to describe a group item, provided neither a decimal point nor an operational sign are part of the literal, and the size of the literal is equal to the size of the group item. A figurative constant may not be used for the predicate of a VALUE clause used at the group level, unless a SIZE clause is also present.	705/7080
VALUE OF Clause	VALUE OF FILE-ID IS external-name	This clause specifies the name by which the file is known to the control program. External-name consists of quotation marks enclosing no more than eight alphabetic and numeric characters, the first of which must be an alphabetic character. Neither the format nor the purpose of this clause is compatible with current systems usage.	System/360
	VALUE OF FILE-IDENTIFICATION IS literal-1 [RETENTION-PERIOD IS integer-1]	This clause specifies the contents of data items in the label record of the file. Literal-1 must be a ten-character nonnumeric literal. Integer-1 must be an integer up to four digits in length.	1410/7010

Table 2. COBOL DIFFERENCES.

COBCL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
VALUE CF Clause	VALUE OF { FILE IDENTIFICATION / FILE-SERIAL-NUMBER / REEL-SEQUENCE-NUMBER / CREATION-DATE / RETENTION-CYCLE / ADDITIONAL-LABEL-INFO } IS literal-1 [label-data-name-2 IS literal-2 ...]	Label-data-names must be chosen from the reserved words shown in braces, representing label-data-name-1.	7070/7074
	VALUE OF { ID / IDENTIFICATION / CREATION-DATE / RETENTION-CYCLE / FILE-SERIAL-NUMBER / ADDITIONAL-LABEL-INFO } IS literal-1 [label-data-name-2 IS literal-2] ...		705/7080
	VALUE OF { ID / IDENTIFICATION } IS literal-1 [PURGE-CYCLE IS literal-2]		705/7080
	VALUE OF { INPUT [OUTPUT] / OUTPUT } { FILE-IDENTIFICATION / FILE-SERIAL-NUMBER / REEL-SEQUENCE-NUMBER / CREATION-DATE / RETENTION-PERIOD / OPTIONAL-USAGE } IS literal-1 [label-data-name-2 IS literal-2 ...]		7070-FOS
	VALUE OF { FILE-IDENTIFICATION / CREATION-DATE / FILE-SERIAL-NUMBER / REEL-SEQUENCE-NUMBER / RETENTION-PERIOD } IS literal-1 [label-data-name-2 IS literal-2] ...		7040/7044

Table 2. COBOL DIFFERENCES.

COBOL ELEMENT	SYNTAX	ADDITIONAL SPECIFICATIONS OR COMMENTS	COMPILER
VALUE OF Clause	VALUE OF {FILE-IDENTIFICATION / FILE-SERIAL-NUMBER / REEL-SEQUENCE-NUMBER / RETENTION-PERIOD} IS literal-1 [label-data-name-2 IS literal-2] ...		7090/7094
WRITE Statement	WRITE record-name [FROM data-name-1] [INVALID KEY imperative-statement] [AFTER ADVANCING {data-name-2 / integer} LINES]	When the AFTER ADVANCING option is used, the first character in each logical record for the file must be reserved for the control character. If an integer is specified, it must be unsigned and have the value 0, 1, 2, or 3.	System/360
	WRITE record-name [FROM area-name]		1410/7010 7040/7044 7070-FOS 7090/7094
		The record is written directly from the area designated by area-name. Any record in the normal record area associated with the file is still available for processing.	705/7080 7070/7074
	WRITE record-name [FROM area-name] [INVALID KEY imperative-statement] [{AFTER / BEFORE} ADVANCING {integer LINES / mnemonic-name}]	Mnemonic-name is assigned in the SPECIAL-NAMES paragraph to a channel in the carriage tape.	1401/40/60

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
ACCEPT	ACCEPT Statement Procedure Verbs
ACCESS Clause	Environment Division
ACTUAL KEY Clause	Environment Division
ADD	Procedure Verbs
ADD CORRESPONDING	ADD Statement
ADDITIONAL-LABEL-INFO	VALUE OF Clause
ADVANCING	WRITE Statement
AFTER	WRITE Statement
ALL	DECLARATIVES Figurative Constant USE Sentence
Alpha-Form	Alphabetic Item
ALPHABETIC	CLASS Clause Class Test

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
Alphabetic Item	Types of Data Items
Alphabetic-Form	PICTURE Clause
Alphabetic-Literal	Alphabetic Item
ALPHANUMERIC	CLASS Clause
Alphanumeric Item	Types Of Data Items
Alphanumeric-Form	PICTURE Clause
Alphanumeric-Non-Report	PICTURE Clause
Alphanumeric-Report	PICTURE Clause
ALTER	Procedure Verbs
AN	CLASS Clause
An-Form	Alphanumeric Item
AND	Series Separator
APPLY Clause	Environment Division
Area-Name	READ Statement WRITE Statement

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
Argument	CALL Statement
Arithmetic Expression	Relation Test
ASSEMBLY-PROGRAM	ENTER Statement
ASSIGN Clause	Environment Division
AT END	AT END Condition READ Statement
Author-Name	Identification Division
AUTHORS	Identification Division
AUTOCODER	ENTER Statement
BCD	RECORDING MODE Clause
BEFORE	WRITE Statement
BEGINNING	USE Sentence
BEGINNING-LABEL	LABEL RECORDS Clause
BEGINNING-TAPE-LABEL	LABEL RECORDS Clause
BINARY	RECORDING MODE Clause

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
BLANK Clause	Record Description entry
BLOCK CONTAINS Clause	File Description Entry
CALL Statement	ENTER Statement
CARD-READER	Accept Statement
CHANGED	Reserved S/360 Only
CHARACTERS	BLOCK CONTAINS Clause LABEL RECORDS Clause RECORD CONTAINS Clause SIZE Clause
CHECK	Editing Clause
CHECKING	Reserved S/360 Only USE Sentence
CLASS Clause	Record Description Entry
Class Test	Test Conditions
CLOSE	Procedure Verbs
COBOL	ENTER Statement
Columns 73-80	Identification Code

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
COMMUNICATION-MODE	ENTER Statement
COMPUTATIONAL	Binary Item USAGE Clause
COMPUTATIONAL-1	Int. Floating Point Item USAGE Clause
COMPUTATIONAL-2	Int. Floating Point Item USAGE Clause
COMPUTATIONAL-3	Internal Decimal Item Reserved S/360 Only USAGE Clause
COMPUTE	Procedure Verbs
Condition-Name Test	Test Conditions
CONFIGURATION SECTION	Environment Division
CONSOLE	ACCEPT Statement DISPLAY Statement Reserved S/360 Only
CONSTANT SECTION	Data Division
COPY	File Description Entry Record Description Entry

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
COUNT-CONTROL	RECORDING-MODE Clause
CREATING	Reserved S/360 Only USE Sentence
CREATION-DATE	VALUE OF Clause
CYCLES	Reserved S/360 Only
DATA RECORDS Clause	File Description Entry
Data-Name	ACCEPT Statement ADD Statement Alphabetic Item Alphanumeric Item Binary Item CALL Statement DATA RECORDS Clause DISPLAY Statement DIVIDE Statement ENTRY Statement Ext. Floating Point Item Group Item Internal Decimal Item LABEL RECORDS Clause MOVE Statement MULTIPLY Statement OCCURS Clause PERFORM Statement

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
	Record Description Entry
	REDEFINES Clause
	Relation Test
	SIZE Clause
	SUBTRACT Statement
DENSITY	RECORDING MODE Clause
DEPENDING ON	GO TO Statement
	OCCURS Clause
	RECORD CONTAINS Clause
	SIZE Clause
DIGITS	SIZE Clause
DIRECT-ACCESS	Reserved S/360 Only
DISPLAY	Alphabetic Item
	Alphanumeric Item
	DISPLAY Statement
	Ext. Floating Point Item
	Procedure Verbs
	Report Item
	USAGE Clause
DISPLAY-ST	Reserved S/360 Only
DIVIDE	Procedure Verbs

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
DOLLAR	Editing Clause
Editing Clause	Record Description Entry
Elementary Item	Types Of Data Items
ELSE	AT END Condition
	Conditional Statement
	INVALID KEY Condition
	SIZE ERROR Condition
END	AT END Condition
	READ Statement
END DECLARATIVES	Declaratives
ENDING	USE Sentence
ENDING-LABEL	LABEL RECORDS Clause
ENDING-TAPE-LABEL	LABEL RECORDS Clause
ENTER	Procedure Verbs
ENTRY Statement	Data Division
	ENTER Statement
	Reserved S/360 Only
Entry-Name	CALL Statement
	ENTRY Statement

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
	RETURN Statement
EQUAL TO	Relation Test
EVEN	RECORDING MODE Clause
EXAMINE	Procedure Verbs
EXHIBIT	Reserved S/360 Only
EXIT	Procedure Verbs
Exponentiation	Arithmetic Expressions
External-Name	VALUE OF Clause
FD	File Description Entry
Figurative Constant	Literals
	Relation Test
File Description Entry	Data Division
FILE SECTION	Data Division
FILE-CONTROL	Environment Division
FILE-ID	Reserved S/360 Only
	VALUE OF Clause

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
FILE-IDENTIFICATION	VALUE OF Clause
File-Name	CLOSE Statement
	Declaratives
	File Description Entry
	OPEN Statement
	READ Statement
	USE Sentence
FILE-SERIAL-NUMBER	VALUE OF Clause
FILES	Declaratives
	USE Sentence
FILLER	Alphabetic Item
	Alphanumeric Item
	Binary Item
	Ext. Floating Point Item
	External Decimal Item
	Group Item
	Int. Floating Point Item
	Internal Decimal Item
	Record Description Entry
	Report Item
FLOAT	Editing Clause
Floating-Point-Form	PICTURE Clause

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
Floating-Point-Literal	Int. Floating Point Item
FORM-OVERFLOW	Reserved S/360 Only
Form-Overflow Test	Test Conditions
FP-Form	Ext. Floating Point Item
FROM	ACCEPT Statement File Description Entry Record Description Entry SUBTRACT Statement WRITE Statement
GIVING	ADD Statement SUBTRACT Statement MULTIPLY Statement DIVIDE Statement
GO TO	Procedure Verbs
GREATER	Relation Test
Group Item	Types Of Data Items
HIGH	RECORDING MODE Clause
HOLD	Procedure Verbs Reserved S/360 Only

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
I-O	OPEN Statement
I-O-CONTROL	Environment Division
IBM-360	Reserved S/360 Only
ID	VALUE OF Clause
ID DIVISION	Identification Division
IDENTIFICATION	VALUE OF Clause
IF	Class Test CONDITIONAL Statement
IN	ACCEPT Statement
INDEXED	Reserved S/360 Only
INPUT	OPEN Statement USE Sentence VALUE OF Clause
INPUT-OUTPUT	OPEN Statement
INPUT-OUTPUT SECTION	Environment Division
Integer	BLOCK CONTAINS Clause Editing Clause

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
	LABEL RECORDS Clause
	MOVE Statement
	OCCURS Clause
	PERFORM Statement
	POINT Clause
	RECORD CONTAINS Clause
	VALUE OF Clause
	WRITE Statement
INTO	DIVIDE Statement
	READ Statement
INVALID KEY	INVALID KEY Condition
	READ Statement
	WRITE Statement
JUSTIFIED Clause	Record Description Entry
JUSTIFIED RIGHT	Alphabetic Item
	Alphanumeric Item
KEY	INVALID KEY Condition
	READ Statement
	WRITE Statement
KEY-CONVERSION	Declaratives
	USE Sentence
LABEL RECORDS Clause	File Description Entry

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
Labels	Reserved S/360 Only
	USE Sentence
LEAVING	Editing Clause
LEFT	JUSTIFIED Clause
	POINT Clause
	SYNCHRONIZED Clause
LESS	Relation Test
Level-Number	Alphabetic Item
	Alphanumeric Item
	Binary Item
	Ext. Floating Point Item
	External Decimal Item
	Group Item
	Int. Floating Point Item
	Internal Decimal Item
	Record Description Entry
	Report Item
LIBRARY	File Description Entry
	Record Description Entry
Library-Name	File Description Entry
	Record Description Entry
LINES	WRITE Statement

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
LINKAGE	ENTER Statement Reserved S/360 Only
LINKAGE SECTION	Data Division
LINKAGE-MODE	ENTER Statement
LINKAGE-SYMBOL	CALL Statement
Literal	ADD Statement CALL Statement DISPLAY Statement DIVIDE Statement Figurative Constant MOVE Statement MULTIPLY Statement Relation Test STOP Statement VALUE Clause VALUE OF Clause
LOAD	RECORDING MODE Clause
LOCATION	POINT Clause
LOCK	CLOSE Statement
LOW	RECORDING MODE Clause
Margin Restrictions	COBOL Program Sheet

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
MD	File Description Entry
Mnemonic-Name	ACCEPT Statement DISPLAY Statement WRITE Statement
MODE	RECORDING MODE Clause
MORE-LABELS	Reserved S/360 Only
MOVE	Procedure Verbs RECORDING MODE Clause
MOVE CORRESPONDING	MOVE Statement
MULTIPLY	MULTIPLY Statement Procedure Verbs
NAMED	Reserved S/360 Only
NEXT SENTENCE	AT END Condition Conditional Statement INVALID KEY Condition SIZE ERROR Condition
NO REWIND	CLOSE Statement OPEN Statement
NO-TAPEMARK	LABEL RECORDS Clause

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
Non-Numeric Literal	Literals
	Alphanumeric Item
Non-Report Item	Types Of Data Items
NON-STANDARD	LABEL RECORDS Clause
NOT	Class Test
	Relation Test
NOTE Statement	Procedure Verbs
NUMERIC	CLASS Clause
	Class Test
Numeric Item	Types Of Data Items
Numeric-Form	Binary Item
	External Decimal Item
	Internal Decimal Item
	PICTURE Clause
	Report Item
Numeric-Literal	Binary Item
	External Decimal Item
	Internal Decimal Item
OBJECT-COMPUTER	Environment Division

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
OCCURS Clause	Alphabetic Item
	Alphanumeric Item
	Binary Item
	Ext. Floating Point Item
	External Decimal Item
	Group Item
	Int. Floating Point Item
	Internal Decimal Item
	Record Description Entry
	Report Item
ODD	RECORDING MODE Clause
OF	VALUE OF Clause
OMITTED	LABEL RECORDS Clause
OPEN	Procedure Verbs
OPTIONAL Clause	Environment Division
OPTIONAL-USAGE	VALUE OF Clause
ORGANIZATION	Reserved S/360 Only
ORGANIZATION Clause	Environment Division
OTHERWISE	AT END Condition
	CONDITIONAL Statement
	INVALID KEY Condition
	SIZE ERROR Condition

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
OUTPUT	OPEN Statement USE Sentence VALUE OF Clause
PARITY	RECORDING MODE Clause
PERFORM	Procedure Verbs
PICTURE Clause	Alphabetic Item Alphanumeric Item Binary Item Ext. Floating Point Item External Decimal Item Internal Decimal Item Record Description Entry Report Item
PLACES	Editing Clause POINT Clause
POINT Clause	Record Description Entry
PRINT-SWITCH	Reserved S/360 Only
Procedure-Name	GO TO Statement MOVE Statement PERFORM Statement
PROCESS	Procedure Verbs Reserved S/360 Only
PROGRAM-ID	Identification Division
Program-Name	Identification Division
PROTECT	Editing Clause
PROVIDE	ENTRY Statement
PURGE-CYCLE	VALUE OF Clause
Qualifiers	Qualification
RD	File Description Entry
READ	Procedure Verbs
READY	Reserved S/360 Only
RECEIVE	ENTRY Statement
RECORD CONTAINS Clause	File Description Entry
Record Description Entry	Data Division
Record-Name	DATA RECORDS Clause WRITE Statement

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
RECORDING MODE Clause	File Description Entry
RECORDS	BLOCK CONTAINS Clause
	DATA RECORDS Clause
	LABEL RECORDS Clause
	READ Statement
	RECORD CONTAINS Clause
REDEFINES Clause	Alphabetic Item
	Alphanumeric Item
	Binary Item
	Ext. Floating Point Item
	External Decimal Item
	Group Item
	Int. Floating Point Item
	Internal Decimal Item
	Record Description Entry
	Report Item
REEL	CLOSE Statement
REEL-SEQUENCE-NUMBER	VALUE OF Clause
Relation Test	Collating Sequence
	Test Conditions
RELATIVE	Reserved S/360 Only

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
RENAMING Clause	Environment Division
REPORT Clause	File Description Entry
Report Element Description Entry	Data Division
Report Group Description Entry	Data Division
Report Item	Types Of Data Items
REPORT SECTION	Data Division
Report-Form	PICTURE Clause
	Report Item
RERUN Clause	Environment Division
RESERVE Clause	Environment Division
RESTRICTED	Reserved S/360 Only
RETENTION-CYCLE	VALUE OF Clause
RETENTION-PERIOD	VALUE OF Clause
RETURN Statement	ENTER Statement
RETURNING	CALL Statement

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
REVERSED	OPEN Statement
REWRITE	Reserved S/360 Only
RIGHT	JUSTIFIED Clause POINT Clause SYNCHRONIZED Clause
ROUNDED	ADD Statement DIVIDE Statement MULTIPLY Statement SUBTRACT Statement
RUN	STOP Statement
SA	File Description Entry Reserved S/360 Only
SAME Clause	Environment Division
Saved-Area	Data Division
Saved-Area-Name	File Description Entry
Scientific-Decimal	PICTURE Clause
SD	File Description Entry
SEARCH	Reserved S/360 Only

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
SEEK Statement	Procedure Verbs
SELECT Clause	Environment Division
Semicolon	Punctuation
SIGN	Editing Clause
Sign Test	Test Conditions
SIGNED Clause	Record Description Entry
SIZE Clause	Record Description Entry
SIZE ERROR Condition	ADD Statement SUBTRACT Statement MULTIPLY Statement DIVIDE Statement
SIZE ERROR	SIZE ERROR Condition
Sort Description Entry	Data Division
Sort-File-Name	File Description Entry
SOURCE-COMPUTER	Environment Division

Table 3. Key Word Index To COBOL Differences.

KEY WCRD CR PHRASE	COBOL LANGUAGE ELEMENT
SPECIAL-NAMES	Environment Division
STANDARD	LABEL RECORDS Clause
STOP	Procedure Verbs
Subprogram-Name	CALL Statement
Subscript Format	Subscripting
SUBTRACT	Procedure Verbs
SUBTRACT CORRESPCNDING	SUBTRACT Statement
Switch-Status-Name Test	Test Conditions
SYMBOLIC KEY Clause	Environment Division
SYNCHRCNIZED Clause	Binary Item Int. Floating Point Item Record Description Entry
SYSIN1	ACCEPT Statement
SYSPCH	DISPLAY Statement Reserved S/360 Only
System-Symbol	CALL Statement

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
THAN	Relation Test
THEN	CONDITIONAL Statement
THRU	MOVE Statement PERFORM Statement VALUE Clause
TIMES	MOVE Statement OCCURS Clause PERFORM Statement
TRACE	Reserved S/360 Only
TRACKS	Reserved S/360 Only
TRANSFORM	Procedure Verbs Reserved S/360 Only
Unary Operations	Arithmetic Expressions
UNIT	CLOSE Statement
UNIT-RECORD	Reserved S/360 Only
UNITS	Reserved S/360 Only
USAGE Clause	Alphabetic Item Alphanumeric Item Binary Item

Table 3. Key Word Index To COBOL Differences.

KEY WORD OR PHRASE	COBOL LANGUAGE ELEMENT
	Ext. Floating Point Item
	Group Item
	Int. Floating Point Item
	Internal Decimal Item
	Record Description Entry
	Report Item
USE	Declaratives
	Procedure Verbs
USE BEFORE REPORTING	USE Sentence
USE FOR RANDOM PROCESSING	USE Sentence
USING	CALL Statement
	ENTRY Statement
UTILITY	Reserved S/360 Only
VALUE Clause	Alphabetic Item
	Alphanumeric Item
	Binary Item
	External Decimal Item
	Int. Floating Point Item
	Internal Decimal Item
	Record Description Entry
VALUE OF Clause	File Description Entry
VALUES	VALUE Clause
WITH NO REWIND	OPEN Statement
WITHOUT COUNT CONTROL	RECORD CONTAINS Clause Reserved S/360 Only
WORKING-STORAGE SECTION	Data Division
WRITE	Procedure Verbs
ZERO	BLANK Clause
ZERO SUPPRESS	Editing Clause
1401-CARD-READER	ACCEPT Statement

COMMENT SHEET

IBM SYSTEM/360 TRANSITION AIDS
COBOL LANGUAGE DIFFERENCES

Form C28-6570-0

FROM

NAME _____ OFFICE/DEPT NO. _____

CITY/STATE _____ DATE _____

To make this manual more useful to you, we want your comments: what additional information should be included in the manual; what description or figure could be clarified; what subject requires more explanation; what presentation is particularly helpful to you; and so forth.

FOLD FOLD

CUT ALONG LINE

FOLD FOLD

How do you rate this manual: Excellent _____ Good _____ Fair _____ Poor _____

Suggestion from IBM Employees giving specific solutions intended for award considerations should be submitted through the IBM Suggestion Plan.

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

FOLD ON TWO LINES, STAPLE, AND MAIL

STAPLE

STAPLE

OLD

FOLD

FIRST CLASS
PERMIT NO. 33504
NEW YORK, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
1271 AVENUE OF THE AMERICAS
NEW YORK, N.Y. 10020

ATTN: PROGRAMMING SYSTEMS PUBLICATIONS,
DEPARTMENT D39

CUT ALONG LINE

LD

FOLD

STAPLE

C28-6570-0

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601

ATTACHMENT I

ENVIRONMENT DIVISION

This division of COBOL is machine-oriented and therefore must be rewritten in its entirety at conversion time. However, to minimize conversion problems avoid the use of:

1. ALTERATION SWITCHES. S/360 does not use physical or electronic switches. Named data switches should be used as replacements.

2. Equating AUTOCODER TAGS and COBOL NAMES not allowed in S/360, for ENTER verb.

3. PROGRAM START. If it is desired to begin object program execution at some procedure other than the first procedure in the program, the first procedure statement should be a GO TO the desired paragraph.

4. SELECT RESTART clause.

5. RERUN clause.

ATTACHMENT I

DATA DIVISION

FILE SECTION.

1. FD entries should be rewritten at conversion time to conform to S/360 requirements, however, the following conventions should be adopted to facilitate the rewriting phase.

   A. BLOCK CONTAINS Clause.
      a. The CHARACTERS option is not implemented in S/360.
      b. Word RECORDS must be plural.

   B. RECORD CONTAINS Clause.
      a. Word CHARACTERS must be plural. This clause must be used in S/360 for variable-length records.

   C. LABEL RECORDS Clause.
      a. LABEL RECORDS ARE is the only form useable in S/360. The LABEL RECORDS clause is used only to specify user labels which may be omitted; the functions of current - system label procedures, if any, may not be required in S/360 COBOL.
      b. The STANDARD and NON-STANDARD options are not allowed.

   D. DATA RECORDS Clause.
      a. This clause is optional in S/360 since all data record descriptions must follow the File Description.

FILE SECTION AND WORKING-STORAGE SECTION.
DATA RECORD DESCRIPTIONS.

1. ALL "literal." A one-character literal must be used. If more than one type of character is required, use a literal or named constant of the same length as field.

2. ALL "Figurative Constant." This usage not valid in S/360. ALL is redundant, e.g. SPACES is equivalent to ALL SPACES.

3. PICTURE Clause.

   A. In S/360 this clause is the only permitted means to describe data items. The SIZE, SIGNED, POINT LOCATION, CLASS, CHECK PROTECT, FLOAT DOLLAR SIGN, ZERO SUPPRESS Clauses are not used. The S/360 PICTURE Clause is required for all elementary items except internal floating point.

   B. Do not use PICTURE Clause for group items.

   C. X is the only allowable character in an alphanumeric non-report item. The A X 9 combination is not allowed.

   D. High-order position CR and DR Symbols are not allowed.

ATTACHMENT 1

4. VALUE Clause

   A. Use only at the elementary level.

   B. Cannot be used in report items.

   C. Except in condition names, cannot be stated in an entry which contains or is subordinate to an entry which contains a REDEFINES clause.

5. OCCURS Clause.

   A. The word TIMES must be plural.

6. JUSTIFIED Clause.

   In S/360 alphanumeric and alphabetic items are justified left unless JUSTIFIED RIGHT is specified. Numeric items are aligned by decimal point. Therefore, JUSTIFIED LEFT is not permissable in S/360 COBOL.

7. SYNCHRONIZED Clause.

   This clause not required in S/360 but it will be accepted by the S/360 Compiler and treated as a comment.

8. REDEFINES Clause.

   A. Data-name-1 may not be subordinate to an item containing an OCCURS clause.

   B. Data-name-2 may not contain or be subordinate to an item containing an OCCURS clause.

CONSTANT SECTION.

There is no Constant Section in S/360 COBOL. Constants should be defined in Working-Storage section.

ATTACHMENT I

PROCEDURE DIVISION

1. Class test.
   The Class Test cannot be performed on a group item in S/360 COBOL.

2. Data comparison.
   Varying data formats of the S/360 may restrict the comparison of certain items in a data test; e.g. alphabetic and alphanumeric data may not be compared with numeric data.

COBOL VERBS.

READ:
In S/360, processing is done in the input area with registers being adjusted to refer to the proper logical record. This technique eliminates a move to a separate record area as is done in most current systems. This means that the record area will not be available following the READ that caused the AT END procedure to be executed.

Note that a READ INTO will still cause a move to a separate work area.

WRITE:
On a WRITE FROM, the prior record is not available in S/360 COBOL.

CLOSE:
S/360 uses the keyword UNIT instead of REEL, and provides the WITH NO REWIND and LOCK options.

DISPLAY:
This verb causes display on the System Output Unit (SYSOUT) unless SYSPCH (Punch) or CONSOLE is specified. A maximum of 72 data characters may be displayed on the punch or console.

ADD:
S/360 COBOL requires use of the TO or GIVING option with all ADD statements. Statement "ADD CURR-MO., YR-TO-DATE" is invalid.

PERFORM:
The word TIMES must be plural.

ENTER:
S/360 COBOL compiles directly into machine language. Current-system ENTER statements and equating of COBOL names will have to be rewritten. Avoid if possible.

NOTE:
NOTE, when used must begin a sentence.

COBOL PROGRAM SHEET

1. Text on the same line as the paragraph name may not start before column 12, and must be separated from the name by a period followed by at least one space.

ATTACHMENT I

2. In file description and record description entries, only the level indicator may appear before column 12.

3. S/360 allows continuation of non-numeric literals only. When a non-numeric literal is continued, it must be preceded by on quotation mark which may not appear before column 12.

4. AND. This keyword may not be used as a series separator or connector. It must be replaced with a blank or comma.

PROGRAM-ID in IDENTIFICATION DIVISION.
In S/360 COBOL, program-name must consist of from 1 to 8 alphabetic and/or numeric characters; the first character must be alphabetic.

USE OF PARENTHESES.
In S/360 COBOL, a left parenthesis cannot be followed by a space; a right parenthesis cannot be preceded by a space.

Compliance with the aforementioned specifications when writing current 7080 COBOL programs will minimize the changes necessary when converting to S/360 at a later date.

Attachment II, which follows, points up additional differences between the S/360 and 7080 COBOL which cannot be implemented currently.

ATTACHMENT II

OTHER SYSTEM/360 COBOL SPECIFICATIONS THAT WILL CAUSE 7080 PROGRAMS TO BE CHANGED FOR CONVERSION

1. The characters J and K cannot be used in the PICTURE clause.

2. SOURCE and OBJECT COMPUTER paragraphs will have to be rewritten in System/360 form.

3. MEMORY SIZE clause will not be allowed.

4. PREASSEMBLED IOCS clause will not be allowed in the System/360 operating system environment.

5. In the ASSIGN clause, assignment to specific units will not be allowed because of the device - independence concept of the System/360 operating system.

6. The RENAMING clause is not allowed in System/360.

7. RECORDING MODE clause is not allowed in FD entry.

8. The CHARACTERS option of the BLOCK CONTAINS clause is not implemented.

9. Data files may have to be changed to match the USAGE clause; the USAGE clause changed to match the data files, or a combination of both.

10. Use of the verb ACCEPT causes data to be accepted on the System Input Unit (SYSIN) unless FROM CONSOLE is specified.

11. Standard mnemonic names for specifying input devices, in connection with the ACCEPT verb, are provided.

12. The differences between System/360 and current system data forms may affect use of the MOVE verb.

ATTACHMENT III

GENERAL COMMENTS

1. System/360 COBOL compiles directly into machine language.

2. System/360 operating system does not use the current IOCS.

3. In System/360 no padding records are provided, whether blocked records are fixed or variable.

4. The System/360 label concept consists of a volume label which is completely controlled by the operating system and optional user labels which must be controlled by the user program. Because of the comprehensive volume - label processing which is provided, the functions of current system label procedures may not be required in System/360 COBOL.

5. The STANDARD and NON-STANDARD options and the BEGINNING-LABEL and ENDING LABEL options of the LABEL RECORDS clause are not allowed.

6. The DATA RECORDS clause is optional since all data record description must follow the File Description.

7. In the report form of a System/360 PICTURE, if all digit positions are defined by asterisks and the data field is all zeros, the edited item will consist of all asterisks instead of all blanks.

8. SYNCHRONIZED clause is never required because binary and floating point items are automatically synchronized. However, it will be accepted and treated as a comment.

9. In System/360 COBOL, declaratives are compiler - directing statements that specify the circumstances under which a procedure is to be executed in the object program. The USE sentence specifies the type of declarative.

10. System/360 has added the following COBOL words to the vocabulary of reserved words. They should not be used except in the specified context:

ATTACHMENT III

ACCESS
ACTUAL
CHANGED
CHECKING
COMPUTATIONAL-3
CONSOLE
COUNT
CREATING
CYCLES
DEVIATION
DIRECT-ACCESS
DISPLAY-ST
ENTRY
EXHIBIT
FILE-ID
FORM-OVERFLOW
HOLD
IBM-360
INDEXED
INVALID
I-O
LABELS
LINK
LINKAGE
MORE-LABELS
NAMED
ORGANIZATION
PRINT - STOP
PROCESS
PROCESSING
RANDOM
READY
RELATIVE
RESTRICTED
REWRITE
SA
SEARCH
SEQUENTIAL
SYMBOLIC
SYSPH
TRACE
TRACKS
TRANSFORM
UNIT
UNIT-RECORD
UNITS
UTILITY
VIA
WITHOUT
ZEROE

7080 - O.S. 360 COBOL DIFFERENCES

ACCEPT Statement.

7080 ACCEPT data-name (FROM mnemonic-name).

The standard ACCEPT device is the card reader.

Mnemonic-name, if specified, must be defined in the SPECIAL-NAMES paragraph.

360 ACCEPT data-name (FROM CONSOLE).

When FROM CONSOLE is specified, data-name may not exceed 72 characters in length. When the FROM CONSOLE option is not written, one logical record is read from the system logical input device (SYSIN).

ADD Statement.

ADD and ADD CORRESPONDING formats are unchanged.

7080 Computation on report items is permitted.

360 Either TO or GIVING is required in an ADD statement. There must be no OCCURS clause governing any of the operands involved in an ADD CORRESPONDING statement.

Arithmetic Expressions.

360 An expression having a succession of exponentiation operators separating a series of operands is evaluated as though parenthesigation were inserted from the right. Thus, A**B**C = (A**(B**)). The minus sign or Plus are the only allowable unary operator.

7080 When the sequence of execution is not specified by the parentheses, the order of execution of consecutive operations of the same hierarchical level is from left to right.

Both the plus and minus signs are valid unary operators.

AT END Condition.

7080 AT END statement - 1 ELSE statement - 2.

AT END is implied for statement - 2.

360 AT END imperative - statement.

AT END may not be implied.

BLANK Clause.

BLANK WHEN ZERO. Same format.

7080 May be used only in the description of an elementary item.

360 May be used only in the description of elementary report or numeric - form items. Numeric - form items to which it is applied are considered to be report items. The BLANK clause may not be applied to an item having an * in its picture.

BLOCK CONTAINS Clause.

7080 BLOCK CONTAINS integer-1 (RECORD, RECORDS, CHARACTERS).

360 BLOCK CONTAINS integer-1 RECORDS.

Specifies the number of records of maximum length in a physical record.

CLASS Clause.

7080 CLASS IS (_____).

This clause is not implemented by System/360.

A PICTURE clause should be used to express the functions of this clause.

CLASS Test.

7080 IF data - name IS (NOT) (NUMERIC, ALPHABETIC),

Data name must be defined in the Data Division as being of the alphanumeric class.

360 Data-name IS (NOT) (NUMERIC, ALPHABETIC), The item to be tested can be elementary alphanumeric, alphabetic, internal decimal, or external decimal.

CLOSE Statement.

7080 CLOSE file - name-1 (WITH LOCK),...

360 CLOSE file - name-1((UNIT) (WITH (LOCK, NO REWIND)),...

The reserved word UNIT replaces the current systems equivalent, REEL.

If the UNIT option is specified, the IOCS volume switching procedures are instituted.

COBOL Program Sheet - Margin Restrictions

For System/360 COBOL, margin A (columns 8 through 11) is restricted to the following entries:

Division Names

Section Names

Paragraph Names

Level Indicators

No other entries may begin before column 12.

Collating Sequence.

360 Comparison of non-numeric items is with respect to the binary collating sequence of characters in the IBM extended BCD interchange code.

Conditional Statement.

IF condition-1 (THEN) (statement-1, NEXT SENTENCE) ((ELSE, OTHERWISE) (statement-2, NEXT SENTENCE)). For Level E statement-1 must be imperative.

Continuation Indicator.

7080 A hyphen must be punched in column 7 of any line in which a word or literal is being continued.

360 A hyphen must be punched in column 7 of each line in which a non-numeric literal is being continued. Continuation of words or numeric literals is not permitted.

Data Division.

System/360 does not implement a Constant Section. Additions are as follows: Sort Description Entries, Saved Area Entries, Linkage Section, and Report Section.

Declaratives.

PROCEDURE DIVISION.

DECLARATIVES.

(section-name SECTION.

USE sentence.

((paragraph-name.)

sentence... .)...)...

END DECLARATIVES.

360 The USE sentence identifies the type of declarative. There are four options of the USE sentence. Each identifies one of the following types of processing:

1. Label-checking procedures.
2. Label-writing procedures.
3. Asynchronous processing procedures.
4. Report-writing procedures.

DISPLAY Statement.

7080 DISPLAY (data-name-1, literal-1) (UPON mnemonic-name).

If UPON is omitted, the console typewriter is assumed as the standard display device.

360 DISPLAY (data-name-1, literal-1) ((UPON (CONSOLE, SYSPCH)).

When the UPON option is omitted, the system logical output device (SYSOUT) is assumed. The sum of the operands may not exceed the maximum logical record length for SYSOUT.

If the UPON option is specified, the sum of the sizes of the operands may not exceed 72 character positions.

Editing Clause.

7080 (ZERO SUPPRESS, CHECK PROTECT, FLOAT DOLLAR SIGN) ((LEAVING integer-1 (PLACE, PLACES)).

This clause is not implemented by System/360. A PICTURE clause should be used to express the functions of this clause.

ENTER Statement.

7080 ENTER (AUTOCODER, COBOL).

360 ENTER LINKAGE.

(CALL statement., ENTRY statement., RETURN statement.)

ENTER COBOL.

The ENTER statement, used in conjunction with ENTRY or CALL statements, permits communication between a COBOL object program and one or more subprograms or other language subprograms.

ENTRY Statement

360 ENTRY entry-name (USING data-name).

This statement is used to establish an entry point in a COBOL subprogram. Control is transferred to the entry point by a CALL statement in another program.

Entry-name defines the entry point.

Data-name must be defined in the LINKAGE SECTION.

ENVIRONMENT DIVISION.

The nature of the Environment Division is such that it must be rewritten entirely whenever the program is to be processed for a different computer.

Figurative Constant

ALL "literal"

System/360 requires that literal be a single character bounded by quotation marks.

GO TO Statement.

GO TO procedure-name-1 procedure-name-2 (procedure-name-3...) DEPENDING ON data-name.

Data-name must be an elementary integral numeric item whose length does not exceed four digits.

IDENTIFICATION DIVISION.

7080 PROGRAM-ID. program-name.

(AUTHOR, AUTHORS). author-name.

Program-name must be either a name or a literal.

360 PROGRAM-ID. "program-name"

(AUTHOR. Author-name.)

Program-name must consist of from one to eight letters and (or digits, first character must be alphabetic, the whole enclosed in quotation marks. The program-name identifies the object program to the control program.

INVALID KEY condition.

360 INVALID KEY imperative-statement.

Applicable to random access devices.

JUSTIFIED Clause

360 JUSTIFIED RIGHT

This clause can only be used with elementary alphabetic or alphanumeric items.

LABEL RECORDS Clause.

7080 LABEL (RECORD, RECORDS) (IS, ARE) (OMITTED, STANDARD (BEGINNING-TAPE-LABEL) (NO-TAPEMARK) (ENDING-TAPE-LABEL)).

360 LABEL RECORDS ARE (OMITTED, data-name).

For other than unit-record devices, the OMITTED option implies either:

1. That there are no user labels on input, or
2. That user input labels are to be bypassed, or
3. That user labels are not created on output.

Use of the data-name option means that user labels can be processed in the declaratives section. Data-name must be described in the linkage section.

Literals. Numeric, Non-numeric, Figurative

1. HIGH VALUE(S) and LOW VALUE(S) are not implemented.
2. ZERO and its plural forms are not permitted in arithmetic statements.

MOVE CORRESPONDING data-name-1 TO data-name-2.

Implemented in the 360 level F compiler.

OCCURS Clause.

7080 OCCURS integer-1 (TIME, TIMES).

Must not be used in a record description entry having a level number of 77.

360 OCCURS integer-1 TIMES (DEPENDING ON data-name).

System 360 does not permit the use of an OCCURS clause in a level-88 entry.

Integer-1 represent a maximum in regards to the value of data-name.

OPEN Statement is different.

PERFORM Statement is identical

7080 The use of TIME and THROUGH in lieu of TIMES and THRU, respectively is permitted.

360 TIME and THROUGH are not permitted.

Data-name must have an integral value and data-name or integer must have a positive value, less than 32,768.

PICTURE Clause.

7080 PICTURE IS (alphabetic-form, numeric-form, alphanumeric-non-report, alphanumeric-report)

The single character pictures "J" and "K" represent a record mark and group mark, respectively. If the picture of an item specifies an implied decimal point, (symbol "V"), it cannot contain more than 99 positions on either side of the decimal point. Numeric items in arithmetic expressions are limited to 18 numeric characters. Report items are limited to 18 numeric characters. Z and * cannot appear in the picture. They may not appear in the same picture with a floating dollar sign. They may appear to the right of an assumed or actual decimal point only if no numerics appear.

360 PICTURE IS (Alphabetic-form, numeric-form, alphanumeric-form, report-form, floating-point-form). The picture symbol "X" is the only valid symbol for an alphanumeric item. Report-form is not a subdivision of the alphanumeric class.

Punctuation.

360 A semicolon may be used optionally to separate a series of statements or clauses.

READ Statement.

7080 READ file-name RECORD (INTO area-name).

When the INTO option is specified, the record is read directly into area-name. The record currently in the record area associated with the input file remains available.

360 READ file-name RECORD (INTO data-name).

If the INTO option is specified, the current record becomes available in the input area as well as in the area specified by data-name.

RECORD CONTAINS Clause.

7080 RECORD CONTAINS (integer-1 TO) integer-2 CHARACTERS.

360 RECORD CONTAINS (integer-1 TO) integer-2 CHARACTERS (WITHOUT COUNT CONTROL).

This clause is required if variable length records are described for the file. It is not necessary for a file having equal length records. If the WITHOUT COUNT CONTROL option is specified, the file is assumed to contain undefined-length or fixed length records and the BLOCK CONTAINS clause may not be specified for the file.

Record Description Entry.

360 (77,01) data-name COPY library-name.

level-number (FILLER, data-name) (REDEFINES Clause) USAGE Clause) (OCCURS Clause) (VALUE Clause) (SYNCHRONIZED Clause) (BLANK Clause) (PICTURE Clause) (JUSTIFIED Clause).

7080 SIZE, POINT, and Editing Clauses are functionally replaced by the PICTURE Clause.

RECORDING MODE Clause

7080 RECORDING MODE IS mode.

System/360 does not implement this clause.

REDEFINES Clause.

Both: data-name-1 REDEFINES data-name-2.

7080 Multiple redefinitions are possible, provided that they occur at the same level and without other entries intervening at that level.

An entry containing the REDEFINES clause must not be written at a lower level within another entry containing the REDEFINES clause, unless the latter entry is at the 01 level, in which case one such lower level redefinition is permitted at a time.

360 REDEFINES clauses may be specified for items subordinate to items containing REDEFINES clauses. Data-name-1 may not contain or subordinate an OCCURS DEPENDING clause, nor be subordinate to an item containing an OCCURS clause. Data-name-2 may not be subordinate to nor contain an OCCURS clause, nor subordinate an item containing an OCCURS DEPENDING clause. No lower level-number entries may intervene between data-name-1 and data-name-2. Length of data-name-1 times occurrences must not be greater than the length of data-name-2.

Relation Test

360 (data-name, literal, arithmetic expression, figurative-constant) IS (NOT) ($<$, $>$, = , GREATER THAN, LESS THAN, EQUAL TO) (data-name, literal, arithmetic expression, figurative-constant).

Reserved System/360 Only

CHANGED	FORM-OVERFLOW	RELATIVE
CHECKING	HOLD	RESTRICTED
COMPUTATIONAL - 3	IBM-360	REWRITE
CONSOLE	INDEXED	SA
CREATING	LABELS	SEARCH
CYCLES	LINKAGE	SYSPCH
DIRECT-ACCESS	MORE-LABELS	TRACE
DISPLAY-ST	NAMED	TRACKS
ENTRY	ORGANIZATION	TRANSFORM
EXHIBIT	PRINT-SWITCH	UNIT-RECORD
FILE-ID	PROGRESS	UNITS
WITHOUT	READY	UTILITY

These words are not reserved in any of the current systems vocabulary.

SIGNED Clause.

7080 SIGNED

This clause is not implemented by System/360.

A PICTURE clause should be used to express the functions of this clause.

SIZE Clause.

7080 SIZE IS integer-1 (CHARACTER, CHARACTERS, DIGIT, DIGITS)

This clause is not implemented.

SIZE ERROR Condition.

7080 ON SIZE ERROR (statement-1, NEXT SENTENCE) ((OTHERWISE, ELSE) (statement-2, NEXT SENTENCE)).

For compatibility with System/360, the option should not be used.

360 ON SIZE ERROR imperative-statement.

STOP Statement

360 STOP (RUN, literal).

STOP RUN returns control to the operating system. STOP literal causes a compiler-generated message code, as well as the specified literal, to be displayed. The message code must be keyed in on the console in order to resume.

SYNCHRONIZED Clause

7080 SYNCHRONIZED (LEFT, RIGHT).

The SYNCHRONIZED Clause is never necessary in a System/360 COBOL program. If the clause is specified, it is treated as comments by the compiler.

Test Condition

7080 Switch-status-name Test

The System/360 environment does not provide for testing switch-status.

USAGE Clause

7080 USAGE IS (COMPUTATIONAL, DISPLAY).

The DISPLAY option is not meaningful and is ignored. If an item is specified as being COMPUTATIONAL, it must be a numeric item.

360 USAGE IS (DISPLAY, COMPUTATIONAL, COMPUTATIONAL-1, COMPUTATIONAL-2, COMPUTATIONAL-3).

DISPLAY specifies character form, one character per byte. COMPUTATIONAL specifies a binary item occupying 2, 4, or 8 byte positions corresponding to the specified decimal lengths of 1-4, 5-9, and 10-18 respectively. The leftmost bit of the reserved area is the operational sign. Computational items are aligned at the next half or full word boundaries, as appropriate. COMPUTATIONAL-1 short-precision floating point. COMPUTATIONAL-2 long-precision floating point. COMPUTATIONAl-3 packed decimal.

USE Statement. Check the manual.

VALUE Clause.

Both VALUE IS literal.

7080 The VALUE clause cannot be used to describe a report item. It can be used to describe a group item, provided neither a decimal point nor an operational sign are part of the literal, and the size of the literal is equal to the size of the group item. A figurative constant may not be used for the predicate of a VALUE clause used at the group level, unless a SIZE clause is also present.

360 A VALUE clause can be used:

1. For elementary items only.
2. In File and Linkage Section, in level 88 entries only.

The VALUE clause cannot be used:

1. For group items.
2. For report items.
3. For external floating-point items.
4. For items containing, or subordinate to an OCCURS clause or a REDEFINES clause (except for level 88 entries).

VALUE OF Clause.

360 VALUE OF FILE-ID IS enternal-name.

This clause specifies the name by which the file is known to the control program. Any other format is invalid.

WRITE Statement.

7080 WRITE record-name (FROM area-name).

The record is written directly from the area designated by area-name. Any record in the normal record area associated with the file is still available for processing.

360 WRITE record-name (FORM data-name-1) (INVALID KEY imperative-statement) (AFTER ADVANCING (data-name-2, integer) LINES).

When the AFTER ADVANCING option is used, the first character in each logical record for the file must be reserved for the control character. If an integer is specified, it must be unsigned and have the value 0, 1, 2, or 3.