



IBM Systems Reference Library

IBM 705/7080 Programming Systems
COBOL: Additional Specifications

This publication, when used with the IBM publication COmmon Business Oriented Language (COBOL), General Information, Form F28-8053, provides the information required to write COBOL programs for the 705 and 7080 Data Processing Systems. The information in this publication applies to both the 7058 COBOL Processor (#0705-PR-131) and the 7080 COBOL Processor (#7080-CB-933). Any differences between these processors are noted, and specific requirements for the individual processors are described.

MAJOR REVISION (April 1964)

This publication, Form J28-6177-3, supersedes the previous edition, Form J28-6177-2, and converts it to Systems Reference Library (SRL) format. The previous edition is now obsolete. The information supplied in Technical Newsletter N28-1058 has been incorporated into the present edition.

This edition also includes descriptions of several changes in the 7080 COBOL Processor that were made subsequent to the appearance of the technical newsletter. These changes are:

1. Hypertape capabilities have been added.
2. The ADD CORRESPONDING and SUBTRACT CORRESPONDING options have been implemented.
3. The BLIXT macro-instruction (to save and restore storage banks) is generated automatically in an interrupt program.
4. The implied AT END option of the READ statement is disallowed. (The 7058 COBOL Processor now produces a warning message with no other change in former generation.)

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

Address comments concerning the contents of this publication to:

IBM Corporation, Programming Systems Publications, Dept. D91, PO Box 390, Poughkeepsie, N.Y. 12602

CONTENTS

INTRODUCTION	5	APPENDIX B: USE OF IOCS	23
Prerequisite Reading	5	705 III Only	23
Acknowledgement	5	7080 Only	23
PHYSICAL DESCRIPTION OF THE PROCESSORS	6	APPENDIX C: PREASSEMBLED IOCS FOR THE 705 III	
PROCESSOR-ORIENTED SPECIFICATIONS	7	(7058 COBOL PROCESSOR ONLY)	24
General	7	Introduction	24
Identification Division	7	Assembling an IOCS	24
Environment Division	7	Macro-Instructions Used in an IOCS Assembly	25
Configuration Section	7	IODEF (Define IOCS)	25
Special-Names Paragraph	8	BLSYS (Include Input/Output Control System)	25
Input/Output Section	9	BLHSK (Include Input/Output Housekeeping Section of IOCS)	26
File-Control Paragraph	9	Assembling an Object Program To Be Used With a	
I-O-Control Paragraph	11	Preassembled IOCS	26
Data Division	13	Inclusion of the 705 III Preassembled IOCS With the	
File Description Entries	13	Object Program	27
Record Description Entries	14	APPENDIX D: PREASSEMBLED IOCS FOR THE 7080	28
Procedure Division	17	Introduction	28
Additional Specifications	17	Assembling an IOCS	28
Procedure Division Verbs	17	Assembling an Object Program To Be Used With a	
Add Corresponding Option	19	Preassembled IOCS	28
Subtract Corresponding Option	19	Inclusion of the 7080 Preassembled IOCS With the	
APPENDIX A: TAPE LABELS AND SPECIALIZED ROUTINES	20	Object Program	28
Label Control Cards	20	APPENDIX E: DEFERRED ELEMENTS OF THE COBOL	
Date Control Card	20	LANGUAGE	29
File Serial Header Control Card	20	APPENDIX F: COMPLETE LIST OF COBOL RESERVED	
705 III End Control Card	20	WORDS	30
7080 End Control Card	20	APPENDIX G: ADDITIONAL CONSIDERATIONS	34
Labels and Special Processing Considerations	20	Restrictions on the Use of the COBOL Language	34
Checking the Record Count	20	Language Incompatibility	34
Coding Special Routines When Using the 7080 IOCS	21	Verbs	34
Use of Specialized Routines	21	Clauses	34
Input Files	21	Special Characters	34
Output Files	22	INDEX	35

The publication Common Business Oriented Language (COBOL), General Information, Form F28-8053-2, describes the COBOL language in detail but does not supply information pertaining to specific computers or to particular COBOL processors. The present publication provides the programmer with additional specifications he needs to write complete COBOL source programs for the IBM 705 and 7080 Data Processing Systems.

This publication applies to both the 7058 COBOL Processor and the 7080 COBOL Processor. Where differences between these processors exist, they are noted, and specific requirements for the individual processors are described. All Hypertape references apply to the 7080 COBOL Processor only.

PREREQUISITE READING

The reader of this publication should be familiar with the contents of the following IBM publications:
Common Business Oriented Language (COBOL), General Information, Form F28-8053
IBM 705/7080 Applied Programming Tape Format and Labeling Standards, Form J28-6123
Input/Output Control System for the IBM 705 III, Form C28-6019
IBM 7080 Input/Output Control System for Use with 729 Magentic Tape Units, Form C28-6237
IBM 7080 Input/Output Control System for 7340 Hypertape Drives -- #7080-IO-932, Form C28-6341

ACKNOWLEDGMENT

In accordance with the requirements of the official government manual describing COBOL, 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, United States Department of Commerce
Burroughs Corporation
David Taylor Model Basin, Bureau of Ships, United States Navy
Electronic Data Processing Division, Minneapolis-Honeywell Regulator Company
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
The Bendix Corporation, Computer Division
Control Data Corporation
E. I. du Pont de Nemours and Company
General Electric Company
General Motors Corporation
Lockheed Aircraft Corporation
The National Cash Register Company
Philco Corporation
Standard Oil Company (New Jersey)
United States Steel Corporation

"This COBOL-61 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 assured 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® I and II, Data Automation Systems ©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 use of this material, in whole or in part, in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

"Any organization interested in reproducing the COBOL report and initial specifications in whole or in part, using ideas taken from this report or utilizing this report as the basis for an instruction manual or any other purpose is free to do so. However, all such organizations are requested to reproduce this section as part of the introduction to the document. Those using a short passage, as in a book review, are requested to mention 'COBOL' in acknowledgment of the source, but need not quote this entire section."

PHYSICAL DESCRIPTION OF THE PROCESSORS

A COBOL source program will be compiled by either the 7058 or 7080 COBOL Processor into Autocoder entries in blocked card-image form on tape. The Autocoder entries will then be assembled by the 7058 or 7080 Processor into an object program.

The 7058 COBOL Processor is designed to operate on the 705 II, 705 III, or 7080, and will compile source programs written for the 705 I, 705 II, 705 III or 7080. The 7080 COBOL Processor will operate only on the 7080 and is designed to compile only source programs written for the 7080.

It is always possible to proceed immediately to compilation, but it is generally advisable to make an analysis run first. The analysis run is several times faster than a full compilation, and a listing of all source statements together with diagnostic messages is produced.

During compilation, a listing of the following information will be produced in sequence, along with the object program:

1. A list of Autocoder literals.
2. A transcription of the COBOL Identification Division.
3. Diagnostic messages from COBOL.
4. A transcription of the COBOL Environment Division.
5. A list of file-names and their assigned tape units, followed by IOCS file tables, when the OBJECT-COMPUTER is the 705 III or 7080.
6. The COBOL Data Division entries organized as follows:

- a. File Section -- for each file and its associated records, a transcription of the File Description entry followed by a transcription of the first record (level 01 and all subordinate Record Description entries), followed by the generated Autocoder entries for the first record; a transcription of the second record, if any, etc.
- b. Working-Storage and Constant Sections -- a transcription of the first independent Record Description entry (level 77) followed by its generated Autocoder entries, followed by a transcription of the second independent Record Description entry, if any, etc.; a transcription of the first group item (level 01 and all subordinate Record Description entries) followed by its generated Autocoder entries, followed by a transcription of the second group item, if any, etc.

7. The COBOL Procedure Division entries organized as follows: a transcription of each procedural paragraph followed by the generated Autocoder entries for the paragraph.

8. Miscellaneous generated Autocoder entries.

9. Glossaries of cross references between COBOL names and Autocoder tags generated by the Processor, listed both in order of the source statements and alphabetically by tag name.

The numeric and alphanumeric literals in the COBOL Procedure Division will be converted into Autocoder literals or constants, as required.

PROCESSOR-ORIENTED SPECIFICATIONS

GENERAL

The programmer should be aware of the following specifications in coding his COBOL program for the 705/7080:

1. The computer character set (using set H representation where applicable) shown in collating sequence from left to right is as follows:

Blank .) ≠ + \$ * - / , (= ' 0⁺ A to I 0⁻ J to R ≠
S to Z 0 to 9

2. The figurative constant LOW-VALUE is represented by the blank character. The figurative constant HIGH-VALUE is represented by the 9 character. It should be remembered that HIGH-VALUE, like all figurative constants except ZERO, is considered to be alphanumeric and should only be used where an alphanumeric literal would be appropriate.

3. Literals are limited to 120 character positions. If a numeric literal contains a decimal point, however, the literal may not contain more than 99 digits on either side of the decimal point. In addition, a numeric literal is limited to 18 digits when it appears in an arithmetic expression.

4. The splitting of words or literals between coding lines requires that a hyphen be written in column 7 of the line containing the continuation. In addition, the continuing portion of alphanumeric literals must be preceded by a quotation mark. In effect, a continuing alphanumeric literal contains at least three quotation marks -- the first begins the literal, the second begins the continuation of the literal on the next line, and the third ends the literal.

5. Condition-names that are written in the Data or Environment Divisions may appear only in conditional expressions in the Procedure Division. Further, they may not be cross-referenced by an Autocoder tag in the Environment Division (see "Special-Names Paragraph").

IDENTIFICATION DIVISION

[DATE-COMPILED. *any sentence.*]

When the current date has been entered into the 705/7080 at compilation time, the Processor will insert the date into the transcription of the DATE-COMPILED paragraph on the Autocoder output listing and will replace "any sentence" that may have been written. In addition, the date will appear in the heading at the top of each page of the Autocoder output listing along with the name (first six characters) of the program written in the PROGRAM-ID paragraph.

ENVIRONMENT DIVISION

When specific Environment Division source entries have been written, they must be entered into the 705/7080 in the following sequence:

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. ----
OBJECT-COMPUTER. ----
SPECIAL-NAMES. ----
INPUT-OUTPUT SECTION.
FILE-CONTROL. ----
I-O-CONTROL. ----

Within a given paragraph, the order of clauses (or sentences) is not critical. Details of specific entry formats are discussed in the remainder of this chapter.

Standard formats must be used in entries calling for the device-names shown below.

Card Readers: READER 01cu
where c represents the card reader control unit number (0 through 9 on the 714 Card Reader, 0 only on the 7502 Card Reader) and u represents the 714 or 7502 Card Reader number (0 through 9).

Tape Units: 729-m c
729-m 02cu (705 III)
729-m 2c0u (7080)
HYPERTAPE-UNIT(S) cu (7080)

where m represents the tape unit model number (1 or 3 on the 705 III; 2, 4, 5, or 6 on the 7080), c represents the channel number (0 through 3 for 729 tape units; 0, 1, 4 through 7 for HYPERTAPE-UNITS), and u represents the tape unit number (0 through 9).

Configuration Section

The Configuration Section is concerned with describing the computers on which the Processor and object programs are to be run and with relating the names of computer components and devices to names written by the programmer in the Procedure Division of his source program.

SOURCE-COMPUTER. *computer-name.*

Computer-name should be 705-2, 705-3, or 7080 if the 7058 Processor is used. Computer-name must be 7080 if the 7080 Processor is used.

OBJECT-COMPUTER. *computer-name*

Computer-name should be 705-1, 705-2, 705-3, or 7080 if the 7058 Processor is used. If the OBJECT-COMPUTER is a 7080 being operated in 705 III mode, 705-3 should be specified. Computer-name must be 7080 if the 7080 Processor is used.

[MEMORY SIZE { *integer-1* CHARACTERS
ADDRESS *integer-2* [THRU *integer-3*] }]

The MEMORY option is required only for 705 III or 7080. Integer-1 under the CHARACTERS option should be 40000, 80000, or 160000. Integer-2 under the ADDRESS option represents a program origin (starting point) other than standard on any of the above computer-names. On the 705 I and 705 II, standard origin is 0160. On the 705 III and 7080, the Channel or Tape Tables begin at 0240 and 0500, respectively. If another origin is specified, it will cause the balance of the program (but not the tables) to begin at the specified location. If that location overlaps the tables, the program will originate immediately following them. Integer-3 under the ADDRESS option must be specified on the 705 III or 7080 as 39999, 79999, or 159999.

[PREASSEMBLED
{ IOCS ADDRESS *integer-4*
HYPERTAPE-IOCS ADDRESS *integer-5*
IOCS ADDRESS *integer-4* HYPERTAPE-IOCS ADDRESS *integer-5* }
IOHSK ADDRESS *integer-6* THRU *integer-7*]

When the 705 III or 7080 Input/Output Control System (IOCS) has been preassembled, the PREASSEMBLED IOCS option must be specified. Integer-4, integer-5, and integer-6 represent the starting locations of 729 IOCS, Hypertape IOCS, and IOHSK, respectively. Integer-7 represents the ending location of IOHSK. Integer-6 will also be considered the first address of erasable housekeeping by the Processor and will be used as the starting location of the IOCS output areas.

[*integer-8* TAPE-UNIT(S)
integer-9 HYPERTAPE-UNIT(S)]

This optional clause is used if the Processor is to check to insure that the number of tapes assigned in the File-Control Section does not exceed the number of available tapes. TAPE-UNIT(S) refers to 729 tape units only.

Special-Names Paragraph

The SPECIAL-NAMES paragraph is never required. If it is written, however, any of the following clauses may appear:

[ALTERATION-SWITCH 091u [IS *mnemonic-name*]
[ON STATUS IS *condition-name-1*]
[OFF STATUS IS *condition-name-2*] ...]

There are six Alteration switches available on the 705 and 7080. The letter u represents the switch number (1 through 6). The mnemonic-name option is required only when condition-name-1 or condition-name-2 is not unique. Mnemonic-name should then be used to qualify the condition-names when referenced in the Procedure Division. Mnemonic-name itself may not be qualified. Condition-names assigned to Alteration switch settings may appear only in conditional expressions in the Procedure Division, and, since the switches are external console switches, they cannot have storage space reserved for them through an entry in the Data Division.

[{ READER 01cu
TYPEWRITER
PUNCH 0300
PRINTER 0400 } IS *mnemonic-name...*]

READER01 *cu* may be assigned to a mnemonic-name associated with the ACCEPT verb in the Procedure Division.

TYPEWRITER, PRINTER 0400, and PUNCH 0300 may be assigned to a mnemonic-name associated with the DISPLAY verb in the Procedure Division.

Any of the above device-names may be assigned to as many different mnemonic-names as desired, as long as they are correctly associated with the ACCEPT and DISPLAY verbs. Mnemonic-name may not be qualified.

Input/output messages are always displayed on the console typewriter by the 705 III or 7080 IOCS regardless of any device-names specified in this paragraph.

[Autocoder-tag IS COBOL-name...]

When Autocoder entries are written under an ENTER AUTOCODER statement in the Procedure Division, it may be desirable to refer to COBOL procedure-names or data-names in Autocoder language. This may be accomplished by assigning Autocoder tags to COBOL names in this paragraph. Autocoder tags must satisfy all of the Autocoder rules for defining tags. In addition, they are limited to five characters in length, they may not consist of an alphabetic character followed by four digits, and they may not begin with CS (7080 only), BL, or IO.

COBOL names may be procedure-names or data-names and may include qualification. They may not, however, be condition-names.

It should be remembered that assigning Autocoder tags to COBOL names allows only an Autocoder instruction to reference an item defined in COBOL language. It does not allow a COBOL statement to reference an item defined in Autocoder language. The latter situation requires an ENTER AUTOCODER statement followed by the appropriate Autocoder instruction.

[PROGRAM-START IS procedure-name]

At object time, program execution normally begins with the first procedure-name written in the Procedure Division (after IOCS Housekeeping has been accomplished on the 705 III or 7080). To begin execution at some procedural statement other than the first statement written in the Procedure Division, a procedure-name should be designated as PROGRAM-START under this paragraph.

The procedure-name designated may be a section-name, or it may be a paragraph-name with or without qualification by a section-name. However, it must not be a paragraph-name preceding the NOTE verb.

Input/Output Section

In the absence of standard tape-record formats and tape-handling conventions for the 705 I and 705 II, all tape input/output operations must be coded in Autocoder. Using the ENTER AUTOCODER statement in the Procedure Division, the programmer should code his own blocking, deblocking, reading, writing, and label-checking routines. The SPECIAL-NAMES paragraph, as outlined above, will be used to establish equivalences between names of work areas that the programmer has defined in the Working-Storage Section of the Data Division and the Autocoder tags used in the blocking and deblocking routines.

Input/output handled by the on-line card reader, card punch, or 717 Printer may, however, be

written in COBOL language for the 705 I and 705 II, as well as the 705 III and 7080, using the appropriate clauses in the FILE-CONTROL paragraph and the File Description entry in the Data Division. The I-O-CONTROL paragraph is not used on either the 705 I or 705 II.

The following discussion of IOCS procedures is directed toward the IBM 705 III and 7080 IOCS.

IOCS automatically blocks and deblocks records, performs checks, and schedules tape movement for maximum efficiency. A brief discussion of some of the terms used in specifying tape input/output operations follows. The actual method of writing the necessary entries will be covered after this general discussion.

IOCS contains a Tape Reel Control System that includes routines for checking input reels to insure that they belong to the proper file and that they are processed in proper sequence. Output reels are checked to make sure valuable information cannot be destroyed through operator error. These checks are regulated by the header (beginning-of-reel) and trailer (end-of-reel or end-of-file) labels.

Standard header labels contain such information as the date the file was written, the length of time the file is to be saved, a file identification, a file serial number, and a reel sequence number, whereas standard trailer labels contain an end-of-reel or end-of-file indication and a record count. (For complete specifications of standard header and trailer labels, see the Tape Format publication.)

In addition to the information provided by labels, IOCS requires such additional information as the current date and the serial numbers of the first reel of each input file. This information must be supplied at object time through label control cards (see Appendix A) that are read either from the card reader or from the beginning of the first reel of an output file. The manner of loading the label control cards is specified in the "File-Control Paragraph."

Checking of labels is automatic only if labels are standard as specified in the Tape Format publication. If the program uses files with nonstandard or omitted labels, the programmer must provide his own checking routines. He can write these special routines in COBOL language in the Procedure Division (see APPLY under "I-O-Control Paragraph" and Appendix A).

Additional procedures, such as taking checkpoint and dumping unreadable tape records, will be accomplished automatically by IOCS if the programmer supplies the necessary entries under the FILE-CONTROL and I-O-CONTROL paragraphs. Provision for restarting a program that has been interrupted is also made available.

File-Control Paragraph

Every file name referenced in the Procedure Division must be selected and assigned in this paragraph. Input files may be card or tape. Output files may be card, tape, or printer.

SELECT *file-name-1* ASSIGN TO

file-name-2
 { 729-m { 02cu } { 2c0u } } { 729-m { 02cu } { 2c0u } } ...
 [FOR MULTIPLE REEL]
 [HYPERTAPE-UNIT *cu* [WITH AUTOLOADER]]
 [HYPERTAPE-UNIT *cu* [WITH AUTOLOADER] ...]
 [READER 01*cu*]
 [PUNCH 0300]
 [PRINTER 0400]

The file-name-2 option of ASSIGN implies merely that the tape units assigned to file-name-2 will be assigned to file-name-1. This option may only be utilized in conjunction with tape files and would normally be used to assign an input file to the same units as an output file when these files are to be used sequentially. (See the CLOSE verb in the Procedure Division.)

The MULTIPLE REEL clause has no significance to IOCS and therefore will only be transcribed by the Processor.

Specific assignments must follow these rules:

1. On the 705 III, all tape units assigned to a file must be of the same model and must be attached to the same channel.
2. For any program, a maximum of five tape units may be assigned to a file.
3. On the 7080, Hypertape units and 729 units may not be mixed for a file.
4. On the 7080, it is recommended (though not required) that all tape units assigned to a file be attached to the same channel.

The remaining options of the ASSIGN clause may be specified for files other than tape.

[RENAMING *file-name-3*]

RENAMING implies that the File and Record Description entries written in the File Section of the Data Division for file-name-3 are to be used in conjunction with file-name-1. However, both files must be selected and assigned, and a tape file may rename only a tape file. This clause would normally be written when the file characteristics of an input and output file are identical and when both files are to share a common record storage area.

File-name-2 under the ASSIGN clause and file-name-3 may be the same name only when sequential use of two files is desired. (See the CLOSE verb in the Procedure Division.)

[RESERVE { *integer-2* } ALTERNATE AREA(S)]
 [NO]

The RESERVE clause is meaningful only when standard IOCS is used on the 705 III or 7080 and the input/output medium is tape. If the clause is not specified, two input/output areas will be reserved. If NO is specified, one input/output area will be reserved. If integer-2 is specified, it may range from 1 through 7, and the total number of areas reserved will be 2 through 8.

Additional File-Control Specifications

Four special "files" may be specified under the FILE-CONTROL paragraph when the 705 III or 7080 IOCS is used. These are files that are under exclusive control of IOCS and must not be referenced by the programmer in the Procedure Division. When these files are specified, the RENAMING and RESERVE clauses are never written. In addition, when a Data Division File Description entry is required, as explained below, the BLOCK, RECORD, and DATA RECORD clauses (as well as the Record Description entries for data records) should never be written. The format and description of each of these special files are as follows.

SELECT ERROR-DUMP ASSIGN TO
 [TYPEWRITER]
 [HYPERTAPE-UNIT *cu* [WITH AUTOLOADER]]
 { 729-m { 02cu } { 2c0u } }

This file is provided when the programmer wants IOCS to dump tape input records that are unreadable. If TYPEWRITER is assigned, no File Description entry is required. If a tape unit is assigned, it must be a separate tape unit and, as such, a File Description entry is required in order to specify label requirements (and also DENSITY on the 7080). (See also APPLY in the "I-O-Control Paragraph.") If ERROR-DUMP is not selected, the console typewriter will be assigned automatically.

```

SELECT RERUN-WORK ASSIGN TO
{
  HYPERTAPE-UNIT cu [WITH AUTOLOADER]
  {
    729-m {02cu}
           {2c0u}
  }
  file-name-n
}

```

When checkpoints are to be taken, an additional work tape (other than the one containing the checkpoint records) is required. The tape is always repositioned, and assigning an output file as RERUN-WORK in no way affects the file. If a separate tape unit is assigned, a File Description entry is required in order to specify label requirements (and also DENSITY on the 7080). No trailer labels may be specified for RERUN-WORK. (See also APPLY in "I-O-Control Paragraph.")

If file-name-n is specified, it may be either a source program output file or ERROR-DUMP. If an output file is specified, the file must not be closed until all checkpoints have been passed. If ERROR-DUMP is specified, it must have been assigned to a tape unit. No File Description entry is required under this option.

```

SELECT LABEL-DATA ASSIGN TO { READER 01cu }
                             { file-name-n }

```

As explained in Appendix A, IOCS requires certain control information for automatic checking of standard labels. This information may be entered into the computer from either the card reader or from the first reel of a source program output file. No File Description entry is required under either option. If this statement is omitted and label cards are required, the card reader will be assumed to be the input device.

```

SELECT RESTART ASSIGN TO
{
  READER 01cu
  HYPERTAPE-UNIT cu [WITH AUTOLOADER]
  {
    729-m {02cu}
           {2c0u}
  }
}

```

When it is desired to restart a program from a checkpoint, provision should be made for loading the restart program. The restart unit may be either the card reader or a separate tape unit not used for other purposes. No File Description entry is required in either case.

If a tape unit is specified, the tape may have either a standard program tape header (see the Tape Format publication) or no header. If present, the header must contain instructions to space over the tape mark, read the first record of the restart program into storage, and turn off the I/O indicator (IOF) before transferring control to the restart program.

I-O-Control Paragraph

I-O-CONTROL paragraph entries may be specified in conjunction with tape files only when standard IOCS is used on the 705 III or 7080.

```

APPLY
{
  BEGINNING-REEL procedure-name-1 [THRU procedure-name-2]
  ENDING-REEL procedure-name-3 [THRU procedure-name-4]
  ENDING-FILE procedure-name-5 [THRU procedure-name-6]
  FILE-PROTECT
  ON file-name-1
} ...

```

The APPLY clause should be written if certain procedures not a part of IOCS are to be executed at the beginning of each reel of a file, at the end of each reel of a file, or at the end of the file, as appropriate. Each procedure-name refers to the name of a paragraph or section written in the Procedure Division. Each of the above procedures must be self-contained, both logically and physically; i. e., only the IOCS may transfer control to these procedures, and control must be returned to the IOCS when the procedure is completed. No entrance into or exit from these procedures by means of the GO TO or PERFORM verbs is permitted, except that return to the IOCS must be provided by the programmer. The statement GO TO NORMAL-IO-RETURN or GO TO ALTERNATE-IO-RETURN should be written as the final statement to be executed (see the LABEL RECORD clause in the Data Division and also Appendix A for additional information). If a given procedure exceeds a single paragraph or section, the THRU option must be written in order to properly define the first and last statements contained within the procedure.

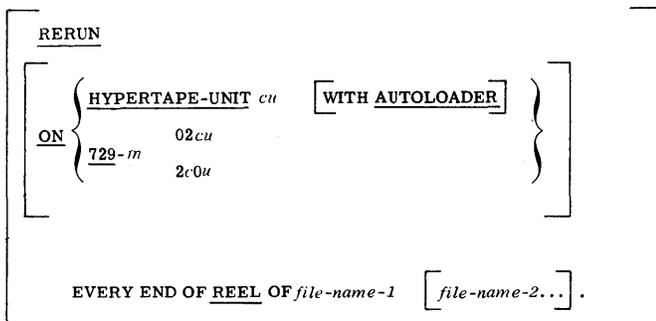
A BEGINNING-REEL entry is required only if a special procedure not a part of IOCS is to be executed at the beginning of each reel of a file. For example, when nonstandard header labels are used, procedure-name-1 (THRU procedure-name-2) must define a procedure for label checking.

An ENDING-REEL entry is required only when a special procedure not a part of IOCS is to be executed at the end of each reel of a file. It will be executed on every reel of an input file and on every reel of an output file except the last. For example,

when trailer labels are not used, a procedure must be written to determine at the end of each reel of an input file whether an end-of-reel or end-of-file condition exists. Procedure-name-3 (THRU procedure-name-4) defines such a procedure. It should be remembered that, on the 705 III, the ENDING-REEL procedure will be executed before the last data record or block of records has been processed; hence, data record counts should not be taken. This restriction does not apply to the 7080.

A procedure to be executed at the end of the last reel of a file is reached by the ENDING-FILE entry. On both the 705 III and 7080, the ENDING-FILE procedure will be executed only after all data records have been processed. If specified for an input file, procedure-name-5 (THRU procedure-name-6) will be executed prior to the IOCS transfer of control to the AT END statement. If specified for an output file, the procedure will be executed on a CLOSE statement.

The FILE-PROTECT option is valid only when applied to a Hypertape file. When used, an indicator will be set on the Hypertape cartridge at each end of reel or end of file to prevent subsequent use of that cartridge for output.



The 705 II or 7080 IOCS provides for resuming a program that has been interrupted before completion. In order to restart the program, a recording of the complete status of the machine is taken at intervals known as "checkpoints." Checkpoint records are written on the RERUN file, that is, on either the reels of a source program output file or else on a reserved tape unit.

When the ON option is utilized to assign the RERUN file to a separate tape unit, checkpoints will be taken at the end of each reel of each source program input and/or output file-name written after REEL. A File Description entry named RERUN is required in order to specify label requirements (and also DENSITY on the 7080). However, no trailer labels will be allowed under this option. (See the LABEL RECORDS clause in the Data Division and the APPLY clause above.)

When the ON option is omitted, only file-name-1 may be written after REEL. In this case, file-name-1 must be a source program output file. Checkpoints will be taken at the end of each reel of file-name-1 except the last and will be written at the beginning of each alternate reel of file-name-1.

Under either option, the specification of a second tape, known as the RERUN-WORK file, is always required (see "File-Control Paragraph").

Note: If RERUN and/or RERUN-WORK is assigned to a shared Hypertape output file(s), the file(s) will be opened in IOCS Housekeeping prior to execution of the programmer's initial OPEN statement. Execution of the OPEN statement will cause the file(s) to be re-opened. If a BEGINNING-REEL routine has been specified for either of these files, it will be executed twice.

DATA DIVISION

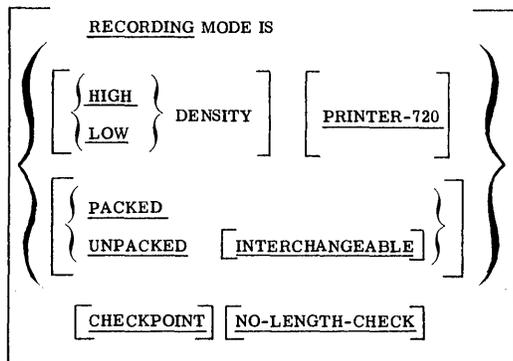
The following discussion is concerned with the limitations and specifications of the 705/7080 COBOL Data Division entries.

File Description Entries

File Description entries are the means by which files of data are described to the COBOL Processor. The programmer should be aware of certain considerations in coding such entries for the 705/7080.

Whenever a file is card input, card output, or printer output, the File Description entry consists only of the level indicator FD, the file-name, and the DATA RECORDS clause. Input/output operations for these file types may be coded in COBOL language for all models of the 705/7080.

Tape input/output operations may be coded in COBOL language only for the 705 III and 7080; they must be coded in Autocoder for the 705 I and 705 II. The following entry format specifications are directed toward tape files when the 705 III or 7080 IOCS is used.



When the OBJECT-COMPUTER is the 7080, HIGH or LOW DENSITY must be specified for each 729 tape file.

When an output file is to contain records prepared for off-line printing on a Model 720 or 730 printer, PRINTER-720 should be specified only if the final character in a block of records is a record mark. When PRINTER-720 is specified, the final character in a block of records, regardless of its nature, is deleted. Such files must be written in LOW DENSITY and cannot re-enter the system as input. Therefore, the remaining options are inapplicable.

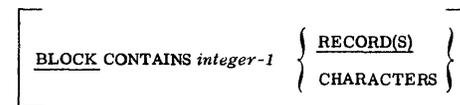
PACKED or UNPACKED is relevant only for Hypertape files attached to a high-speed channel. PACKED is used if data compression is desired. UNPACKED is used if data compression is to be disallowed. If the option is omitted, PACKED will be assumed.

INTERCHANGEABLE (relevant only for Hypertape files) is used for files containing variable-length

records in the inter-machine Hypertape format. If this option is used, the RECORDING MODE must be UNPACKED.

When an input file contains checkpoint records, CHECKPOINT must be specified.

Every block of a variable-format tape file must begin with a five-character field, the last two characters of which specify a blocking factor of 99 records or less. The remaining three characters may be used to specify the block size. When the block size is present, it will be checked against the actual block size read into the computer. If the block size is not specified, NO-LENGTH-CHECK must be specified. Since the block size and blocking factor are automatically written onto COBOL output tapes of this type, NO-LENGTH-CHECK should never be specified for input files that were created as output from a previous COBOL run. (See the Tape Format publication.)



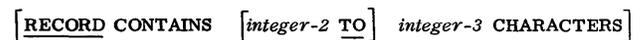
This clause is required unless a block contains one and only one complete record.

The following rules apply to fixed-length blocks:

1. When RECORD(S) is written, integer-1 represents the exact number of records in each block.
2. When CHARACTERS is written or when RECORD(S) is omitted, integer-1 represents the exact number of characters in each block.

The following rules apply to variable-length blocks:

1. When CHARACTERS is written or when RECORD(S) is omitted, integer-1 represents the number of characters in the largest block, including the five-character blocking factor field (see the Tape Format publication).
2. When it is desired to control the number of records in a block of an output file, RECORD(S) may be specified, in which case integer-1 represents the fixed number of variable-length records in each block. The RECORD clause must also be written and both integer-2 and integer-3 must be specified.



This clause is always required when describing variable-length records. Integer-2 represents the number of characters in the smallest record, and integer-3 represents the number of characters in the largest record.

If desired, the RECORD clause may be written when describing fixed-length records. In this case, integer-2 must not be written, and integer-3 represents the exact number of characters in each record.

It should be remembered that each IOCS-controlled record must start in a 0 or 5 memory

LABEL RECORD(S) } IS }
 } ARE }

{
 OMITTED
 STANDARD
 BEGINNING-TAPE-LABEL } NO-TAPEMARK }
 ENDING-TAPE-LABEL }

location, and, with the exception of unblocked, fixed-length records, must be a multiple of five characters in length and must end with a record mark (see the discussions of SYNCHRONIZED and PICTURE). The LABEL RECORDS clause is required only when the file is assigned to tape. When OMITTED is specified for an input file, the programmer must determine, as part of the end-of-reel routine for each reel, whether an end-of-reel or end-of-file condition exists. It should be remembered that on the 705 III the last record or block of records may not have been processed when the tape mark was sensed and therefore cannot be relied upon to provide end-of-file detection. (See the APPLY clause in "I-O-Control Paragraph.") STANDARD implies that both header and trailer labels exist and are in the format described in the appropriate IOCS publication. If this clause applies to a Hypertape file, the only valid option is STANDARD. Additional operations may be performed on header and/or trailer labels by defining the labels in Record Description entries with the level 01 fixed data-names BEGINNING-TAPE-LABEL and/or ENDING-TAPE-LABEL for 729 files, and BEGINNING-HYPERTAPE-LABEL and/or ENDING-HYPERTAPE-LABEL for files assigned to Hypertape drives. (See the APPLY clause in "I-O-Control Paragraph.")

The remaining information describing the LABEL RECORDS clause applies only to 729 labels.

Nonstandard label records must be no longer than 80 characters. IOCS will read and write these labels into and from the IOCS label area. The header label named BEGINNING-TAPE-LABEL or the trailer label named ENDING-TAPE-LABEL or both should be specified under the LABEL RECORDS clause. A tape mark will separate trailer label records from data records. Header label records will also be separated from data records by a tape mark unless NO-TAPEMARK is specified. Record Description entries with the level 01 fixed data-names BEGINNING-TAPE-LABEL and/or ENDING-TAPE-LABEL must be provided. (See the APPLY clause in the "I-O-Control Paragraph.")

When BEGINNING-TAPE-LABEL and/or ENDING-TAPE-LABEL appear as label record names of two or more files, they must be qualified by their associated file-names whenever referenced in the Procedure Division.

No trailer labels will be allowed on RERUN-WORK or RERUN files when they are assigned to separate tape units. (See the discussions of FILE-

CONTROL and I-O-CONTROL in the Environment Division.)

Header and trailer labels other than the types described above must be named under the DATA RECORDS clause and must be completely programmed as data records. In this case, OMITTED must be specified, and the names BEGINNING-TAPE-LABEL and ENDING-TAPE-LABEL are disallowed. In addition, the automatic end-of-tape indications must be circumvented.

{
 VALUE OF } ID }
 IDENTIFICATION } IS 'up to 10 alphanumeric characters
 excluding special characters'
 PURGE-CYCLE IS up to 3 digits }

The VALUE clause must be written when label records are STANDARD. The PURGE-CYCLE option can be written for output files only (see Appendix A).

Record Description Entries

Record Description entries are the means by which items of data are described to the COBOL Processor. The programmer should be aware of certain considerations in coding such entries for the 705/7080. The remainder of this discussion is devoted to specific limitations and interpretations.

{
 SYNCHRONIZED } LEFT }
 RIGHT }

SYNCHRONIZED RIGHT has no significance to the 705/7080 and will be ignored by the Processor.

SYNCHRONIZED LEFT causes data to be stored beginning in a 0 to 5 memory location. This clause must be written in the File Section when the tape label or data records described are under control of the 705 III or 7080 IOCS (see the discussion of PICTURE). It may also be used in the Working-Storage and Constant Sections to position data suitably for high-speed transmission. The size of the item must be a multiple of five characters in length and, on the 705 II, must also end in a record mark (see PICTURE).

On the 7080, a level 77 entry containing a SYNCHRONIZED LEFT clause, or a level 01 entry that becomes a left-synchronized as a result of a lower-level entry containing a SYNCHRONIZED LEFT clause, will be assigned memory locations beginning in a 0 position. If the item is a multiple of ten characters in length and ends in a record mark (see the discussion of PICTURE), the ten-character transmission feature will be available.

The SYNCHRONIZED clause should be written only in an elementary item Record Description entry; it has no effect on the efficiency of arithmetic operations.

PICTURE IS any allowable combination of COBOL characters and symbols

The PICTURE symbols J and K have special uses on the 705/7080. Both symbols may be required when specifying records in the File Section, and both may be used wherever desired in the Working-Storage and Constant Sections.

Data records that are to be read from or written on tape in blocked format (using the IOCS) must be a multiple of five characters, must be left-synchronized, and must end in a record mark. Variable-length records must begin with a five-position field containing the data record length (see Tape Format publication).

Note: No more than five digits can be contained within the parentheses after the significant character in a PICTURE clause.

An entry consisting of only a level number, data-name, or FILLER, in combination with the PICTURE IS J clause, will cause the Processor automatically to insert a record mark in the next 4 or 9 memory position. This clause may also be written in Record Description entries in the Working-Storage and Constant Sections when it is desired to take advantage of high-speed transmission, provided that the data is a multiple of five characters and is left-synchronized.

Data records that are to be written on the on-line card punch or 717 Printer must be followed by a group mark. An entry consisting of only a level number, data-name, or FILLER, in combination with the PICTURE IS K clause, will cause the Processor automatically to insert one group mark. Normally, this clause would be written only in the File Section, but under certain circumstances it may be desirable to write it in either the Working-Storage or Constant Sections.

The following interpretations and restrictions are peculiar to 705/7080 Record Description pictorials.

1. Numeric items:
 - a. When the assumed decimal point (V) appears in a PICTURE, the item may not contain more than 99 positions on either side of the decimal point.
 - b. When a NUMERIC item appears in an arithmetic expression, it is limited to 18 numeric characters.
2. Report items:
 - a. A report item may not contain more than 18 numeric characters.
 - b. An actual decimal point (.) should not appear as the rightmost character in a PICTURE. It will be considered a period.
 - c. The letter Z and an * may not appear in the same PICTURE.
 - d. The letter Z or an * may be preceded only by another Z or *, respectively, a

- e. The letter Z, a \$, or an * may appear to the right of an actual decimal point (.) or an assumed decimal point (V) only if all numeric character positions are represented by Z, \$, or *, respectively.
- f. The letter Z or an * may not appear in the same PICTURE with more than one \$ (floating dollar sign).
- g. When a PICTURE contains one or more commas, the Processor generates an edited field with a comma after every third character to the left of the decimal point. If there is no decimal point, the Processor generates an edited field with a comma after every third character to the left, assuming the field to be an integer. In either case, the leftmost character cannot be a comma. If commas are not specified in the PICTURE, commas will not be generated.

For example, a PICTURE containing commas should be written as illustrated below.

With a decimal point:

Z,ZZZ,ZZZ,ZZZ.ZZ- or ZZZ,ZZZ,
ZZZ.ZZ-

With no decimal point:

Z,ZZZ,ZZZ,ZZZ, - or ZZZ,
ZZZ,ZZZ-

CLASS IS

ALPHABETIC
NUMERIC
ALPHANUMERIC
AN

When the CLASS of an item is NUMERIC, an operational sign, if any, will appear over the units position of the item as a 12-zone (+) or 11-zone (-). The sign of the item will be considered always present if the SIGNED clause or the "S" character in PICTURE is written, and never present otherwise. In order to produce efficient coding for computations, NUMERIC items on the 705/7080 should be signed in the units position and left-protected. Left-protection occurs whenever the rightmost character of the preceding item is not an unsigned digit. If the preceding item, as described by its Record Description entry, does not always contain such a character in the rightmost position, the programmer should consider rearranging the data or inserting a blank character. Note that an ALPHABETIC item always affords left-protection to the item on its right, while an ALPHANUMERIC item may or may not do so.

Elementary non-report ALPHANUMERIC items, which do not always contain an alphabetic character, i. e., whose PICTURE contains no A's, will be treated as unsigned integers when referenced by an arithmetic verb or when associated with NUMERIC or report items, as in a MOVE operation. A warning message will be produced by the Processor.

Alphanumeric items of the report type will never contain operational signs over the units position. However, when an arithmetic verb references a report item, an object-time scan of the item will be provided which will extract the numeric contents of the item, taking into account any decimal positions and sign indication. Thus computations are allowed on report items. Similarly, a report item will be moved arithmetically to a NUMERIC item, another report item, or a non-report item, as described above.

[USAGE IS {COMPUTATIONAL
DISPLAY}]

Since all data represented within the 705/7080 is in external format, the DISPLAY option of the USAGE clause is not meaningful and will be ignored by the Processor. However, if an item is specified as being COMPUTATIONAL, it must be a NUMERIC item.

[VALUE IS literal]

The VALUE clause has no meaning in a Record Description entry that describes a report item, and it cannot be used to specify an initial value. This restriction also implies that a level 88 condition-name entry cannot be associated with a report item.

When the VALUE clause appears in an entry at the group level, the size of the literal must be equal to the size of the group item, i. e., the total size of the lower-level items within the group. The group area will be initialized without consideration for the individual elementary or group items contained within this group. The decimal point and operational sign must not be written in a numeric literal at the group level. A figurative constant may be specified at the group level only if the SIZE clause is also written in the entry.

[REDEFINES data-name]

The entries redefining an area must immediately follow the entries originally describing the same area. Multiple redefinitions are possible, however, 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 a REDEFINES clause, unless the latter entry is at the 01 level, in which case one such lower level redefinition is permitted at a time.

PROCEDURE DIVISION

The following discussion contains specifications concerning 705/7080 COBOL Procedure Division entries. The first part of the discussion contains specifications not included in the COBOL general information publication, whereas the second part discusses Processor limitations and interpretations pertaining to specific verb formats.

Additional Specifications

Data items that appear in relational expressions may not contain more than 255 character positions.

When a data-name written in a Data Division Record Description entry is not unique, it must be qualified by one or more additional names to make it unique, whenever it is referenced in the Procedure Division. A file-name written in a Data Division File Description entry is the highest level name that may be used as a qualifier. In addition a file-name that is RENAMING another file-name in the Environment Division FILE-CONTROL paragraph may be used as a qualifier for an associated Data Division data-name.

When the OCCURS clause is written in a Data Division Record Description entry, the name of the item can be referenced in the Procedure Division by means of subscripting only. In addition, any item that is subordinate to an item described by the OCCURS clause must also be referenced in the Procedure Division by means of subscripting.

When the OBJECT-COMPUTER is a 705 I or 705 II, the OPEN, READ, WRITE, and CLOSE verbs may only be written in conjunction with card input, card output, or printer output files.

The Class Condition IF data-name NUMERIC (ALPHABETIC) . . . may be used only to test an ALPHANUMERIC item at object time to determine whether it is wholly numeric or wholly alphabetic in content. An item will be considered NUMERIC if it consists of unsigned digits only or if all character positions except the units position are unsigned digits and the units position contains one of the digits 0 through 9 with a 12-zone (+) or 11-zone (-). It is therefore possible for a one-character data item to be considered NUMERIC or ALPHABETIC depending upon which test is made.

Conditional statements may be written in three basic forms, as shown in Figure 1.

Statement-1 can be only a simple or compound imperative statement.

Statement-2, statement-3, and statement-7 can be either imperative or conditional. If conditional, they can, in turn, contain conditional statements in arbitrary depth. When the statements are conditional, the conditions within them are "nested."

Statement-4 must be a READ statement, statement-5 must be an arithmetic statement, and statement-6 can be only a simple or compound imperative statement. Statement-8 followed by statement-9 (to which the previous paragraph applies since it is conditional) is an illustration of an imperative statement followed by a conditional statement. This is logically equivalent to statement-8 followed by a period followed by statement-9 beginning a new sentence. Option 3 in its entirety may be substituted for statement-2 and/or statement-3 under option 2.

It is important to note that, when conditional statements are nested, an ELSE or OTHERWISE is always associated with the IF, READ, or arithmetic verb with ON SIZE ERROR that most closely precedes it. An ELSE or OTHERWISE must be explicitly written for every conditional statement within a sentence. However, the phrase ELSE (OTHERWISE) NEXT SENTENCE may be eliminated if and only if the phrase immediately precedes the period ending a sentence.

Procedure Division Verbs

The remainder of this chapter discusses specific verb format limitations and interpretations that the programmer should understand before he incorporates such verbs into his 705/7080 source program.

ACCEPT *data-name* [**FROM** *mnemonic-name*]

The standard ACCEPT device on the 705/7080 is the card reader. When the FROM option is not written, the card reader address 0100 will be assumed by the Processor. (See the "Special-Names Paragraph.")

Since the AT END clause may only be written in conjunction with the READ verb, the programmer must provide his own end-of-file procedures when using ACCEPT.

<u>Option 1</u>	IF <i>conditional expression</i> statement-1.
<u>Option 2</u>	IF <i>conditional expression</i> { <i>statement-2</i> } { OTHERWISE } { <i>statement-3</i> } { NEXT SENTENCE } { ELSE } { NEXT SENTENCE }
<u>Option 3</u>	{ { <i>statement-4</i> AT END } { <i>statement-6</i> } { OTHERWISE } { <i>statement-7</i> } } { { <i>statement-5</i> ON SIZE ERROR } { NEXT SENTENCE } { ELSE } { NEXT SENTENCE } } <i>any imperative statement-8 followed by any conditional statement-9</i>

Figure 1. Conditional Statement Options

DISPLAY $\left\{ \begin{array}{l} \text{data-name-1} \\ \text{literal-1} \end{array} \right\} \left[\left[\begin{array}{l} \text{data-name-2} \\ \text{literal-2} \end{array} \right] \dots \right] \\ \left[\text{UPON mnemonic-name} \right]$

The standard DISPLAY device on the 705/7080 is the console typewriter. When the UPON option is not written, the typewriter address 0500 will be assumed by the Processor. (See "Special-Names Paragraph.")

WRITE *record-name* $\left[\text{FROM } \textit{area-name} \right]$

When record-name is associated with more than one output file, either explicitly by means of the DATA RECORDS clause in the File Description entry or implicitly by means of the FILE-CONTROL paragraph RENAMING option, it must always be qualified by the appropriate file-name.

The FROM option implies that the area-name will be moved directly to the file output area associated with the record-name. The record-name area will be left unchanged.

CLOSE *file-name-1* $\left[\text{WITH LOCK} \right] \left[\textit{file-name-2} \dots \right]$

At the present time, the LOCK option is the only available option of the CLOSE verb permitted for use with the 705 III/7080 IOCS. If LOCK is specified, all reels of a file will be made unavailable for reading or writing when the CLOSE verb is executed and when intermediate reels of a file reach the physical end of tape. A subsequent attempt to OPEN a locked file will be rejected. If any CLOSE verb referencing a file uses the LOCK option, all CLOSE verbs for that file will be interpreted by the Processor as having used the LOCK option.

When a reel or a series of reels is to contain both input and output files within the course of a program, i. e., when an output file created by the program is to be read as an input file or when an input file used by the program is to be written over as an output file, the LOCK option is appropriate only when closing the last file. In addition, each file must have a unique file-name.

ENTER *language-name*

For all models of the 705 and 7080, the programmer may code in Autocoder language. In fact, all input/output operations for tape files must be coded in

Autocoder on the 705 I and 705 II. In Autocoder the programmer has access to the FORTRAN language, the Report/File and Decision languages, and all other special languages normally available. Autocoder entries may be placed anywhere within the Procedure Division, provided each group of entries is preceded by the statement ENTER AUTOCODER and terminated by a COBOL paragraph whose first statement is ENTER COBOL. When the Procedure Division ends with a group of Autocoder entries, a final paragraph-name followed by the sentence ENTER COBOL must be written. The names AUTOCODER and COBOL are the only language-names permitted with the ENTER verb.

A paragraph-name associated with an ENTER AUTOCODER statement may be referenced directly in COBOL language by the GO TO and PERFORM verbs. A paragraph-name associated with an ENTER COBOL statement may only be referenced indirectly in Autocoder language. An Autocoder tag should be assigned to the COBOL paragraph-name in the SPECIAL-NAMES paragraph of the Environment Division.

The 705/7080 Auxiliary Storage Unit (ASU) settings are determined by the COBOL Processor. ASU 01 through ASU 05 are set to one to five positions, respectively. ASU 13 is set to ten and ASU 14 is set to four. These settings are fixed in each program. ASU 15 and the accumulator are used as variable-length storage units.

The settings of ASU 06 through ASU 12 normally vary from one source program to another. These ASUs will be set by the Processor according to the sizes of items most frequently referenced in the Procedure Division. However, it is possible to preset any of these ASUs by modifying the Autocoder Communication Word prior to compiling the COBOL source program. The Processor will honor such settings and compile accordingly.

When a programmer wants to write in Autocoder language by means of the ENTER AUTOCODER statement, he may wish to alter the settings of certain ASUs for that phase of the program. This is permissible as long as the Autocoder ASU macro-instruction is never written and as long as those ASUs that have been altered by Autocoder instructions are reset by the programmer before returning to the COBOL language.

The required settings of ASUs 01 through 05 and 13 and 14 are known at the time the program is being written. They may be reset accordingly. However, the settings of ASU 06 through ASU 12 are not available until the program has been compiled. In order for the programmer to be able to reset any of these ASUs without prior knowledge of their ultimate settings, specific Autocoder tags have been assigned to each setting. BLASU06 through BLASU12 may be written as the operands of Autocoder SET instructions in order to reset the appropriate ASUs. The numeric columns (21 and 22) of the Autocoder coding sheet should contain a number from 06 to 12 that identifies the ASU involved.

STOP $\left\{ \begin{array}{l} \textit{literal} \\ \textit{RUN} \end{array} \right\}$

It is recommended that STOP RUN be written in every source program to indicate the end of the program.

On the 705 III or 7080, STOP RUN initiates IOCS end-of-job procedures (such as closing the check-point tape) before the final halt is reached.

On all models of the 705/7080, the message END OF RUN will be displayed on the console typewriter and a STOP 9999 followed by a transfer back to the stop will be produced by the Processor.

Add Corresponding Option

The CORRESPONDING option of the ADD verb allows the programmer to specify the addition of corresponding items in one operation in a manner similar to MOVE CORRESPONDING.

The general form of ADD CORRESPONDING is:

ADD CORRESPONDING *data-name-1* TO *data-name-2*

$\left[\begin{array}{l} \textit{ROUNDED} \\ \textit{ON SIZE ERROR any imperative statement} \end{array} \right]$

Numeric elementary items within *data-name-1* are added to numeric elementary items with matching names in *data-name-2*. *data-name-1* and *data-name-2* must be nonelementary items. The rules stated for the simple ADD verb apply to each pair of items in the ADD CORRESPONDING option.

Only the initial description of items in *data-name-1* and *data-name-2* is considered in the implementation of the ADD CORRESPONDING option. That is, where a REDEFINES clause has been used within *data-name-1* or *data-name-2*, the description of the data contained within the REDEFINES clause is ignored by ADD CORRESPONDING. The ROUNDED option and the SIZE ERROR option may be used with ADD CORRESPONDING. For a detailed description of these options, the reader is referred to the COBOL general information publication.

Note: The SIZE ERROR test is made only after the completion of all the add operations. If any of the additions produced a SIZE ERROR, the resultant field for that add remains unchanged and the "any imperative statement" is executed.

To illustrate the use of the ADD CORRESPONDING option, assume that the programmer wishes to add items from a work area named RECEIPTS to corresponding items in an area designated STOCK-ON-HAND. The programmer could write this statement:

ADD CORRESPONDING RECEIPTS TO
STOCK-ON-HAND

Figure 2 shows what will result from this statement. Note that noncorresponding items in the STOCK-ON-HAND area are not affected.

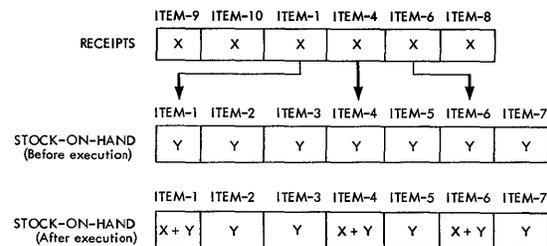


Figure 2. Add Corresponding

Subtract Corresponding Option

The CORRESPONDING option of the SUBTRACT verb functions in the same way as the CORRESPONDING option of the ADD verb. The general form of SUBTRACT CORRESPONDING is:

SUBTRACT CORRESPONDING *data-name-1*

$\left[\begin{array}{l} \textit{FROM data-name-2} \\ \textit{ROUNDED} \\ \textit{ON SIZE ERROR any imperative statement} \end{array} \right]$

APPENDIX A. TAPE LABELS AND SPECIALIZED ROUTINES

LABEL CONTROL CARDS

Label control cards are used whenever tape files with standard header labels are to be written or read. They may be read from either the card reader or the first reel of a source program output file. (See "File-Control Paragraph.")

The Date control card must be the first read in. It is required if any file has standard headers and is used in conjunction with the purge cycle to check the erasability of output tapes.

A File Serial Header control card must be provided for every input file with standard header labels. It is used to insure that the correct reel has been mounted.

An End control card is required as the last of the label control cards.

It should be noted that information from the label created for output files will be typed automatically after the label is placed on tape (IOCS Message 10298). The information is in the following order: tape serial number, file serial number, reel sequence number, file identification, creation date, purge cycle.

The formats of the cards are described below.

Date Control Card

<u>Columns</u>	<u>Information</u>
1-6	Program identification (first six characters of the name written in PROGRAM-ID paragraph)
7	Blank
8-9	Year (e.g., 64)
10-12	Calendar day (001 to 366)

File Serial Header Control Card

<u>Columns</u>	<u>Information</u>
1-10	File identification (same as that entered in the File Description entry for this file)
11	Blank
12-15/16	File serial number of all reels of an input file (see Tape Format publication). Four digits for 729 tape files; five digits for Hypertape files.

705 III End Control Card

<u>Columns</u>	<u>Information</u>
1-6	Program identification (first six characters of the name written in PROGRAM-ID paragraph)
7	Blank
8-10	The characters END

7080 End Control Card

<u>Columns</u>	<u>Information</u>
1-6	Program identification (first six characters of the name written in PROGRAM-ID paragraph)
7-15	Blank
16-18	The characters END

LABELS AND SPECIAL PROCESSING CONSIDERATIONS

When nonstandard labels (729 files only) are used, or when the programmer wishes to put special information in the standard label, the label format should be placed in Record Description entries with the level 01 fixed data-names BEGINNING-TAPE-LABEL and/or ENDING-TAPE-LABEL (or BEGINNING-HYPERTAPE-LABEL and/or ENDING-HYPERTAPE-LABEL for Hypertape files). The label in the Record Description entry may be used for either label checking or writing.

The programmer should note that IOCS will automatically do the following:

1. Check the file serial number and reel sequence number on standard input header labels.
2. Check to see if the output file can be written on, and place a label identification, file serial number, reel sequence number, creation date, purge cycle, file identification, and record format in standard output header labels.
3. Check the standard input trailer label termination code to determine whether end of reel or end of file has been reached.
4. Insert the record count and skip count in standard output trailer labels.

Checking the Record Count

For standard labels, the 7080 IOCS automatically compares the number of tape records read with the number specified in the trailer, prior to transferring to the ENDING-REEL or ENDING-FILE routine. If the counts are not equal, a message is typed and the programmer has the option of continuing or restarting from the last checkpoint.

If standard labels are not used on the 7080, and the programmer desires to compare the tape record count or skip count fields of the trailer label of an input file with those accumulated by IOCS, he must use a special ENDING-REEL or ENDING-FILE routine. Such a routine is required for the checking on the 705 III. In addition to coding the returns

correctly (see below), the programmer must ENTER AUTOCODER for each of the routines. The counts accumulated are referred to by the following tags:

	<u>Tape-record Count</u>	<u>Noise/Skip Count</u>
705 III	IOWRKFTREC	IOWRKFTSKP
7080 (729)	CSF0003022	CSF000302
7080 (Hypertape)	CSFCSHYFRC	CSFCSHYTSK

Both fields are signed numeric: Tape-record Count is six positions and Noise/Skip Count is two.

Coding Special Routines When Using the 7080 IOCS

The 7080 IOCS normally operates in the interrupt program. This results in a shift of the Starting Point Counter from bank 0 (accumulator) to bank 3 (CASUs) when the interrupt program is entered. Since 7080 IOCS does not leave the interrupt program upon entering a special routine, the Processor will generate a BLIXT macro-instruction that will shift the Starting Point Counter to bank 0 and save the contents of bank 0 and bank 1 before executing the special routine.

Immediately before each transfer to NORMAL or ALTERNATE-IO-RETURN, the Processor generates a BLIXT macro-instruction that restores the saved contents of bank 0 and bank 1 and shifts the Starting Point Counter to bank 3.

USE OF SPECIALIZED ROUTINES

Special header routines (BEGINNING-REEL) or trailer routines (ENDING-REEL or ENDING-FILE) are coded in the Procedure Division with a return to the IOCS at either NORMAL-IO-RETURN or ALTERNATE-IO-RETURN. The specifications that follow should be used to determine which return point to use.

It is important to note that no tape input/output commands can be executed during a BEGINNING-REEL, ENDING-REEL, or ENDING-FILE routine.

Input Files

Standard Labels

Beginning-Reel Routine: This routine is executed upon opening a file and at the beginning of alternate reels, and can be used for additional checking. If the reel is proper, return must be made at NORMAL-IO-RETURN. If the reel is improper, processing should be stopped and a message typed directing the operator to mount a new reel; return must be made at ALTERNATE-IO-RETURN. (The routine will then be repeated.)

Ending-Reel Routine: This routine is executed when a standard end-of-reel trailer is encountered. It is used only for extra checking. Return must be made at NORMAL-IO-RETURN.

Ending-File Routine: This routine is executed when a standard end-of-file trailer is encountered (at end of last reel). It is used only for extra checking. Return must be made at NORMAL-IO-RETURN.

Nonstandard Labels (729 Files Only)

Beginning-Reel Routine: This routine is required to check header labels, and is executed upon opening a file and at the beginning of alternate reels. If it has been determined that the reel is the proper one, return must be made at NORMAL-IO-RETURN. Otherwise, operator action should be specified by a message and return made at ALTERNATE-IO-RETURN. The computer then stops processing and displays message 30291; the operator should follow the instructions associated with the message number.

Ending-Reel Routine: This routine is executed when a tape mark and nonstandard trailer are encountered, and is required for determining whether an end-of-reel condition exists. If an end-of-reel condition does exist, return must be made at NORMAL-IO-RETURN; if not, return must be made at ALTERNATE-IO-RETURN. If the return was made at ALTERNATE-IO-RETURN, the system assumes that an end-of-file condition exists; if an ENDING-FILE routine exists, it will be executed.

Ending-File Routine: This routine, which is optional, is executed after the ENDING-REEL routine has determined that no end-of-reel condition exists; it then continues label checking. Return is made at NORMAL-IO-RETURN if an end-of-file condition exists; otherwise, return is made at ALTERNATE-IO-RETURN and the computer stops processing and displays message 30293. The operator should follow the instructions associated with the message number.

No Labels (729 Files Only)

Beginning-Reel Routine: This routine, which is optional, is executed upon opening a file and at the beginning of alternate reels. Return is made at NORMAL-IO-RETURN.

Ending-Reel Routine: This routine is executed when a tape mark has been reached. It is required to determine if there is an end-of-reel condition. If the end of reel has been reached, return must be made at NORMAL-IO-RETURN. If not, return must be made at ALTERNATE-IO-RETURN. IOCS assumes that the end of file has been encountered if return was made at ALTERNATE-IO-RETURN and will transfer to the ENDING-FILE routine if one has been specified.

Ending-File Routine: This routine is executed when the ENDING-REEL routine has returned at ALTERNATE-IO-RETURN. It continues the checking that was begun by the ENDING-REEL routine with return at NORMAL-IO-RETURN.

Output Files

Standard Labels

Beginning-Reel Routine: This routine, which is optional, is executed upon opening a file and at the beginning of alternate reels. It is used to add information to header labels. Return is made at NORMAL-IO-RETURN.

Ending-Reel Routine: This routine, which is optional, is executed at the end of reel (reflective spot). It is used to add information to trailer labels. Return is made at NORMAL-IO-RETURN unless the programmer wants to ignore the end-of-reel condition and continue writing, in which case ALTERNATE-IO-RETURN should be specified. When ALTERNATE-IO-RETURN is specified, IOCS will branch to the ENDING-REEL routine for each subsequent tape record placed on the reel until NORMAL-IO-RETURN is specified. When NORMAL-IO-RETURN is specified, a tape mark, a trailer, and a final tape mark will be written following the last record on the reel.

Ending-File Routine: This routine, which is optional, is executed upon closing a file. It is used to add information to trailer labels. Return should be made at NORMAL-IO-RETURN.

Nonstandard Labels (729 Files Only)

Beginning-Reel Routine: This routine is required to check the header label of the output tape and to insure that it is erasable; if so, it creates a new header in the label area. If the tape is erasable, return should be at NORMAL-IO-RETURN; if not, operator action should be specified by a message and return made at ALTERNATE-IO-RETURN. After ALTERNATE-IO-RETURN, the computer automatically stops processing and displays message 30290; the operator should follow the instructions specified. The procedure will then be repeated.

Ending-Reel Routine: This routine is required to create a trailer label that includes end-of-reel indication. Return is made at NORMAL-IO-RETURN unless the programmer chooses to ignore the end-of-reel condition and continue writing, in which case he specifies ALTERNATE-IO-RETURN. When ALTERNATE-IO-RETURN is specified, IOCS branches to the ENDING-REEL routine for every subsequent tape record placed on the file until return at NORMAL-IO-RETURN is specified. On a NORMAL-IO-RETURN, a tape mark, a trailer, and a final tape mark will be written following the last record on the reel.

Ending-File Routine: This routine is required to create an end-of-file trailer. It is executed upon closing a file. Return is at NORMAL-IO-RETURN.

No Labels (729 Files Only)

Beginning-Reel Routine: This routine, which is optional, is executed upon opening a file and at the beginning of alternate reels. Return is at NORMAL-IO-RETURN.

Ending-Reel Routine: This routine, which is optional, is executed at the end of reel (reflective spot). Return is made at NORMAL-IO-RETURN unless the end-of-reel condition is to be ignored in order to write additional records on the reel, in which case ALTERNATE-IO-RETURN should be specified. When ALTERNATE-IO-RETURN is specified, IOCS will go to ENDING-REEL routine for every subsequent tape record placed on the file until return at NORMAL-IO-RETURN is specified. After NORMAL-IO-RETURN, a tape mark, a trailer, and a final tape mark will be written following the last record on the reel.

Ending-File Routine: This routine, which is optional, is executed upon closing a file. Return is made at NORMAL-IO-RETURN.

The following list of input/output assumptions and restrictions should be studied by programmers familiar with IOCS.

1. An OPEN command primes all input (buffer) areas.
2. The record-length-checking feature of IOCS will always be used when reading fixed-length records or variable-length records that contain the block size, and when writing any record.
3. No provision for accumulating hash totals is included.
4. Input headers will not be typed unless they are erroneous.
5. The V label option (a special case of standard labels where the EOF or EOR indicator in the trailer is ignored) is not available.
6. The first three characters of the file identification for standard labels should not be numeric since the Cycle Checking option is not allowed.
7. Tape files that have been assigned to the same base tape will share input/output buffer areas if they are all input or all output. Two such files should never be open at the same time.

705 III ONLY

1. The ASUSAVE and ASURESTORE options are not used.

2. The Checkpoint routine records all of memory, the contents of the accumulator, ASU 06 to ASU 15, and the Alteration switch settings.

3. The INITIATE mode is assigned to all files with only one data record per block, except input files with only one input area.

4. The STACKING mode is assigned to input files with only one data record per block and only one input area, and to all blocked files (files with more than one data record per block).

5. For blocked files, the READ command includes a test for ready channel. Input/output operations will be initiated on each channel that is in ready status.

7080 ONLY

1. The Checkpoint routine records all of memory, the contents of central storage, and Alteration switch settings.

2. The INITIATE mode is assigned to all files using only one input/output area.

3. The STACKING mode is assigned to all files using more than one input/output area.

4. Mode 2 end of reel will be specified for all files.

05 specifies the Input/Output Housekeeping routines that are associated with the features specified for the IOCS. The entry of a 1 in the numerical column indicates that this IOCS (including Housekeeping) is to be used with COBOL programs that may not contain the OPEN ALL FILES command.

MACRO-INSTRUCTIONS USED IN AN IOCS ASSEMBLY

The following pages describe in detail the formats and contents of the macro-instructions used in creating the preassembled IOCS.

IODEF (Define IOCS)

Function

The function of IODEF is to define the sections of the IOCS that are to be included in this assembly of an IOCS.

Instruction Format

Tag	Operation	Num	Operand
	IODEF		X1 X2 X3 X4 X5

where:

- X1 is IOSYS.
- X2 is IODTS.
- X3 is IOEOF.
- X4 is IOTRS or omitted.
If IOTRS, the tape labeling section of IOCS will be included in the IOCS to be assembled. Must be used if any programs specify either STANDARD or nonstandard labels. If omitted, the tape labeling section will not be included in the IOCS to be assembled.
- X5 is IOMRD or omitted.
If IOMRD, the checkpoint/restart section of IOCS will be included in the IOCS to be assembled. If omitted, the checkpoint/restart section will not be included in the IOCS to be assembled.

Coding Generated: None

Memory Requirements: None

BLSYS (Include Input/Output Control System)

Function

The function of BLSYS is to call the IOCS routines from the COBOL System Tape. BLSYS defines options for each section of IOCS specified in the macro-instruction IODEF.

Instruction Format

Tag	Operation	Num	Operand
	BLSYS		X1 X2 X3 ... X12

where:

- X1 is a one-digit number specifying the maximum number of DSs used.
- X2 is STACK.
- X3 is a one-digit number (1-8), specifying the maximum number of input/output areas used by any tape file handled by IOCS.
- X4 is CKLNG.
- X5 is F, V or M.
If F, routines that handle only fixed-length records are to be included in the IOCS to be assembled.
If V, routines that handle only variable-length records are to be included in the IOCS to be assembled.
If M, routines that handle both fixed- and variable-length records are to be included in the IOCS to be assembled.
- X6 is STANDARD, NONSTAND, or omitted.
If STANDARD, routines for handling standard header labels, as well as nonstandard header labels of 80 characters or less, will be included in the IOCS to be assembled.
If NONSTAND, routines for handling standard header labels will not be included, but those routines that handle non-standard header labels of 80 characters or less will be included.
If omitted, no routines for handling header labels will be included. This operand must be omitted if IOTRS is not an operand of the macro-instruction IODEF.
- X7 is STANDARD, NONSTAND, or omitted.
If STANDARD, routines handling standard trailer labels, as well as nonstandard trailer labels of 80 characters or less, will be included in the IOCS to be assembled. If NONSTAND, routines for handling standard trailer labels will not be included, but routines for handling nonstandard trailer labels of 80 characters or less will be included.
If omitted, no routines for handling trailer labels will be included. This operand must be omitted if IOTRS is not an operand of the macro-instruction IODEF.
- X8 is ASUANDSWS or omitted.
ASUANDSWS must be specified if the checkpoint routines are to be included in the IOCS to be assembled; i.e., if IOMRD is an operand of the macro-instruction IODEF. The operand must be omitted if IOMRD is not an operand of the macro-instruction IODEF.
- X9 is a one digit number (2 or 4) or omitted.
If the checkpoint routines are to be included, i.e., if IOMRD is an operand of the macro-instruction IODEF, operand X9 specifies the number of memory quadrants: 2 for a 40K Model III or 4 for an 80K Model III.

This operand must be omitted if IOMRD is not an operand of the macro-instruction IODEF.

X10 is omitted.

X11 is ERRORDUMP.

Routines associated with the dumping of unreadable records will be included in the IOCS to be assembled.

X12 is omitted (single lozenge required).

Note: The routines described in the above operands can be included in the IOCS assembly, if desired, even if a given object program does not use all of the features.

Coding Generated

Tag	Operation	Num	Operand
	INCL		IOaaa
	INCL		IObbb
	NAME	0	
	RCD	0	
BLSYS		xxxxx	

where:

aaa and bbb are the first and last IOCS subroutines to be included.

xxxxx is the number of memory positions used by IOCS.

Memory Requirements

The memory requirements are those specified by xxxxx under BLSYS "Coding Generated."

BLHSK (Include Input/Output Housekeeping Section of IOCS)

Function

The function of BLHSK is to call the Input/Output Housekeeping routines from the COBOL System Tape.

Instruction Format

Tag	Operation	Num	Operand
	BLHSK	X1	X2□X3□X4□X5□

where:

X1 is 1.

1 must be specified in the numeric columns since none of the COBOL object programs to be used with the IOCS being assembled may contain the command OPEN ALL FILES.

X2 is a one-digit number specifying the total number of DSs used. Operand X1 in the macro-instruction BLSYS should specify the same one-digit number.

X3 is STACK.

X4 is STANDARD or omitted.

If STANDARD, routines associated with the handling of standard header labels will be included in the Input/Output Housekeeping to be assembled. Operand X6 in the macro-instruction BLSYS should also specify STANDARD.

If omitted, routines associated with the handling of standard header labels will not be included. Operand X6 in the macro-instruction BLSYS should be omitted or specify NONSTAND.

X5 is ERRORDUMP.

Routines associated with the dumping of unreadable records will be included in the Input/Output Housekeeping to be assembled. Operand X11 in the macro-instruction BLSYS should also specify ERRORDUMP.

Note: The routines described in the above operands can be included in the Input/Output Housekeeping assembly, if desired, even if a given object program does not use all of the features.

Coding Generated

Tag	Operation	Num	Operand
	INCL		IOaaa
	INCL		IObbb
	NAME	0	
	RCD	0	
BLHSK		xxxxx	

where:

aaa and bbb are the first and last IOHSK subroutines to be included.

xxxxx is the number of memory positions used by IOHSK.

Memory Requirements

The memory requirements are those specified by xxxxx under BLHSK "Coding Generated."

ASSEMBLING AN OBJECT PROGRAM TO BE USED WITH A PREASSEMBLED IOCS

The COBOL source program to be used with a pre-assembled IOCS must include the IOCS ADDRESS and IOHSK ADDRESS entries under the OBJECT-COMPUTER paragraph of the Environment Division.

The IOCS ADDRESS entry specifies, in five-digit form, the initial and final memory locations of the IOCS, excluding the literals and the Input/Output Housekeeping section. These locations may be obtained from the preassembled IOCS listing beneath

the coding generated for the macro-instruction BLSYS. The initial location is specified in the NAME entry and the final location in the RCD entry tagged BLSYS.

The IOHSK ADDRESS entry specifies, in five-digit form, the initial and final memory locations of the Input/Output Housekeeping section. These locations may be obtained from the preassembled IOCS listing beneath the coding generated for the macro-instruction BLHSK. The initial location is specified in the NAME entry and the final location, in the RCD entry tagged BLHSK.

In all other respects, the source program must be set up in the manner described in this publication and in the COBOL general information publication.

The source program compilation is a normal COBOL compilation.

INCLUSION OF THE 705 III PREASSEMBLED IOCS WITH THE OBJECT PROGRAM

After the source program has been compiled, the program cards should be combined with the preassembled IOCS program deck as follows:

1. Load program (or object program load card)
2. Preassembled IOCS
3. Object program
4. Object program "00" card

The "00" card and load card of the preassembled IOCS should be discarded. The object program load card must be discarded if an alternate load program is used.

APPENDIX D. PREASSEMBLED IOCS FOR THE 7080

INTRODUCTION

The 7058 COBOL Processor cannot generate a 7080 Input/Output Control System. Therefore, a pre-assembled 7080 IOCS must be incorporated into object programs compiled by this processor. The 7080 COBOL Processor is capable of generating a 7080 IOCS as part of a normal compilation. However, a preassembled IOCS is also acceptable. The chief advantage to be derived from the preassembled IOCS is a substantial saving of machine time during compilation.

In order to prepare for the use of the preassembled IOCS, it is necessary to compile a version of the IOCS that is adequate to the needs of all of the programs with which it is to be used. Once this has been done, COBOL programs that are to be run using the preassembled IOCS must include the PRE-ASSEMBLED entries under the OBJECT-COMPUTER paragraph in the Environment Division. In all other respects, they are coded as if there were no pre-assembled IOCS.

The entries IOCS ADDRESS and IOHSK ADDRESS are used to indicate to the 7080 COBOL Processor the memory areas required by the IOCS. The formats and contents of these entries are given below.

ASSEMBLING AN IOCS

The IOCS being assembled must begin at memory location 0500 and must be set up so that the Input/Output Housekeeping routines will be the last section of the IOCS. This is required because, during the COBOL compilation of source programs using this IOCS, memory assignment will begin at the initial location of the Input/Output Housekeeping section.

Output areas will be assigned to overlap the Input/Output Housekeeping routines, and the remainder of the object program will be located following this section. The Tape Tables always begin at memory location 0500.

Full details of specifying the IOCS to be assembled are given in the following publications:

IBM 7080 Input/Output Control System for Use
with 729 Magnetic Tape Units, Form C28-6237
IBM 7080 Input/Output Control System for 7340
Hypertape Drives -- #7080-IO-932, Form
C28-6341

ASSEMBLING AN OBJECT PROGRAM TO BE USED WITH A PREASSEMBLED IOCS

The COBOL source program to be used with a pre-assembled IOCS must include the IOCS ADDRESS and IOHSK ADDRESS clauses (for 729 files) and/or the HYPERTAPE-IOCS ADDRESS clause (for Hypertape files) under the OBJECT-COMPUTER paragraph of the Environment Division.

The IOCS ADDRESS clause specifies, in five-digit form, the initial memory location of the IOCS, excluding the Input/Output Housekeeping section. This location should be 00500.

The HYPERTAPE-IOCS ADDRESS clause specifies, in five-digit form, the initial location of the Hypertape IOCS. This location can be obtained from the preassembled IOCS listing by referring to the Autocoder tag CSHYOPLBL.

The IOHSK ADDRESS clause specifies, in five-digit form, the initial and final memory locations of the Input/Output Housekeeping section. These locations may be obtained from the preassembled IOCS listing by referring to the Autocoder tag CSA028005 and the highest memory location used by IOCS.

In all other respects, the source program must be set up in the manner described in this publication and in the COBOL general information publication.

INCLUSION OF THE 7080 PREASSEMBLED IOCS WITH THE OBJECT PROGRAM

Following the compilation of the COBOL object program, the program cards should be combined with the preassembled 7080 IOCS in the following order:

1. Load program
2. Preassembled IOCS
3. Object program
4. Object program "00" transfer control card

The "00" card and load program of the preassembled IOCS should be discarded.

Note: The object program load cards must be discarded if an alternate load program is used.

APPENDIX E. DEFERRED ELEMENTS OF THE COBOL LANGUAGE

The following items, which appear in the COBOL general information publication, will be deferred features of COBOL for the 705/7080:

1. Environment Division.
 - a. COPY clause wherever it appears.
 - b. OBJECT-COMPUTER paragraph: ASSIGN option.
 - c. FILE-CONTROL paragraph.
 - i. OPTIONAL file-name option of SELECT.
 - ii. All features that provide for Processor assignment of tape units to a file.
2. Data Division.
 - a. COPY clause wherever it appears.
 - b. JUSTIFIED clause.
 - c. PICTURE - Report Items.
 - i. + or - on the left option.
 - ii. Floating + or - option.
 - iii. 0 and B options.
3. Procedure Division.
 - a. NO REWIND option of the CLOSE verb.
 - b. REEL option of the CLOSE verb.
 - c. Implied AT END option following the READ verb.

APPENDIX F. COMPLETE LIST OF COBOL RESERVED WORDS

The words listed below are an inherent part of the COBOL System. Their use as data or procedure names should be avoided. The list includes all words that have special meaning as either key words or optional words for any IBM COBOL Compiler.

ABOUT	CARDS
ACCEPT	CF
ADD	CH
ADDITIONAL-LABEL-INFO	CHARACTER
ADDRESS	CHARACTERS
ADDRESSES	CHECK
ADVANCING	CHECKPOINT
AFTER	CHECKPOINT-UNIT
ALL	CHECKSUM
ALL-FILES	CHECKSUM-ERROR
ALPHABETIC	CLASS
ALPHANUMERIC	CLOCK-UNITS
ALTER	CLOSE
ALTERATION-SWITCH	COBOL [-xxxx]
ALTER-LABEL-INFO	CODE
ALTERNATE	COLLATE-COMMERCIAL
ALTERNATE-IO-RETURN	COLLATE-MACHINE
ALT-SWITCH-n	COLUMN
AN	COMMUNICATION-MODE
AND	COMPUTATIONAL
APPLY	COMPUTATIONAL-1
ARE	COMPUTATIONAL-2
AREA	COMPUTE
AREAS	CONFIGURATION
AS	CONSOLE-PRINTER
ASCENDING	CONSOLE-SWITCH
ASSEMBLY-PROGRAM	CONSTANT
ASSIGN	CONTAIN
AT	CONTAINS
AUTHOR	CONTROL
AUTHORS	CONTROL-COUNT-ERROR
AUTOCODER	CONTROLS
AUTOLOADER	COPY
BCD	CORRESPONDING
BEFORE	CREATION-DATE
BEGINNING	CREATION-DAY
BEGINNING-FILE-LABEL	CREATION-YEAR
BEGINNING-HYPERTAPE-LABEL	DATA
BEGINNING-REEL	DATE-COMPILED
BEGINNING-TAPE-LABEL	DATE-WRITTEN
BINARY	DE
BIT	DECLARATIVES
BITS	DEFINE
BLANK	DENSITY
BLOCK	DEPENDING
BLOCKS	DESCENDING
BLOCK-COUNT	DETAIL
BLOCK-OR-CHECKSUM-ERROR	DIGIT
BLOCK-SEQUENCE-ERROR	DIGITS
BUFFER-OVERFLOW	DISPLAY
BY	DIVIDE
CALL	DIVIDED
CARD-PUNCH	DIVISION
CARD-READER	DOLLAR
	DOUBLE-PRECISION
	ELECTRONIC-SWITCH
	ELIMINATION
	ELSE
	EL-SWITCH-n
	END
	ENDING

ENDING-FILE	IBM-7044
ENDING-FILE-LABEL	IBM-7070
ENDING-HYPERTAPE-LABEL	IBM-7072
ENDING-REEL	IBM-7074
ENDING-TAPE-LABEL	IBM-7080
END-OF-FILE	IBM-7090
END-OF-REEL	IBM-7094
END-OF-TAPE	I-O-CONTROL
ENTER	I-O-SWITCH
ENVIRONMENT	ID
EOF-SIU	IDENTIFICATION
EQUAL	IF
EQUALS	IN
ERROR	INCLUDE
ERROR-CONDITION	INCOMPLETE-WORD-ERROR
ERROR-DUMP	INDEX-WORD-n
ERROR-FILE-ADDRESS	INDICATE
ERROR-RECORD	INITIATE
EVEN	INPUT
EVERY	INPUT-OUTPUT
EXAMINE	INSTALLATION
EXCEEDS	INTERCHANGEABLE
EXIT	INTO
EXPONENTIATED	INVALID-CHARACTER-CHECK
FD	IOCS
FILE	IOHSK
FILE-CONTROL	IORETURN
FILE-IDENTIFICATION	IORTN1
FILE-PROTECT	IORTN2
FILE-REFERENCE	IS
FILE-SERIAL-NUMBER	JUSTIFIED
FILES	KEY
FILLER	KEY-S
FILLING	KEY-1
FINAL	⋮
FIRST	⋮
FLOAT	KEY-35
FLOATING-POINT-HARDWARE	LABEL
FOOTING	LABEL-DATA
FOR	LABEL-IDENTIFIER
FORMAT	LABEL-RECORDS
FROM	LAST
GENERATE	LEADING
GIVING	LEAVING
GO	LEFT
GREATER	LESS
GROUP	LIBRARY
HASHED	LIMIT
HEADER	LIMITS
HEADER-LABEL	LINE
HEADING	LINE-COUNTER
HIGH	LINES
HIGH-VALUE	LINKAGE-MODE
HIGH-VALUES	LOAD
HYPERTAPE-IOCS	LOCATION
HYPERTAPE-UNIT	LOCK
HYPERTAPE-UNITS	LONG-LENGTH-RECORD
IBM-1401	LOW
IBM-1410	LOW-VALUE
IBM-1460	LOW-VALUES
IBM-705-1	LOWER-BOUND
IBM-705-2	LOWER-BOUNDS
IBM-705-3	MEMORY
IBM-7010	MEMORY-DUMP
IBM-7040	MEMORY-DUMP-KEY

MINUS
MIXED-MODE-ERROR
MODE
MODULES
MONITOR-DATE
MONITOR-SWITCH
MOVE
MULTIPLE
MULTIPLIED
MULTIPLY
NEGATIVE
NEXT
NO
NO-LENGTH-CHECK
NO-MEMORY-DUMP
NONE
NON-STANDARD
NO-OVERLAP
NO-PRINT-STORAGE
NO-RELEASE
NO-TAPEMARK
NO-TAPE-MARK
NORMAL-IO-RETURN
NOT
NOTE
NUMBER
NUMERIC
OBJECT-COMPUTER
OBJECT-PROGRAM
OCCURS
ODD
OF
OFF
OH
OMITTED
ON
OPEN
OPEN-WITHOUT-REWIND
OPTIONAL
OPTIONAL-USAGE
OR
OTHERWISE
OUTPUT
OUTPUT-FILES
OV
OVERFLOW
OVERLAP-FILE-AREA
PACKED
PADDING
PAGE
PAGE-COUNTER
PARITY
PERFORM
PERMANENT-WRITE-ERROR
PF
PH
PICTURE
PLACE
PLACES
PLUS
POINT
POSITION
POSITIVE
PREASSEMBLED

PREPARED
PRINTER
PRINTER-720
PRIORITY
PROCEDURE
PROCEED
PROGRAM-ID
PROGRAM-START
PROTECT
PROTECTION
PUNCH
PURGE-CYCLE
PURGE-DATE
QUOTE
QUOTES
RANGE
RD
READ
READER
RECORD
RECORD-COUNT
RECORDING
RECORD-MARK
RECORDS
REDEFINES
REDUNDANCY-ERROR
REEL
REEL-NUMBER
REELS
REEL-SEQUENCE-NUMBER
REEL-SERIAL-NUMBER
REFERENCE
RELEASE
REMARKS
REMEMBER
RENAMES
RENAMING
REPLACING
REPORT
REPORTING
REPORTS
RERUN
RERUN-RECORDS
RERUN-WORK
RESERVE
RESET
RESTART
RETENTION-CYCLE
RETENTION-PERIOD
RETURN
REVERSED
REWIND
RF
RH
RIGHT
ROUNDED
RUN
SAME
SD
SECTION
SECURITY
SEGMENT-LIMIT
SELECT
SELECTED

SENSE-SWITCH-1-6
SENTENCE
SENTINEL
SEQUENCE-CHECK
SEQUENCED
SHORT-ALPHA-WORD
SHORT-LENGTH-RECORD
SIGN
SIGNED
SIZE
SORT
SOURCE
SOURCE-COMPUTER
SPACE
SPACES
SPECIAL-NAMES
STANDARD
STANDARD-LABEL
STATUS
STOP
SUBTRACT
SUM
SUPERVISOR
SUPPRESS
SYNCHRONIZED
SYNCH-N
SYSTEM-INPUT-UNIT
SYSTEM-OUTPUT-PRINTER
SYSTEM-OUTPUT-PUNCH
SYSTEM-OUTPUT-UNIT
TALLY
TALLYING
TAPE
TAPES
TAPE-UNIT
TAPE-UNITS
TAPE-WITH-LABELS
TAPE-WITHOUT-LABELS
TERMINATE
TEST-PATTERN
THAN
THEN
THROUGH
THRU

TIME
TIMES
TO
TRAILER-LABEL
TYPE
TYPEWRITER
UNEQUAL
UNIT-RECORD-I-O-RECORD
UNOPENED-FILE
UNPACKED
UNTIL
UPON
UPPER-BOUND
UPPER-BOUNDS
USAGE
USE
USING
VALUE
VALUES
VARYING
WHEN
WITH
WITH-LABELS
WITHOUT-LABELS
WORDS
WORKING-STORAGE
WRITE
ZERO
ZEROES
ZEROS
1401-SS
1402-P
1402-R
1403-CT
1403-P
1403-P-CB
1403-P-CV
1403-P-C9
705-1
705-2
705-3
729-m
7080

APPENDIX G. ADDITIONAL CONSIDERATIONS

RESTRICTIONS ON THE USE OF THE COBOL LANGUAGE

The implied AT END with READ statement is disallowed on the 7080 COBOL Processor.

The format of a READ statement is:

READ file-name AT END any imperative statement

Each READ statement must have an explicit AT END. If no AT END is specified by the programmer, the Processor will create an ending routine, consisting of a loop, for that particular READ statement.

LANGUAGE INCOMPATIBILITY

If the COBOL program is to be compiled for more than one computer, the following verbs and clauses should be avoided as much as possible since their definitions may not be compatible in all systems.

Verbs

The following COBOL verbs and their associated language specifications cannot be defined in compatible terms in any two systems.

1. ACCEPT
2. UPON option of the DISPLAY verb
3. ENTER
4. USE

Clauses

The three editing clauses, ZERO SUPPRESS, CHECK PROTECT, and FLOAT DOLLAR SIGN are redundant, since the PICTURE clause can be used for all their functions. In addition, certain errors in their specification, resulting in conflicting requirements, make implementation difficult. It is therefore recommended that PICTURE be used whenever editing of this kind is required.

Special Characters

The 7058 and 7080 COBOL Processors permit alphanumeric literals to contain any character in the computer's character set. However, only the COBOL character set will insure complete compatibility with other IBM systems.

ACCEPT 8, 17
 ADD 2, 19
 ADDRESS 8
 ALPHABETIC 15, 17
 ALPHANUMERIC 15, 17
 Alphanumeric literals 6
 AN 15
 Analysis run 6
 APPLY 11
 Assembling an IOCS:
 For the 705 III 24
 For the 7080 28
 ASSIGN 10, 11
 ASU settings 18
 ASUSAVE 23
 AT END 2, 17, 29, 34
 Autocoder tags 7, 9, 18
 AUTOLOADER 10, 11, 12

BEGINNING-HYPERTAPE-LABEL 14, 20
 BEGINNING-REEL 11, 21, 22
 BEGINNING-TAPE-LABEL 14, 20
 BLASU06 18
 BLASU12 18
 BLHSK 24, 26
 BLIXT 2, 21
 BLOCK 10, 13
 BLSY 24
 BLSYS 24, 25

CASU settings 21
 Channel tables 8, 24
 Character set 7
 CHARACTERS 8, 13
 Checking:
 of labels 9, 20
 of record count 20
 of record length 23
 of reels 9, 20
 CHECKPOINT 13
 Checkpoint taking 9, 11, 12, 13, 23
 CLASS 15
 Class B subroutines 24
 CLOSE 12, 17, 18, 29
 Compilation listing 6
 Compilation run 6
 Conditional statements 17
 Condition-names 7
 Configuration Section 7
 Control cards 20
 CORRESPONDING 2, 19

DATA RECORD 10
 Date 7
 Date control card 20
 DATE-COMPILED 7
 Deblocking 9
 Decimal point 15

Deferred features 29
 Device name formats 7
 Diagnostic messages 6
 DISPLAY 8, 15, 18

ELSE 17
 End control card 20
 END OF RUN message 19
 ENDING-FILE 11, 20
 ENDING-HYPERTAPE-LABEL 14, 20
 ENDING-REEL 12, 20
 ENDING-TAPE-LABEL 14, 20
 ENTER 18
 ENTER AUTOCODER 9, 21
 Environment Division 7
 ERROR 17, 19
 Error Checking 20
 ERROR-DUMP 10

FILE-CONTROL 9
 File Description entries 13
 FILE-PROTECT 12
 File Serial Header control card 20
 Format of device names 7
 Format of input and output 12
 FROM 17

GO TO 11, 18

Hash totals 23
 Header labels 9, 14
 HIGH 13
 HIGH-VALUE 7
 Hypertape capabilities 2, 5

Identification Division 7
 IF 17
 Incompatibility 34
 INITIATE mode 23
 Input/Output section 9
 Input reel checking 9
 INTERCHANGEABLE 13
 I-O-CONTROL 11
 IOCS use 23
 IODEF 24, 25

Label control cards 20
 LABEL-DATA 11
 Labels 9, 13, 20
 LASN 24
 LEFT 14
 Limitations on verb formats 17
 Listing, from compilation 6
 Literals 7
 LITOR 24

LOCK 18
 LOW 13
 LOW-VALUE 7

Macro-instructions:
 BLHSK 24, 26
 BLIXT 21
 BLSY 24
 BLSYS 23, 24
 IODEF 24
 MEMORY 8
 Messages, diagnostic 6
 Mnemonic name 8
 MOVE 15
 MULTIPLE REEL 10

NO 10
 NO-LENGTH-CHECK 13
 NO-TAPEMARK 14
 NUMERIC 15, 17
 Numeric items 15
 Numeric literals 6

OBJECT-COMPUTER 8, 13, 17, 24
 OCCURS 17
 OMITTED 14
 ON SIZE ERROR 17
 OPEN 17, 18, 23
 OTHERWISE 17
 Output reel checking 9

PACKED 13
 PERFORM 11, 18
 PICTURE 14
 PREASSEMBLED 8, 24, 26,
 Preassembled IOCS 24, 26
 Procedure Division 17
 PROGRAM-ID 7
 PROGRAM-START 9

QUALIFICATION 17

READ 17, 23, 29, 34
 RECORD 10, 13
 Record blocking 9
 Record counts 12, 20
 Record Description entries 14
 Record length checking 23
 RECORDING 13
 REDEFINES 16
 RENAMING 10, 17
 Report items 15
 RERUN 12
 RERUN-WORK 11
 RESERVE 10

NOTE: Underscored numbers indicate the principal reference.

Reserved words 30
RESTART 11
Restrictions 34
RIGHT 14
ROUNDED 19
RUN 19

SELECT 10
SIZE 16, 17, 19
Size errors 17
SOURCE-COMPUTER 7

SPECIAL-NAMES 8, 17
STACKING mode 23
STANDARD 14
STOP 19
SYNCHRONIZED 14
SUBTRACT 2, 19

Tape Reel Control System 9
Tape unit assignment 10
Trailer labels 9, 14

UNPACKED 13
UPON 17, 18
USAGE 15
V label option 23
VALUE 16

WRITE 18

ZERO 7

